Course 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.

Course description

Developing Solutions with IBM Decision Server Insights V8.8

Duration: 4 days

Purpose

In this course, you learn about the main features of the Decision Server Insights component of IBM Operational Decision Manager Advanced V8.8.

Decision Server Insights enables real-time, in-memory, rule-based, event-driven, and analytical decision making. You experience how to use analytics, time-based reasoning, and location-based reasoning to build a real-world solution that detects and responds to business situations. You also learn the key capabilities of the multi-agent architecture of Decision Server Insights by developing several agents that are bound to a single entity for different purposes.

This course focuses on solution development, deployment, testing, and administration. You learn how to implement the business logic that detects business situations and uses situational context to decide and take the next best action.

The course begins with an overview of the programming model for Decision Server Insights and the architecture for the Decision Server Insights runtime environment. You learn how to design a Decision Server Insights solution, model the business entities and events that you care about, and implement the business logic. You work with a realistic test client to test the behavior of your implementation after deployment.

The course also covers administration topics, including installation, configuration of the Decision Server Insights reference topology, solution deployment in a grid environment, and grid administration.

Audience

This course is designed for developers.

Prerequisites

Before taking this course, you should have:

- Experience with the Java programming language and object-oriented concepts
- Basic knowledge of Extensible Markup Language (XML)
- Basic knowledge of the WebSphere Application Server Liberty profile
- Familiarity with the Representational State Transfer (REST) architectural style
- Familiarity with WebSphere eXtreme Scale

Objectives

- Describe the Decision Server Insights programming model and architecture
- Design and create a Decision Server Insights solution
- Define the business model for the events, entities, and concepts that are relevant to your domain
- Use global aggregates for calculations across all events or a population of entities
- Implement business logic with rule agents and rules to detect and respond to business situations
- Deploy solutions to the Insight Server runtime and test runtime behavior
- Explain Decision Server Insights integration capabilities

Curriculum relationship

Decision Server Insights is a module of IBM Operational Decision Manager Advanced V8.8.

- Developer topics for IBM Operational Decision Manager V8.8 are covered in the course: WB395-ZB395, Developing Rule Solutions in IBM Operational Decision Manager V8.8

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:30) Course introduction
(01:30) Unit 1. Introducing IBM Decision Server Insights
(01:30) Exercise 1. Getting started with Decision Server Insights
(01:30) Unit 2. Designing Decision Server Insights solutions
(00:15) Exercise 2. Creating a solution in Insight Designer
(01:00) Unit 3. Creating the business model
(00:30) Exercise 3. Defining the business model
(01:30) Unit 4. Authoring the business logic

Day 2

(00:30) Exercise 4. Creating a rule agent
(00:30) Exercise 5. Writing and testing rules
(00:45) Unit 5. Working with aggregates
(01:15) Exercise 6. Using global aggregates in rules
(00:45) Exercise 7. Using event aggregates in rules
(00:30) Exercise 8. Using time-based and location-based reasoning in rules
(00:30) Exercise 9. Testing for the absence of events
(01:00) Unit 6. Testing solutions
(00:30) Exercise 10. Testing solutions
(00:30) Exercise 11. Using the Map Viewer

Day 3

(01:00) Unit 7. Modeling and defining connectivity
(00:30) Exercise 12. Defining connectivity
(01:00) Unit 8. Integrating Decision Server Insights
(01:00) Unit 9. Configuring Insight Server
(01:00) Exercise 13. Installing Decision Server Insights
(02:00) Exercise 14. Configuring Decision Server Insights

Day 4

(01:00) Exercise 14. Configuring Decision Server Insights
(01:00) Unit 10. Managing deployment
(01:30) Exercise 15. Deploying solutions
(01:00) Unit 11. Administering Decision Server Insights
(01:00) Exercise 16. Administering Decision Server Insights
(00:30) Unit 12. Course summary

Unit 1. Introducing IBM Decision Server Insights

Estimated time

01:30

Overview

This unit introduces you to the Decision Server Insights programming model and architecture.

How you will check your progress

- Review
- Exercise

IBM Training

Unit objectives

- Describe Decision Server Insights and explain how it works
- Explain the programming model
- Describe the Decision Server Insights architecture
- Outline the user roles that are associated with Decision Server Insights

Figure 1-1. Unit objectives

IBM Training

How to check online for course material updates

Note: If your classroom does not have Internet access, ask your instructor for more information.

Instructions

1. Enter this URL in your browser: **ibm.biz**/CloudEduCourses
2. Find your course in the list and then click the link.
3. The wiki page displays information for the course. If there is a course corrections document, this page is where it is found.
4. If you want to download an attachment, such as a course corrections document, click the **Attachments** tab at the bottom of the page.

5. To save the file to your computer, click the document link and follow the prompts.

Introducing IBM Decision Server Insights

Figure 1-2. How to check online for course material updates

IBM Training

Topics

- Overview of Decision Server Insights
- How Decision Server Insights works
- Decision Server Insights architecture
- Decision Server Insights roles
- Installing Decision Server Insights

Introducing IBM Decision Server Insights

Figure 1-3. Topics

This slide shows the list of topics for this unit.

1.1. Overview of Decision Server Insights

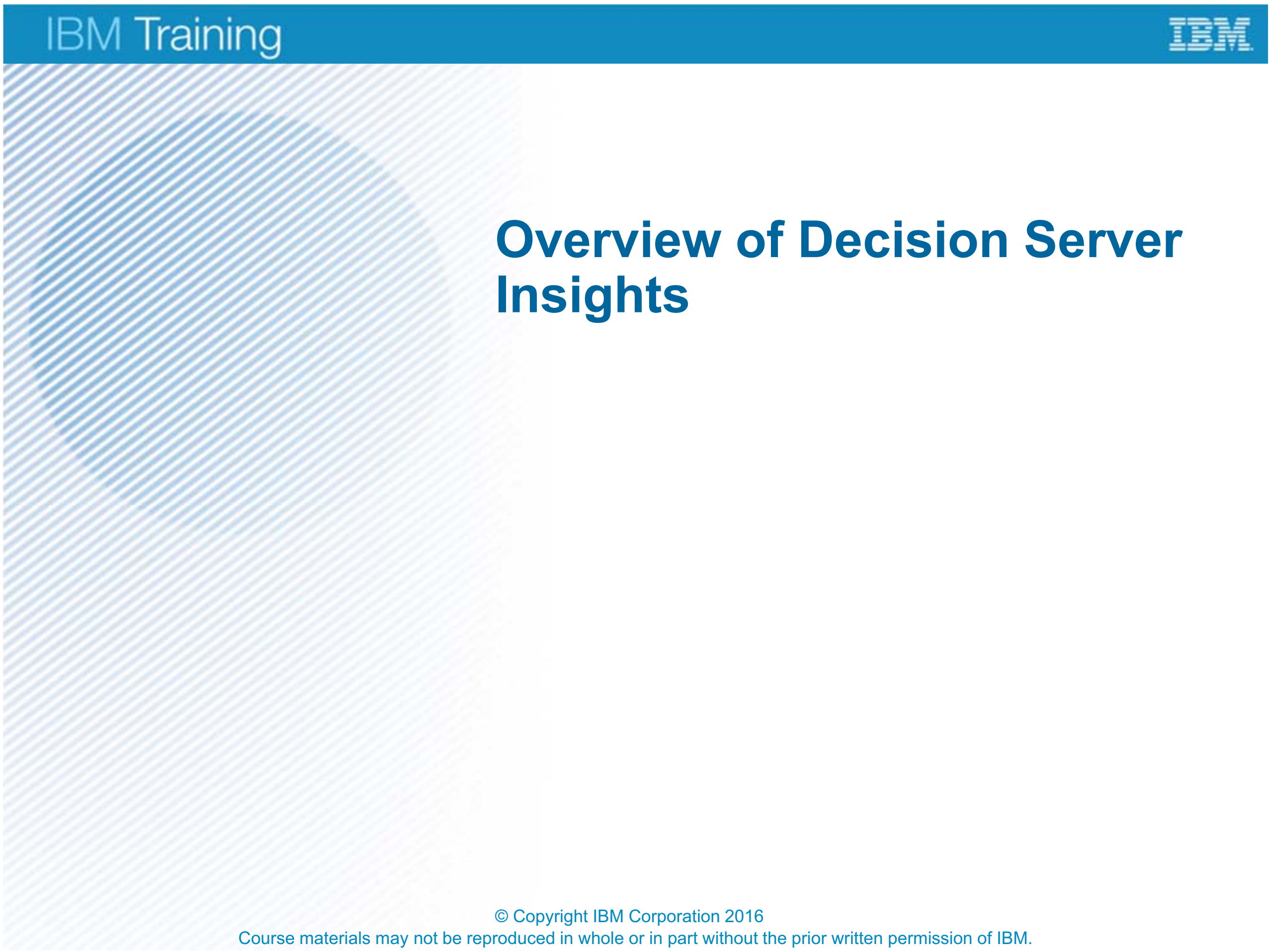


Figure 1-4. Overview of Decision Server Insights

IBM Training

What is Decision Server Insights?

- Decision Server Insights is a module in IBM Operational Decision Management Advanced
- Combines rule, events, and predictive analytics on a single, elastic platform
  - Identifies risks and opportunity on a continuous basis
  - Helps applications to decide and act at the precise moment your business needs it
- Model-driven, scalable, transactional event processing system with rule-based temporal reasoning and analytics capabilities
  - Build scalable solutions that listen for and respond to events that affect your business
  - Use the insights that are generated from these business activities to make informed decisions and initiate appropriate actions

Introducing IBM Decision Server Insights

Figure 1-5. What is Decision Server Insights?

Decision Server Insights provides the flexibility and agility of **prescriptive decision management** in a situational context.

Decision Server Insights is included as a component of IBM ODM Advanced.

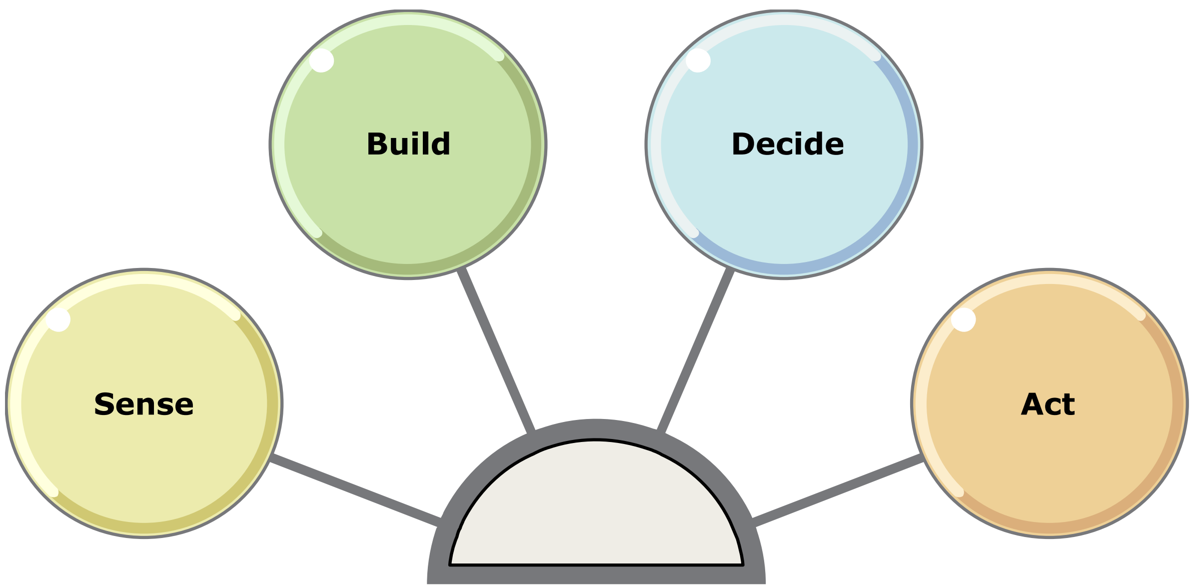
Decision Server Insights combines business rules, event processing, and predictive analytics tools on a single elastic platform. By using Decision Server Insights, you can build applications that make decisions in near real-time and in situational contexts.

Decision Server Insights adds value for use cases where businesses need to make sense and take advantage of past and present events and data to proactively detect typical business situations. These situations include both risks and opportunities, and businesses can use Decision Server Insights to respond with the right action at the right time.

IBM Training

Four steps from data collection to intelligent business action

Goal: Gain insight from your data in real time, so you can act while it can make the greatest difference



Introducing IBM Decision Server Insights

Figure 1-6. Four steps from data collection to intelligent business action

The revolution of mobile, the advent of the Internet of Things, and the emergence of the digital economy have all contributed to an overwhelming increase in data. Also, the pattern of interaction between an enterprise and its customers and its employees is changing. Customers and clients do not expect the enterprise to respond only. They expect the enterprise to anticipate their needs without them asking.

ODM Advanced is about putting events and data into context to extract relevant insight to support further decisions. From those insights, you can then apply analytical models and rules to make the decisions and then propagate those decisions to the places that need the action, including processes and applications.

However, merely collecting data does not help you manage the response. The goal of data collection is to gain insight from what the data tells you, so you can act in real time to improve business results.

Decision Server Insights helps you accomplish this goal through a four-step process summarized as **Sense, Build, Decide, Act**:

- Sense what is happening.
- Build the context **of your situation**.
- Decide what to do when **something happens that affects the situation**.

- Then, Act **<u>on the changed situation</u>** quickly and consistently.

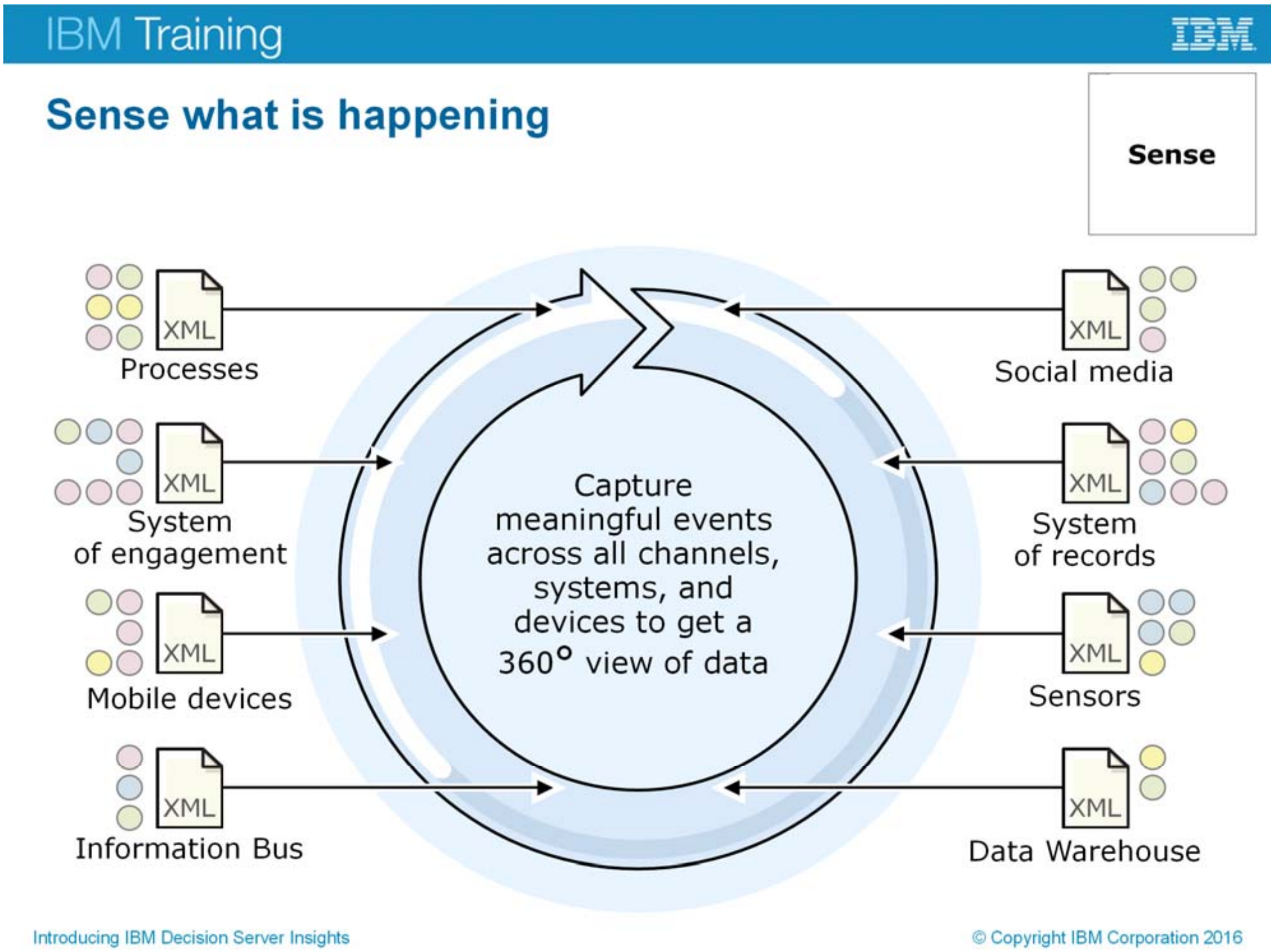


Figure 1-7. Sense what is happening

To Sense what is happening, you must be able to recognize and capture events from the outside world.

Decision Server Insights listen to a wide range of event providers, such as social media, physical sensors, mobile devices, and other channels or systems as depicted here. The Internet of Things, mobile networks, social networks, and call centers provide the data that Decision Server Insights needs. Decision Server Insights uses this data to listen to for use cases such as healthcare, fleet management, banking, real-time promotions, and real-time churn management.

Events are received as XML messages by Decision Server Insights.

While many organizations have silo applications for each of the event sources, Decision Server Insights provides “event fusion” to get a 360-degree view across all event sources.

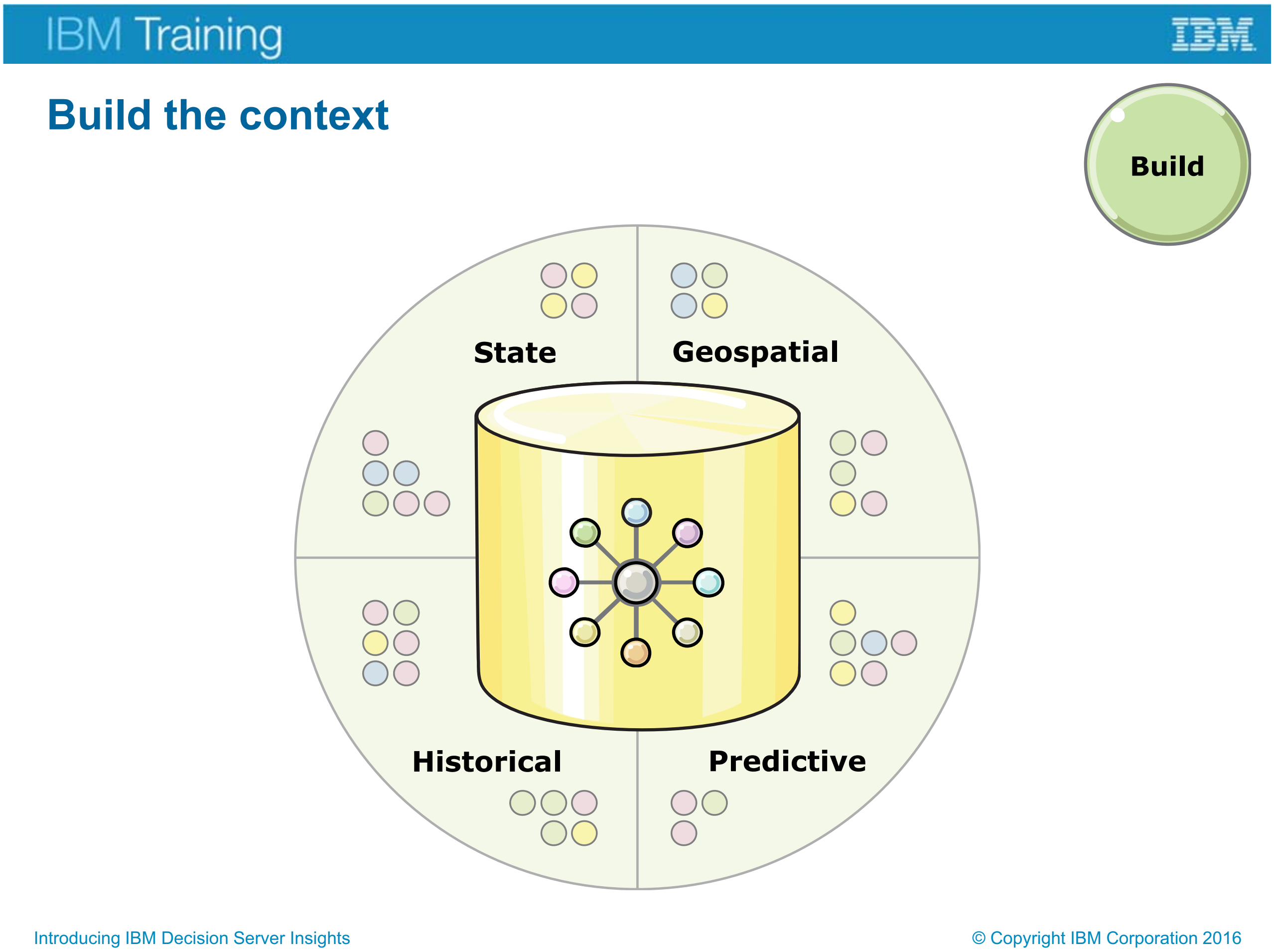


Figure 1-8. Build the context

Next, building the context involves putting data and events into context to make sense of the data and evaluate correlations.

The context is built from basic state information, geospatial information such as locations, predictive information **from** using predictive models, and from historical information that you recorded.

As events occur that involve the business entities that you care about, such as customers or accounts, Decision Server Insights builds a context for each entity.

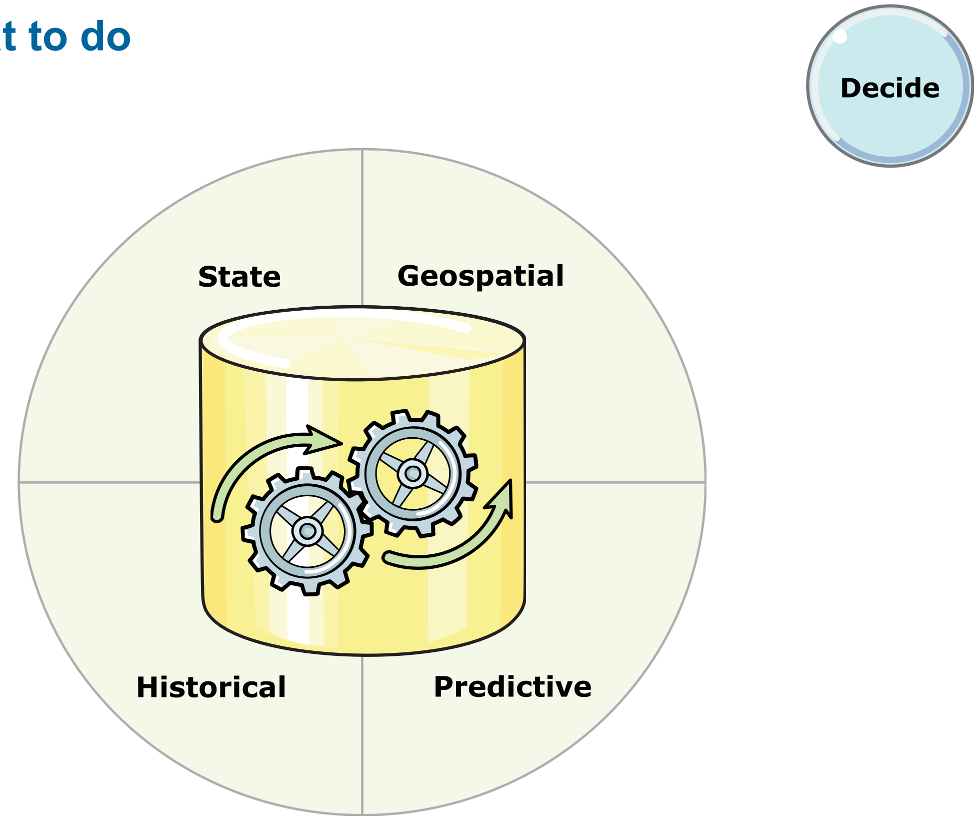
The context captures what is known about:

- The past of the entity, which is modeled by accumulated entities
- The present, which is modeled by the current state, including the current geospatial position
- The future, which is represented by predictive score that indicates what is likely to happen, such as the propensity of your client to churn, or switch to another company

You depend on the context that you built and you determine when the situation occurred, when you can take advantage of a potential opportunity, or when you can avoid a risk. After you have this type of insight, you can then determine how to act.

IBM Training

Decide what to do



Introducing IBM Decision Server Insights

Figure 1-9. Decide what to do

In Decision Server Insights, you implement the business logic to apply the models, policies, and suggested practices as established by your subject matter experts.

The business rules are expressed as *situation detection* patterns. When a relevant business situation is detected, the rules tell you the next best action to take.

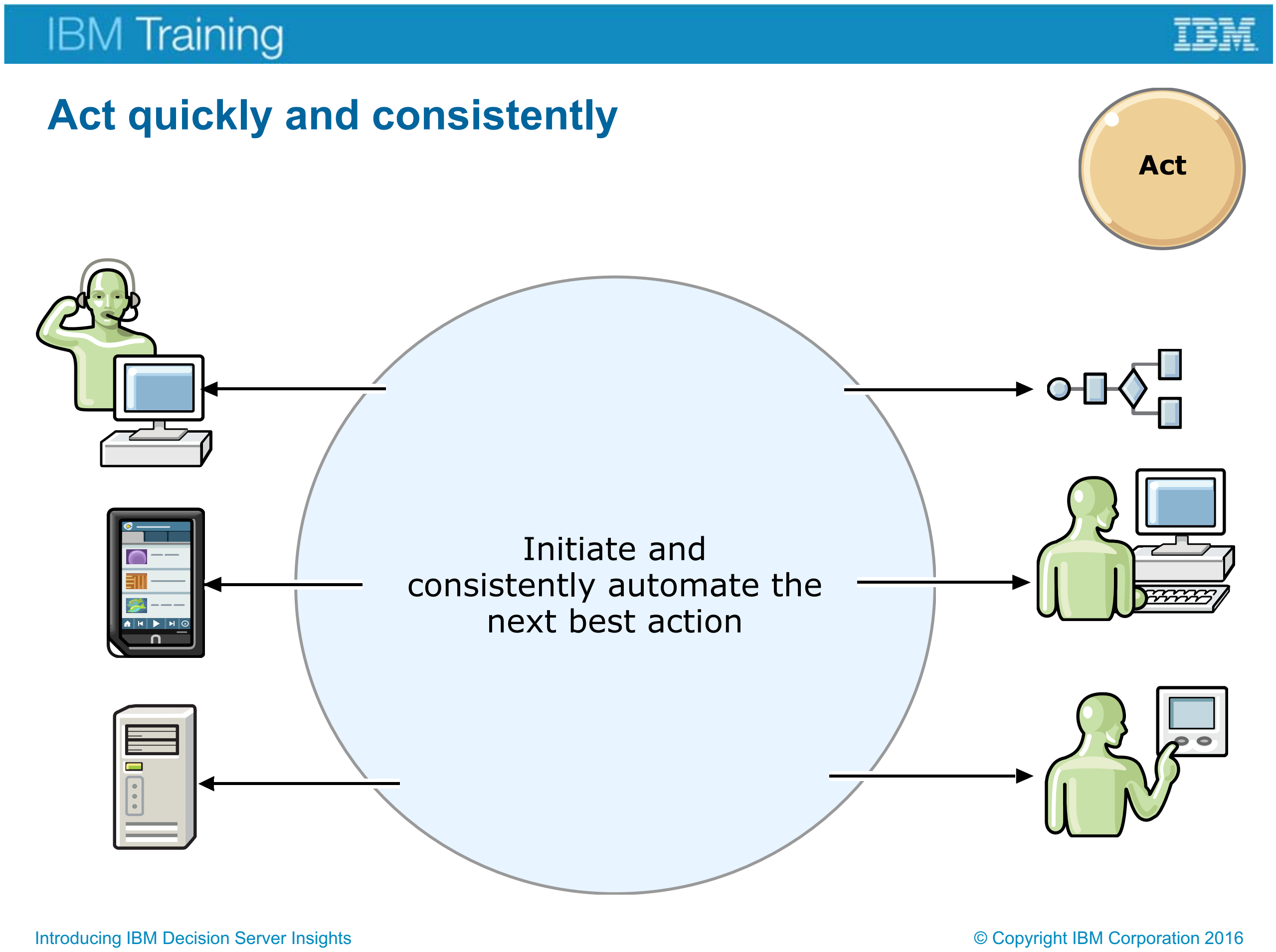


Figure 1-10. Act quickly and consistently

Finally, you act.

Decision Server Insights emits a response event. That event is published as an XML message on the outbound queue or posted to a URL as HTTP for some external system to pick up. The message can be transformed into something that the external system can use to interact with the outside world.

Actions can range from alerting systems to risk and opportunity, to maximizing the efficiency of your operations, to predicting equipment maintenance.

1.2. Decision Server Insights programming model

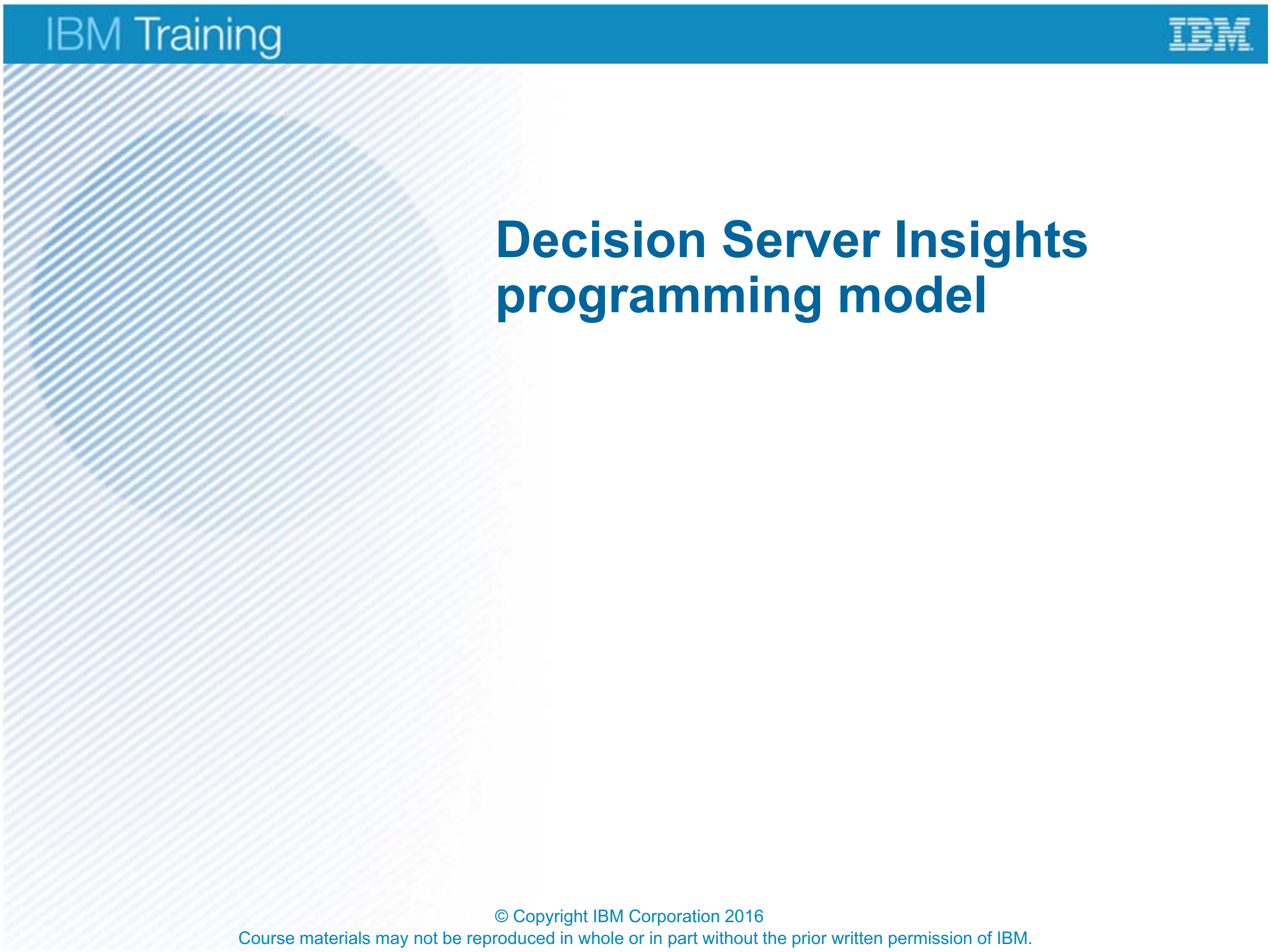


Figure 1-11. Decision Server Insights programming model

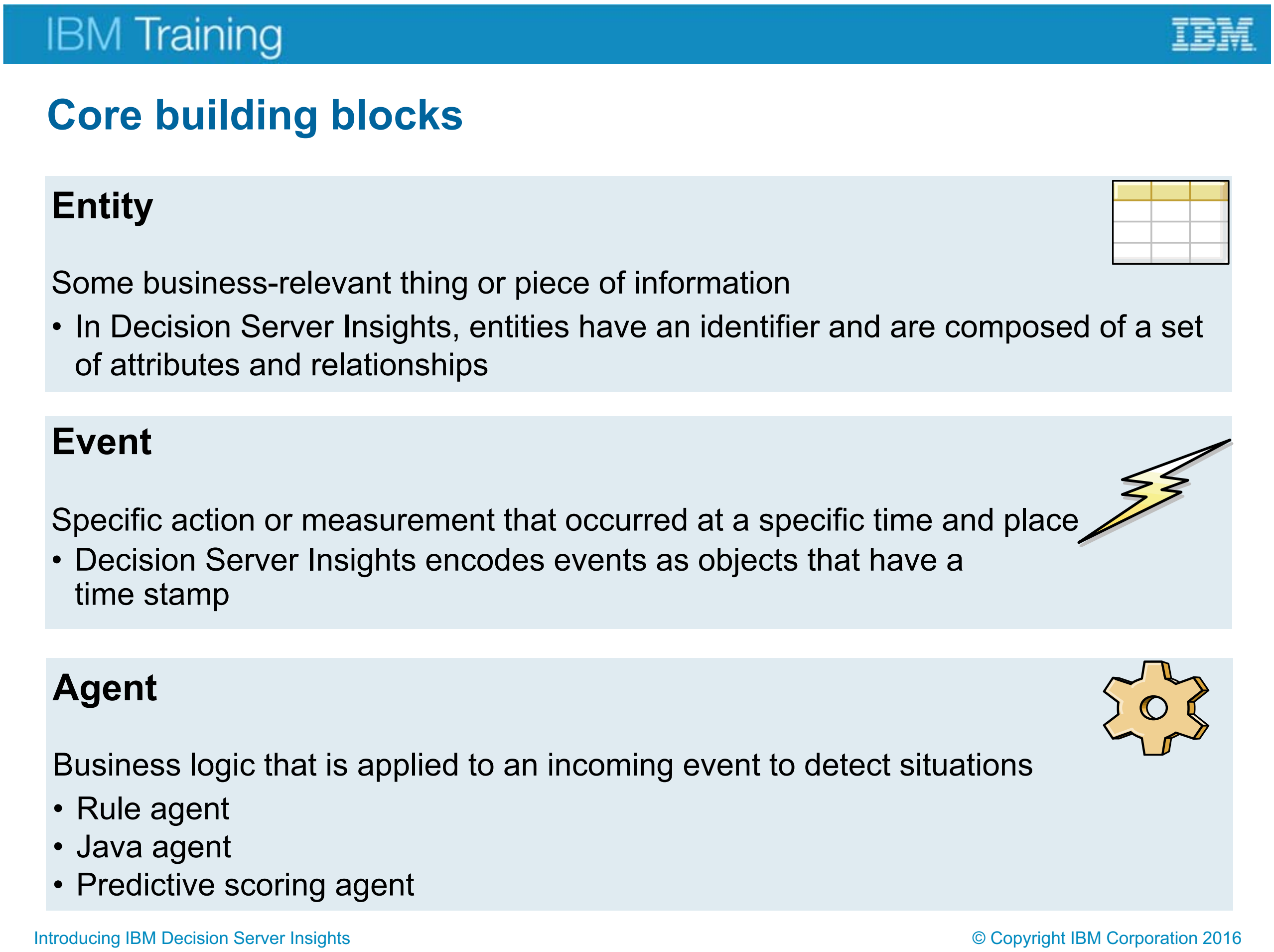


Figure 1-12. Core building blocks

Decision Server Insights uses three core building blocks: events, entities, and agents.

Entities model real-life objects, such as the client of a bank or an ATM. The entity provides insight into what happened in the past, what events came in, the current state of the entity and what is likely to happen in the future.

Events are actions or occurrences that happen in real life, at a certain moment. Every event is timestamped, and potentially geo-localized.

An event can affect one or more entities. For example, a withdrawal event relates to an ATM entity, a client entity, and an account entity.

The third building block is the agent. Decision Server Insights provides three types of agents:

- Rule agents
- Java agents
- Predictive agents

Rule agents take advantage of ODM rules language, making it possible for business stakeholders to manage the business logic to react to the events.

Java agents are written in Java code.

Predictive scoring agents are an extension of the Java agent, with a built-in API for invoking an SPSS Scoring Configuration. Associated with each entity, you have a number of predictive scores. If the entity models a client, the entity carries the predictive score that represents that client's propensity to defect or churn. These predictive scores can then be used in the rules to make a decision.

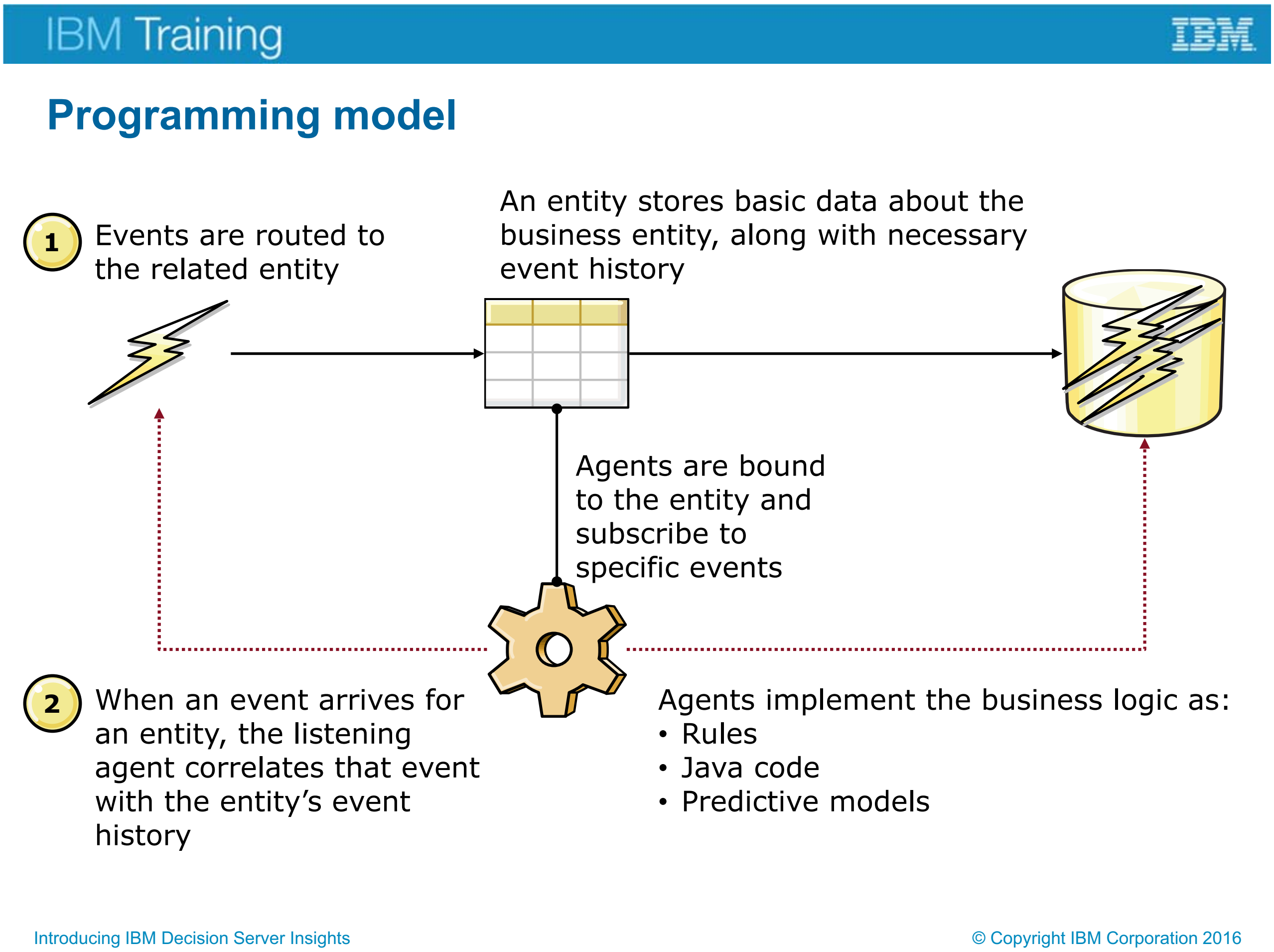


Figure 1-13. Programming model

This diagram outlines how the programming model works with those building blocks.

Starting with an event, the event is routed to the entity.

The entity stores basic data about the business entity, and any event history that is needed to detect the situation and patterns, and to calculate the analytics.

At the same time, the agents are bound to entities and process specific events. The agents are logic fragments that implement the business logic.

At design time, an agent is bound to an entity and subscribes (or listens) to certain types of events. Multiple agents can be bound to the same entity and listen for the same events, but for different purposes.

When an event arrives and invokes an agent, the agent evaluates the event that occurred, the state of the entity, and most importantly the context. The context contains insight into the past events that are related to the entity. The agent also evaluates the current state and location of the entity, and the likely future of the entity through predictive scores.

This access enables the agent to do event fusion by looking for patterns across these three streams. Agents can emit new events, either internal to the solution to trigger additional event-entity-agent bindings, or externally, to trigger system actions.

IBM Training

Analytic capabilities

- Decision Server Insights performs complex event processing and powerful in-memory analytics
  - Listens to events across multiple channels and detects patterns that represent situations of interest to the business
  - Sophisticated time management, correlation, and aggregation capabilities
- Global aggregates provide the ability to run calculations across your entire population or subset of entities, and across all your events
  - Run computations on distributed event or entity data to generate an aggregated value
  - Define global event and global entity aggregates in a solution to perform calculations across your data model and identify outliers
- Decision Server Insights provides easy integration with SPSS predictive scoring models
  - Simplified integration with SPSS Collaboration and Deployment Services Server

Introducing IBM Decision Server Insights

Figure 1-14. Analytic capabilities

Analytics plays a major role in Decision Server Insights. You have support for both ***global*** analytics and ***predictive*** analytics.

With global analytics and aggregates, you can extract valuable insights over entire populations of business entities in near real-time and use them to further optimize your decision models.

A global aggregate is a value that is computed from a collection of objects.

Decisions Server insights can compute global aggregates on entities or on events. Event aggregates continuously perform calculations on incoming events. Entity aggregates perform calculations across all instances of an entity type or a selected subset of that entity population. The calculations are scheduled to be performed regularly, on a schedule that you decide, for example, everyday.

This type of measurement can help you find entities or events that do not match the average. For example, consider a situation where you have some average measurement for a population of customer entities and you detect that certain customers are well below that average. You can then take a special action for those customers that fall outside the average. The idea is that these aggregates can help you focus on the most relevant cases.

But Decision Server Insights is not merely reactive. It also helps you get ahead of the events by integrating with SPSS predictive models. You can use the aggregate values that are managed with

Decision Server Insights, send them to be modeled in SPSS, and from SPSS you can receive a likelihood score. The score is then used to trigger the prescribed next best action.

1.3. Decision Server Insights architecture

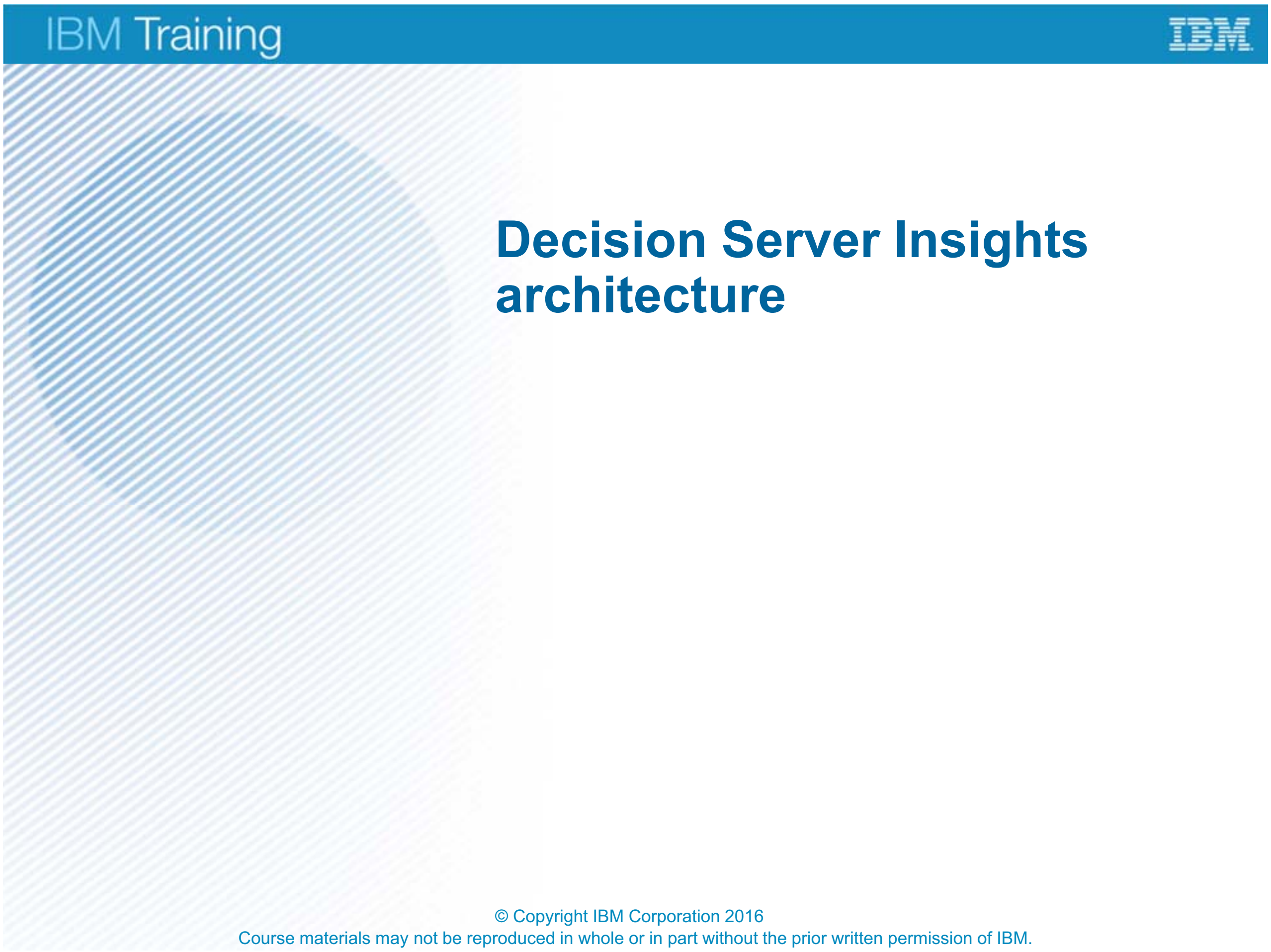
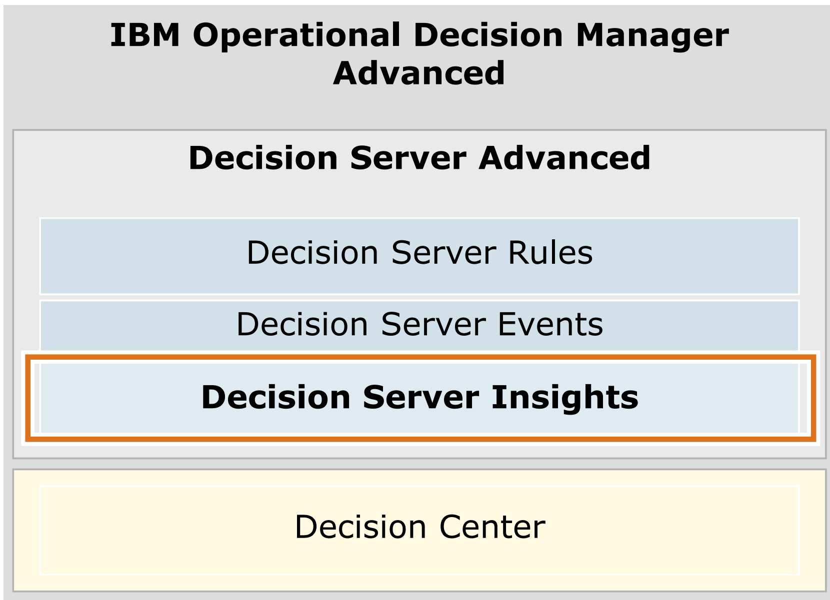


Figure 1-15. Decision Server Insights architecture

IBM Training

IBM Operational Decision Manager Advanced offering

- Decision Server Insights is a module in IBM Operational Decision Manager Advanced V8.8



Introducing IBM Decision Server Insights

Figure 1-16. IBM Operational Decision Manager Advanced offering

Decision Server Insights is exclusively available with the IBM ODM Advanced edition.

For more information about the IBM ODM packaging, see the product documentation.

IBM Training

Decision Server Insights components

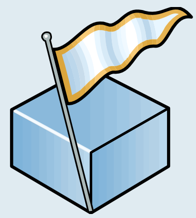
Decision Server Advanced: Decision Server Insights

Insight Designer

Eclipse IDE

- Model entities
- Model events
- Create agents or aggregates
- Manage business logic
- Solution and connectivity deployment

Insight Server



- Route events
- Persist context
- Manage aggregates
- Execute agents

Insight Inspector

- Record and analyze event activity
- Visualize event timeline in a browser

Introducing IBM Decision Server Insights

Figure 1-17. Decision Server Insights components

Decision Server Insights has a similar structure as ODM Decision Server Rules, but for the moment it has no interaction with Decision Center. Decision Server Insights includes Insight Designer, which is a development environment in Eclipse. Insight Server is a runtime environment that handles complex event processing and agent execution. You use Insight Inspector, which is a browser-based tool, to visualize event processing.

During the course, you work extensively with Insight Designer and Insight Server. You also work with Insight Inspector.

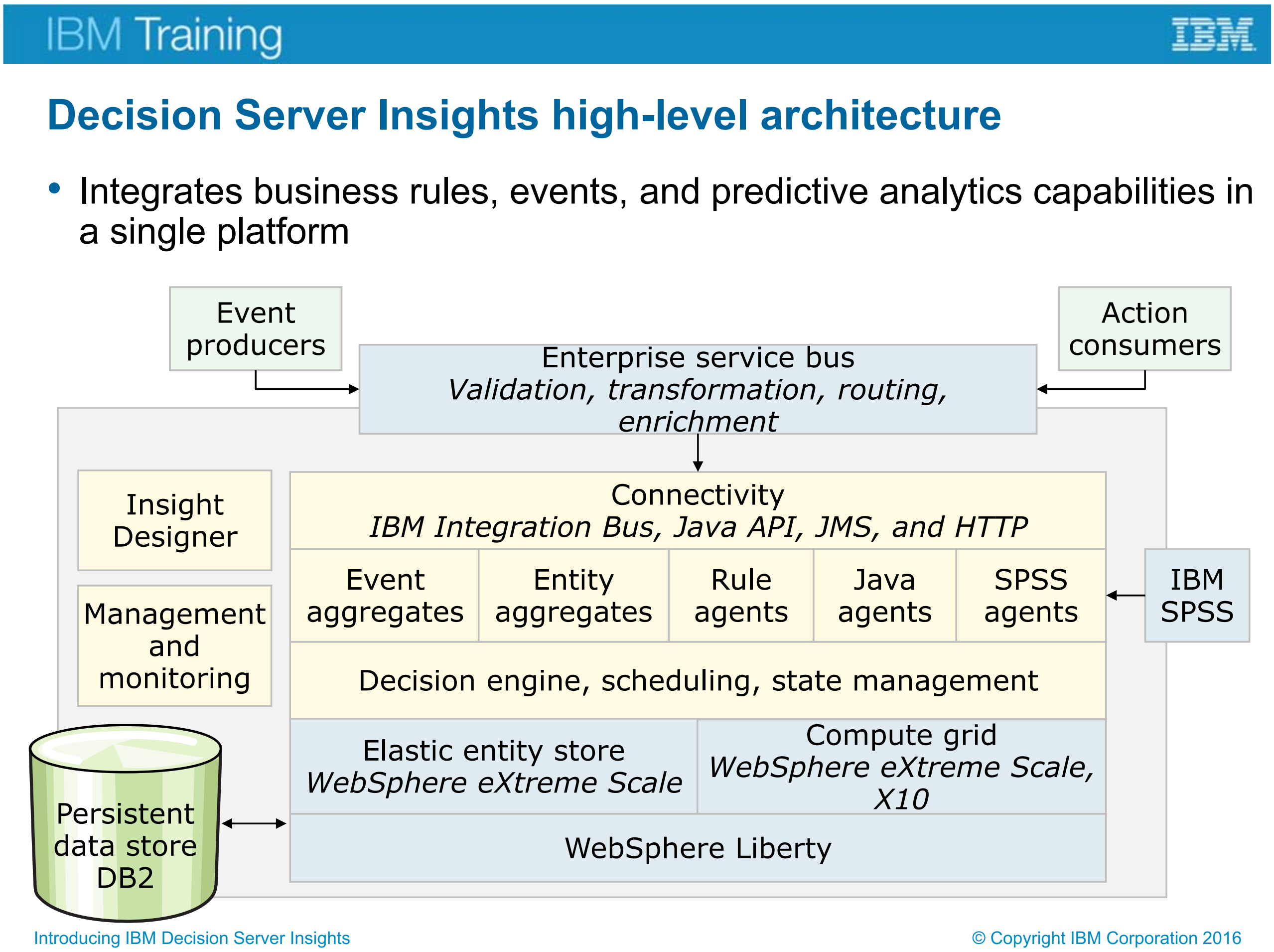


Figure 1-18. Decision Server Insights high-level architecture

The Decision Server Insights architecture provides reliability, elastic horizontal scalability, and automatic state management to evaluate your business logic. It integrates business rules, events, and predictive analytics capabilities in a single platform.

As shown in this diagram, Decision Server Insights uses WebSphere Liberty, which is a modular OSGi micro kernel, and WebSphere eXtreme Scale as an elastic and scalable in-memory compute and data grid.

WebSphere eXtreme Scale is an elastic, scalable, in-memory data-and-compute grid that can store millions of entities and large event histories. Extreme scale is used as a data grid for high-performance access to data. Extreme scale is also used with X10 as a compute grid to execute business logic as close to the data as possible. X10 provides a globally distributed, asynchronous programming model.

Next, you see the core Insights components, including the decision engine for rule execution, the scheduling capability, and state management.

With this architectural design, the Insights runtime takes care of all the complexity and horizontal scalability for you.

And you do that in the Eclipse-based Insight Designer. Insight Designer helps you easily model the entities and events, implement the business logic agents, and build the aggregates, including easy integration with SPSS scoring models.

Management is done by a set of JMX MBeans and RESTful web services. You can also use IBM SmartCloud Analytics Embedded to run diagnostic tests, check logs, and generate analytical charts from real-time data.

The runtime receives events through connectivity infrastructure (shown as the top layer). The connectivity tier works with an Enterprise Service Bus, which connects to external systems to produce the events and consume the actions. Event bus is not part of the product, but is a desirable prerequisite to facilitate integration by having one single place where all the events are published. You can define the connectivity by using HTTP, JMS, Java API, or IBM Integration Bus protocols.

For disaster recovery, you can choose to write runtime state to a backend data store for disaster recovery. DB2 is the supported database for restoring persistent data.

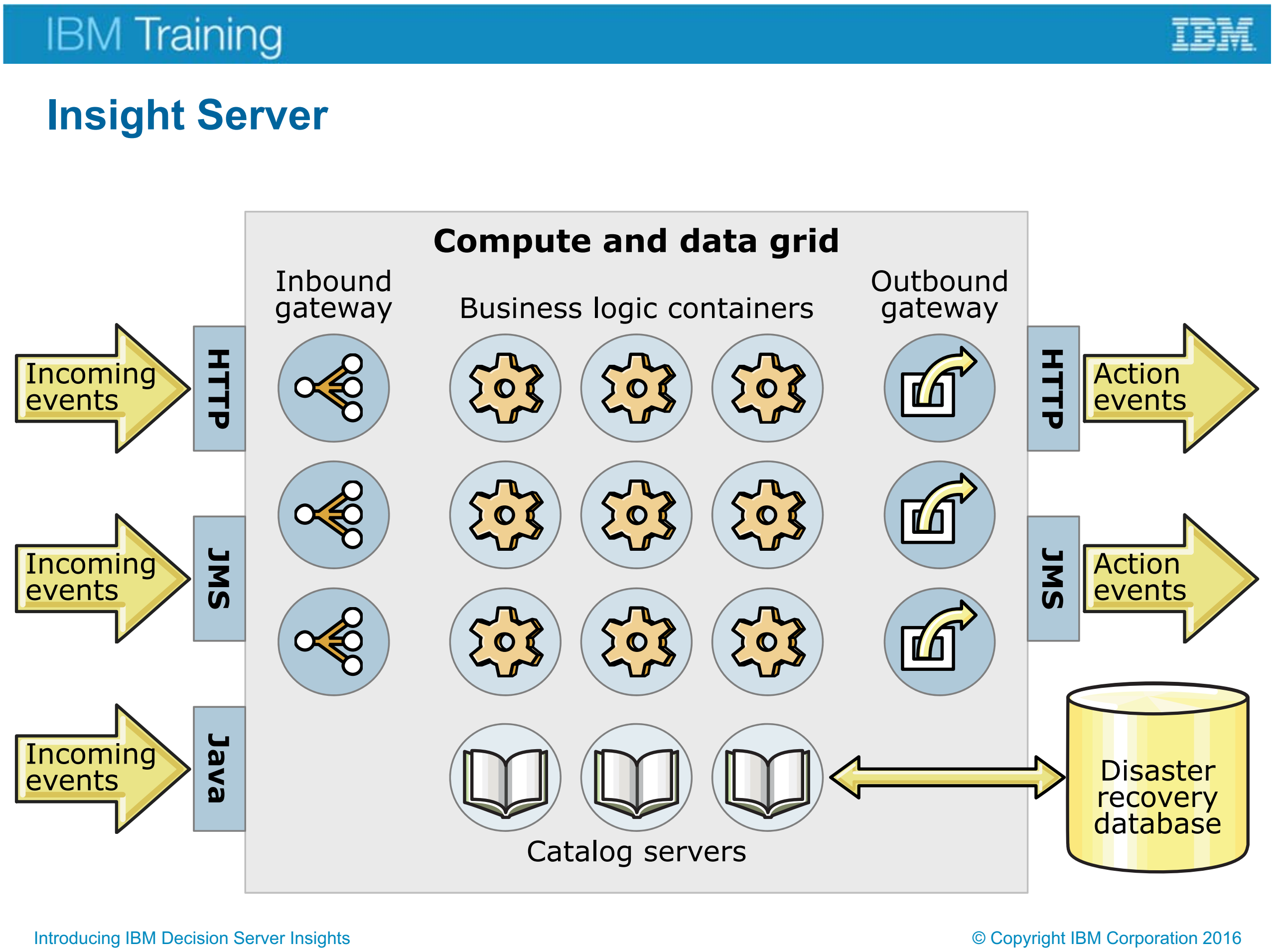


Figure 1-19. Insight Server

The business logic containers are the servers where you store your events and run the agents.

With this architecture, you can dynamically add connectivity and computing resources to keep up with the workload. It collocates the rules and analytics computing resources with context data. By running the computations where the data is, data movement is minimized, which results in the best performance.

This architecture provides the capability to analyze millions of interactions and maintain the context over periods of days weeks or even months.

1.4. Decision Server Insights roles

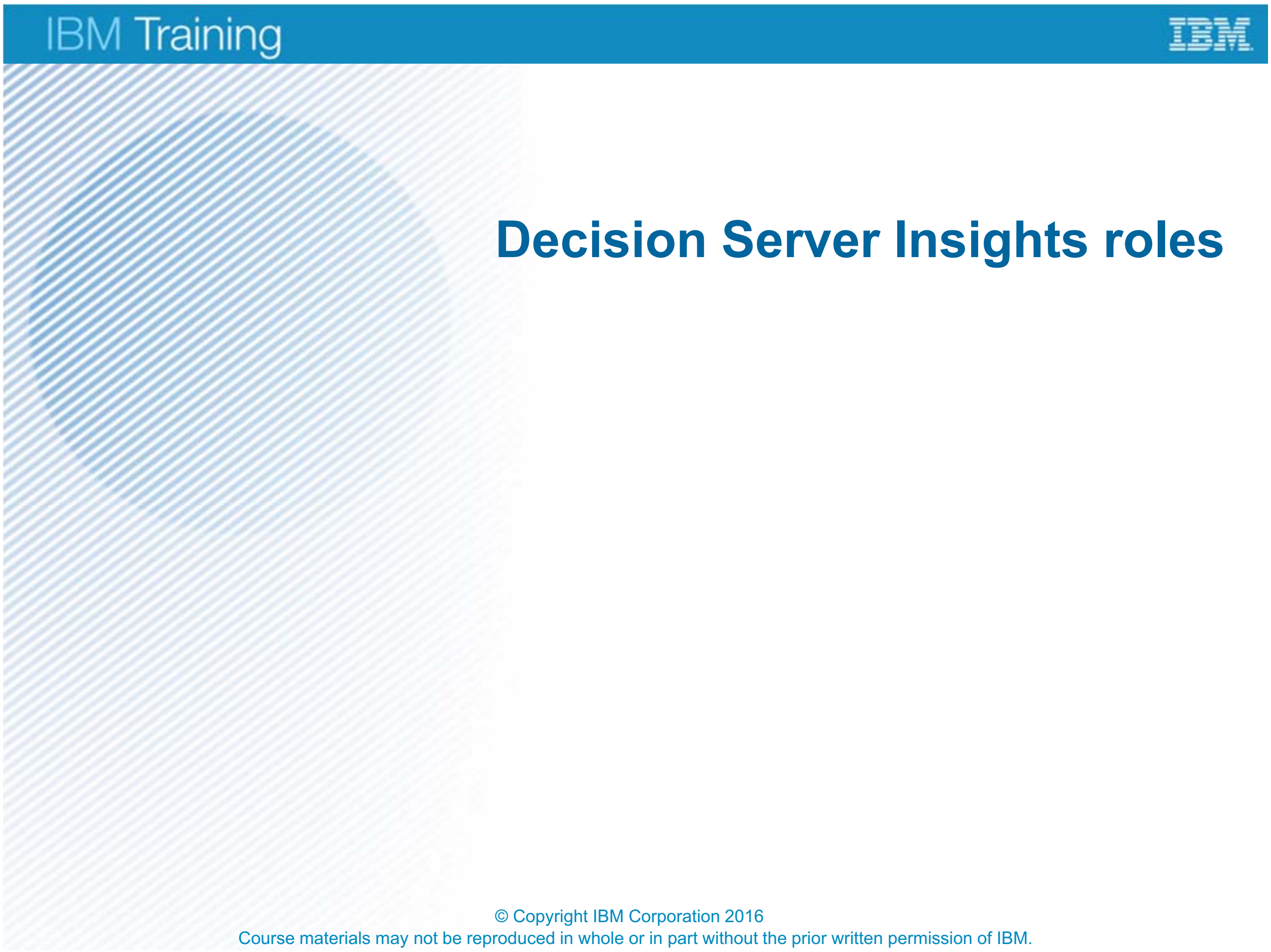


Figure 1-20. Decision Server Insights roles

IBM Training

Decision Server Insight roles

Business users		
	Modeler	Models the definitions that are required for the solution and authors rules
Technical users		
	Architect	Integrates configurations for insight solutions, identifies incoming events that require updates to entities, outbound events that are emitted to external systems, and general management of integration touch points
	Developer	Develops, tests, and deploys solutions that include entity models, event definitions, and agents to process the events
	Administrator	Installs, configures, and maintains all the tools across the development, staging, and production environments

Introducing IBM Decision Server Insights

Figure 1-21. Decision Server Insight roles

Who works with Decision Server Insights?

As you see listed here, users are usually categorized by four main roles: modeler, architect, developer, and administrator.

Keep in mind that the roles do not necessarily correspond to individuals within the organization, but instead refer to tasks and responsibilities, which might overlap for certain individuals. So the same person might have several roles.

The one business user role that is defined for Decision Server Insights is the Modeler role, whose tasks would include defining the models that are required for the solution and authoring the rules.

The architect is generally responsible for designing and managing the integration touch points. What does Decision Server Insights need to listen to in terms of incoming events and what outbound events need to go to which systems.

The developer develops tests and employs the solutions, including entity models, event definitions, and agents.

The administrator installs, configures, and maintains all the tools across the development staging and production environments.

1.5. Installing Decision Server Insights

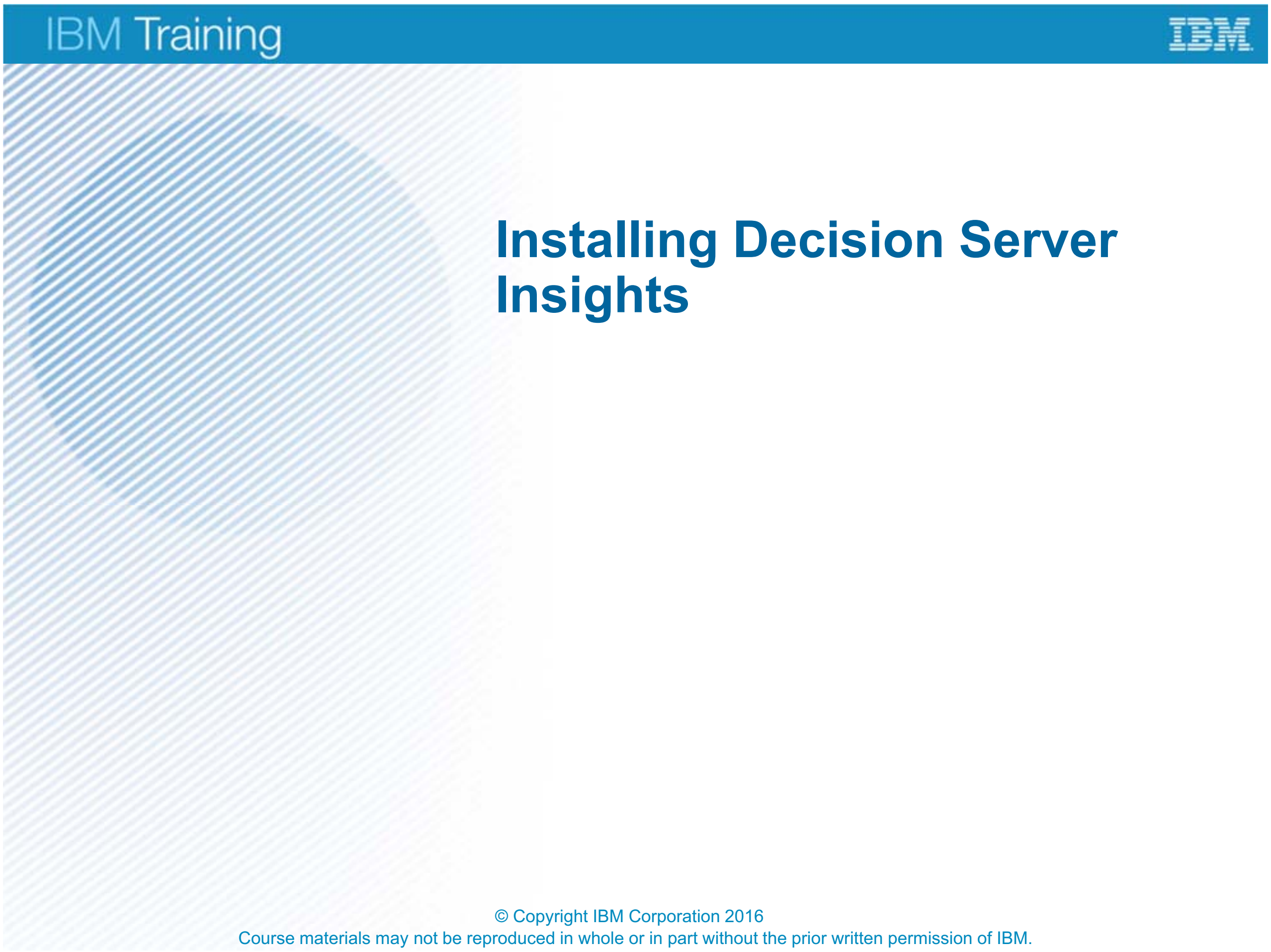


Figure 1-22. Installing Decision Server Insights

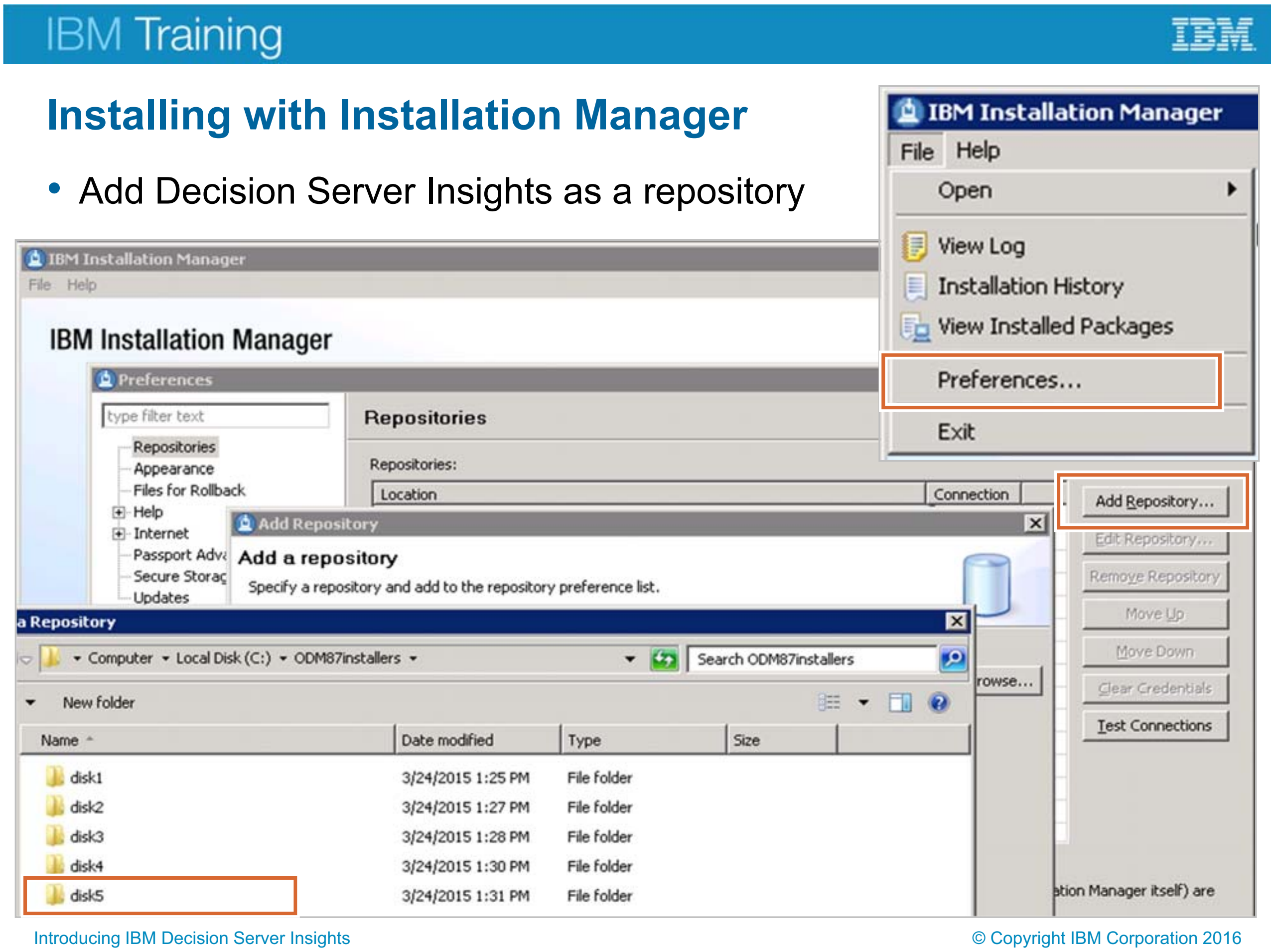


Figure 1-23. Installing with Installation Manager

Decision Server Insights is installed separately from Operational Decision Manager. You must install it in a separate *package group* or installation directory.

The Decision Server Insights installation files are provided in the **disk5** folder. In Installation Manager, you add a repository to point to the installation files as a repository. You must also install the prerequisite software, including WebSphere Application Server, before installing Decision Server Insights.

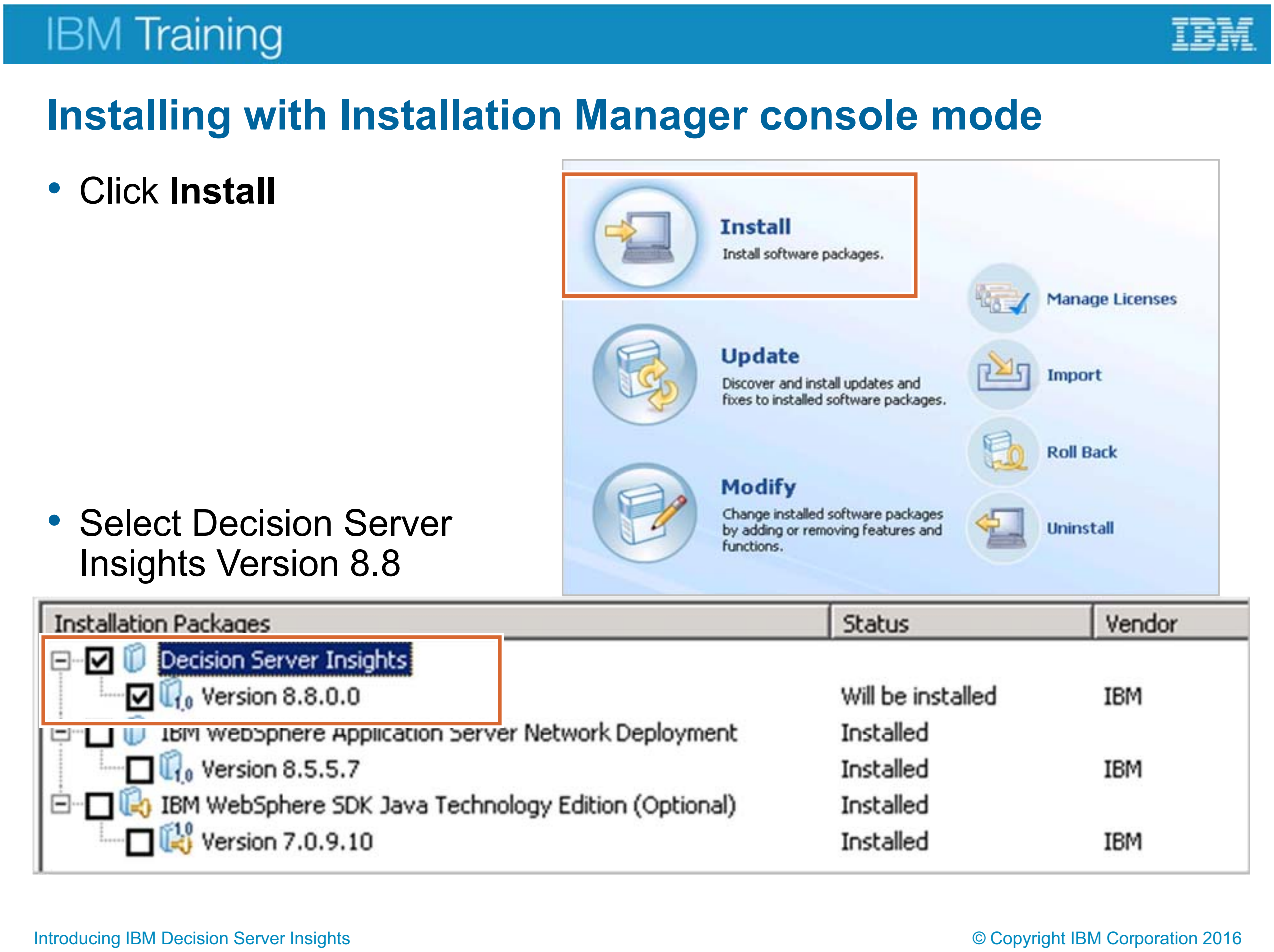


Figure 1-24. Installing with Installation Manager console mode

After the installation repository is accessible to Installation Manager, you can use the Installation Manager console to install Decision Server Insights.

IBM Training

Installing: Defining the installation path

- To avoid conflicts with Operational Decision Manager, choose a separate installation directory
  - Do not use “Program Files” or “Program Files (x86)”

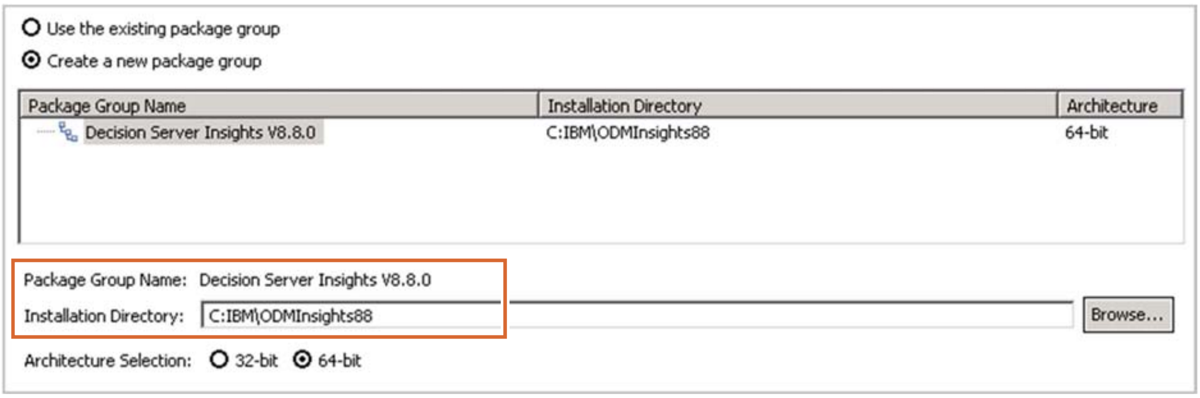


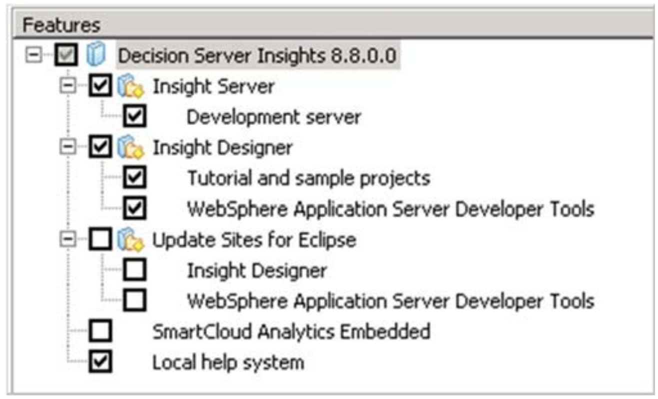
Figure 1-25. Installing: Defining the installation path

To avoid conflicts with Operational Decision Manager, you use a separate installation path, such as the one you see here, to install Decision Server Insights.

IBM Training

Installing: Choosing the product features

- Installing for development
  - Keep the default selection of features
  - If Eclipse is already installed, clear **Update Sites for Eclipse**



- When installing for production, you install Insight Server only

Introducing IBM Decision Server Insights

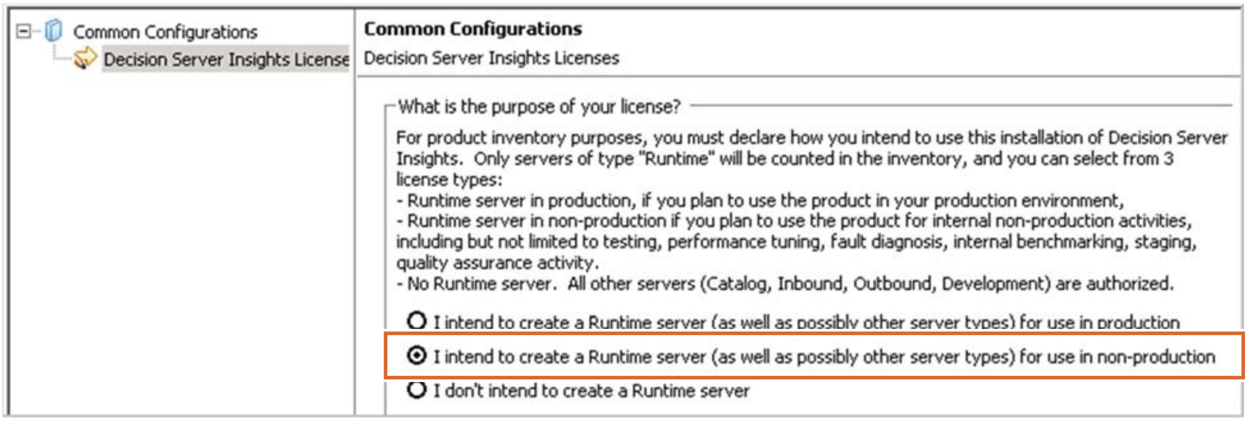
Figure 1-26. Installing: Choosing the product features

When you install Decision Server Insights for development, you can keep the default selection of features. When you install for production, you do not need to install Insight Designer. You install Insight Server only.

IBM Training

Installing: Choosing the installation type

- Choose the installation type that is supported by your license
  - During the exercises, you install for non-production



Introducing IBM Decision Server Insights

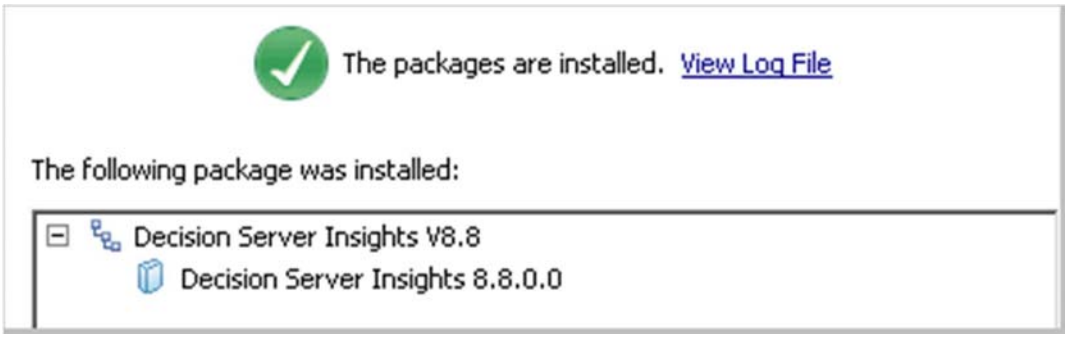
Figure 1-27. Installing: Choosing the installation type

You choose the installation type that is supported by your license. For example, if you want to install multiple server types to develop and test solutions in a grid environment, you choose the second option: `I intend to create a Runtime server (as well as possibly other server types) for use in non-production`

IBM Training

Installing: Summary

- A summary window lists your installation choices before you install
- After installation completes, you see a confirmation message that Decision Server Insights was installed



Introducing IBM Decision Server Insights

Figure 1-28. Installing: Summary

After installation completes, you see a confirmation message that Decision Server Insights is installed.

IBM Training

Installing Decision Server Insights on multiple servers

- Use the `CIS_Silent.xml` template for silent installation to install the product features on multiple workstations
  - Provided in the ***InstallDir*\doc\silent** directory
  - Must install Decision Server Insights on a workstation to obtain the Decision template
- Edit placeholders (delimited with "!") with actual values

Parameter	Description
!CIS_REPOSITORY!	Location of Decision Server Insights installation files
!CIS_HOME!	Installation path for Decision Server Insights

Introducing IBM Decision Server Insights

Figure 1-29. Installing Decision Server Insights on multiple servers

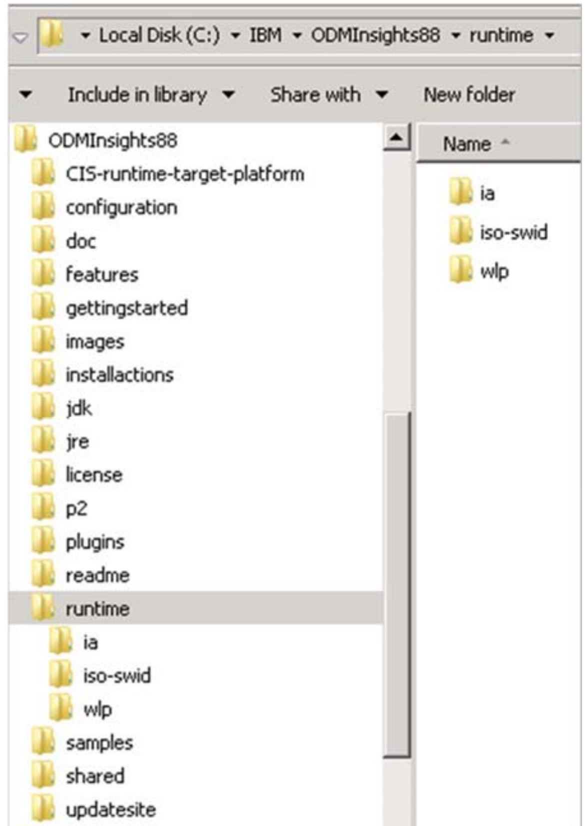
You can also run silent installation of Decision Server Insights by using the template that is provided with Decision Server Insights.

The template includes place holders for the location of the installation files and the target installation path. During the exercises, you install Decision Server Insights on multiple hosts.

IBM Training

Verifying your installation

- Development environment
  - Full Decision Server Insights installation
  - Includes Insight Designer and sample cisDev server
- Production environment
  - Install only Insight Server on multiple hosts
- Runtime tools installed in `<InstallDir>/runtime` directory
- Production servers created in the WebSphere Liberty Profile (`wlp`) folder: `runtime/wlp/usr/servers`



Introducing IBM Decision Server Insights

Figure 1-30. Verifying your installation

To verify your installation, you open the target installation directory that you specified.

During the exercises, you work with the runtime tools that are installed in the **runtime** folder.

IBM Training

Unit summary

- Describe Decision Server Insights and explain how it works
- Explain the programming model
- Describe the Decision Server Insights architecture
- Outline the user roles that are associated with Decision Server Insights

Introducing IBM Decision Server Insights

Figure 1-31. Unit summary

IBM Training

Review questions

- True or false: Global entity aggregates provide the ability to run calculations across your entire population of entities or a subset of those entities.
- The development component of Decision Server Insights is an Eclipse plug-in that is called:
  a. Insight Studio
  b. Insight Designer
  c. Event Designer
- The Decision Server Insights programming model uses which types of agents? Select all that apply:
  a. Rule agent
  b. Java agent
  c. Predictive scoring agent

Introducing IBM Decision Server Insights

Figure 1-32. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers (1 of 2)

- True or false: Global entity aggregates provide the ability to run calculations across your entire population of entities or a subset of those entities.
  Answer: True

- The development component of Decision Server Insights is an Eclipse plug-in that is called
  a. Insight Studio
  b. Insight Designer
  c. Event Designer
  Answer: b. Insight Designer

Introducing IBM Decision Server Insights

Figure 1-33. Review answers (1 of 2)

IBM Training

Review answers (2 of 2)

- The Decision Server Insights programming model uses which types of agents? Select all that apply:
  a. Rule agent
  b. Java agent
  c. Predictive scoring agent
  d. Secret agent
  Answer: All (rule agent, Java agent, predictive scoring agent)

Figure 1-34. Review answers (2 of 2)

IBM Training

Exercise: Getting started with Decision Server Insights

Introducing IBM Decision Server Insights

Figure 1-35. Exercise: Getting started with Decision Server Insights

IBM Training

Exercise objectives

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

Introducing IBM Decision Server Insights

Figure 1-36. Exercise introduction

Unit 2. Designing Decision Server Insights solutions

Estimated time

01:30

Overview

This unit teaches you how to plan and design a Decision Server Insights solution.

How you will check your progress

- Review
- Exercise

IBM Training

Unit objectives

- Model a solution
- Outline design factors
- Describe the solution project

Designing Decision Server Insights solutions

Figure 2-1. Unit objectives

IBM Training

Topics

- Design considerations
- Planning for development and test environments
- Creating solutions

Designing Decision Server Insights solutions

Figure 2-2. Topics

2.1. Designing a solution

Figure 2-3. Designing a solution

IBM Training

Steps to developing a solution

1. Identify your situation
2. Model the entities and events
3. Create agents
4. Detect you situation
5. Write rules

Figure 2-4. Steps to developing a solution

To ensure that the solution correctly addresses your needs, so it is important to take the right approach and analyze your business situation thoroughly before you start your project.

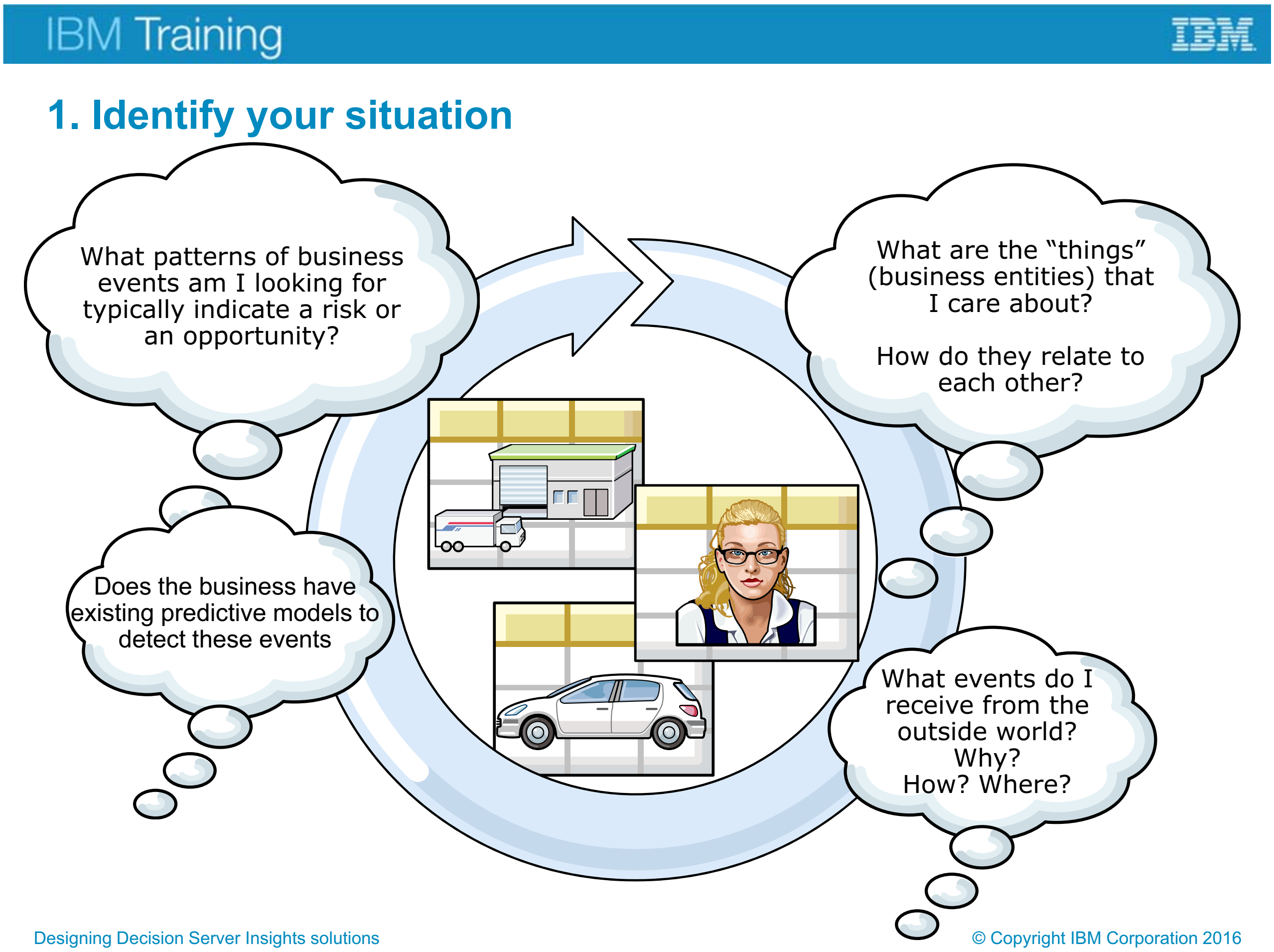


Figure 2-5. 1. Identify your situation

To get started on a project, you begin by documenting the business struggles that you want to respond to, and what situations you are already using to affect your business results, internal productivity, efficiency, and agility. Clarify which business events or transactions you can analyze over time to detect patterns, trends, or interactions that the business is currently missing. Then, analyze what the business already knows about the events that you want to detect, and which rules or predictive models are used to detect them.

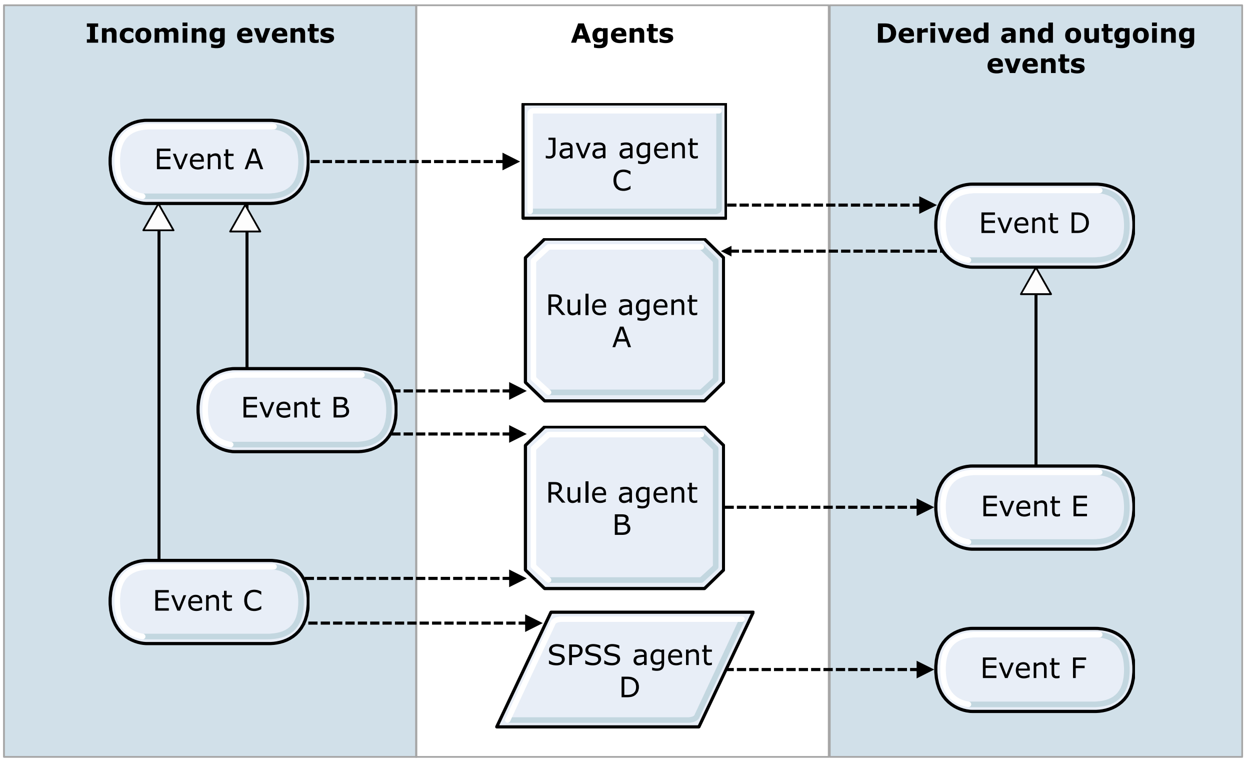
Before you implement a Decision Server Insights solution, the architect and modeler should sketch out an outline of the entities, events, and relationships that must be accounted for in the solution. You must consider which entities are relevant. These entities are the things that have a long lifecycle within the business domain. The events and event patterns are interesting typically have some lifecycle implication on the entity.

Based on that type of information, you can determine the lifecycle for your entities, which types of agents should be bound to those entities, and how the agent should update the entity.

To model this type of information, you can use solution diagrams, as described next.

IBM Training

2. Model entities and events: Solution diagrams



Designing Decision Server Insights solutions

Figure 2-6. 2. Model entities and events: Solution diagrams

You might have existing models, such as class diagrams or schema files, for your domain. Or, you can use modeling tools, such as UML diagrams to model the entities, events, relationships, and temporal and spatial logic.

This slide shows an example of the notation that you can use to model agents, events, and their relationships within a solution.

Notice the various elements that are used as notation in the diagram. These elements include the columns, or swimlanes, to distinguish between incoming events, agents, and outbound events.

The elements also include various shapes that are used to differentiate between rule agents, Java agents, or predictive scoring.

Also, notice that solid lines with arrows represent inheritance relationships, while dashed lines show which events the agents are listening to and which events the agents emit.

This type of overview diagram is to show just the logical view of event processing without physical information about the inbound and outbound endpoints.

IBM Training

Agent diagrams

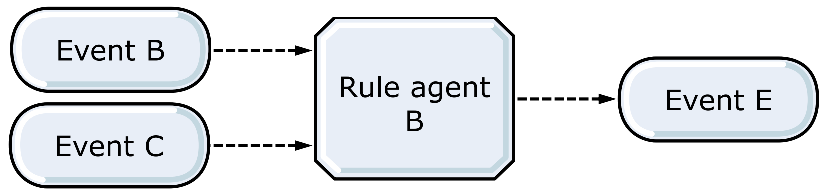


Figure 2-7. Agent diagrams

This slide shows an example of an agent diagram that you can use to focus on a single agent, and which events the agent listens to and which events it emits.

The events that the agent is subscribed to are on the left. Emitted events are on the right.

No inheritance relationships are depicted between the events as the purpose of this type of diagram is to focus on the agent.

By modeling your agents, you can target which events trigger updates to the entity that this agent is bound to.

As a good practice, when you name your agents, include the name of the bound entity in the agent name.

For example, an agent that is bound to a client entity might include “Client Agent” as part of the name. Multiple agents can be bound to the same entity, so use meaningful names for your agents.

IBM Training

3. Create agents

- Agents update entities when events or event patterns are detected
  - One agent can process multiple events
  - Multiple agents can be bound to a single entity
- Rule agents are the default choice of agent
  - Store the state of static fields in the working memory
  - This data is available in a stateful way
- Java agents are stateless by default
  - To maintain state in a Java agent, you must add attributes to the bound entity
- General guidelines can also be used to make your choice of agent:
  - If you need event aggregation in your solution, use a rule agent because the business model language makes these operations much simpler
  - Use a predictive agent, which is a Java agent that you use when you integrate with an external service

Designing Decision Server Insights solutions

Figure 2-8. 3. Create agents

In a common business situation, you are likely to need one or more event types, aggregates over time, and an update of the entity state when the situation occurs. When you identify which entities require updates when an event or event pattern occurs, you know that an agent is needed to process the events. An agent can process one or more event types. You can also have multiple agents that act on a single entity, but for different purposes.

When choosing an agent type, in most cases, a rule agent is likely to be your first choice of agent. In a rule agent, you can express rule-based logic in a high-level rule language. You use rule agents for decisions that are based on the business logic that you want to expose to business users. Because the logic is subject to frequent changes, you want the business stakeholders to validate the changes and even make the changes themselves.

Possible reasons for using a Java agent instead of a rule agent:

- To access the system in some way that is not possible in rules. For example, if you want to call a web service. (You can also use an OSGi service for this purpose.) Notice that introducing latency in this way might have severe performance impacts on high-throughput systems.
- To calculate some complex computation (complicated loops, data handling, Java library calls), which is difficult to implement in rules.
- To do something simple, and you are more comfortable writing Java code.

- To write some logic that does not change often and this logic exists in Java.
- For preprocessing or post-processing. For example, converting an inbound event that has some mismatch between the real world and your model by implementing Java code. An example of post-processing is to translate a displayed message to another language before sending the message to the outside world.

How do you choose which type of agent to use in your solution?

You can use all three types of agents together in a solution. Each agent can monitor different entities or the same entity, subscribe to events and emit events.

By default, you ***want*** to use rule agents. These agents make the decisions. Rule agents express the business logic according to the rules that the business stakeholders write.

For example, you can use a rule agent to respond to the arrival of a new event, such as an ATM withdrawal in New York. You can apply business rules to recognize particular combinations of events and entity states, such as a sequence of withdrawals in different cities, but on the same day.

The business logic is subject to frequent changes, so it must be exposed to the business stakeholders for them to validate the implementation, and even make changes themselves.

Rule agents use the ODM rule authoring technology so that the business language is expressed in high-level natural language.

If you need to write some code to do additional processing, such as data enrichment, you can use Java agents.

You can use predictive agents to invoke SPSS models and update predictive scores.

It is important to note that all decision making should be done by the rule agents. Do not include business logic within the Java agents or predictive scoring agents.

IBM Training

4. Detect your situation

- Consider the sequences of events and the state of entities to test correct pattern or situation detection
  - Use cases
  - Insight Designer test tools

Designing Decision Server Insights solutions

Figure 2-9. 4. Detect your situation

IBM Training

5. Write rules

- Rules define what action to take when an agent receives an event
- Rule vocabulary comes from the elements that are defined in the business model and the agent descriptor
- As you develop your business logic, you might discover ambiguities in the definition of the situation
  - Ongoing collaboration between project stakeholders required
- Consider these questions and how they would be implemented in the rules:
  - What happens when the situation is moved to a different time zone like EST, GMT, CET?
  - What time zone does your solution work in?
  - What happens when you receive an event that is late?
  - What if an event never receives the stop event for a start event that is received?
  - What can you do when you receive events out of logical order?

Designing Decision Server Insights solutions

Figure 2-10. 5. Write rules

Rules define the action to be taken when an agent receives an event. The vocabulary available in the rules comes from the elements that are defined in the business model and the agent descriptor. A rule can be expressed by using business terms, variables, operators, and values.

As you develop your business logic, it is not unusual that you discover hidden ambiguities in the definition of the situation and these tests need to be implemented in rules. As you seek answers to the following questions, the number of rules are likely to increase and provoke further conversations with the stakeholders.

IBM Training

Planning for event delivery

- Model connectivity for how the events are delivered to and from the solution.
  - Where will the events come from?
  - What integration and connectivity infrastructure is required to receive the incoming events?
  - Will the incoming events need to be modified or transformed so that the Insight Server runtime can understand them?
  - What about outbound events?
  - What external systems need to be notified by agents?
- Solution gateway API
- Connectivity
  - Inbound
  - Outbound

Designing Decision Server Insights solutions

Figure 2-11. Planning for event delivery

When you design your solution, you must also model connectivity for how the events are delivered to and from the solution.

- Where will the events come from?
- What integration and connectivity infrastructure is required to receive the incoming events?
- Will the incoming events need to be modified or transformed so that the Insight Server runtime can understand them?
- What about outbound events? What external systems need to be notified by agents?

You need to plan for these questions during the design phase.

IBM Training

Understanding "time" in Decision Server Insights

- In Decision Server Insights, every event is assigned a time stamp
- Be familiar with these concepts of time

Time units	Seconds, minutes, hours, days, weeks, months, years
Time scales	Sequence of time points that are defined by time units
Time points	An instant of time that is measured in a time scale where the smallest unit is the second
Time zones	Decision Server Insights uses time operations that are based on the Gregorian calendar Time points for the same event at the same instance but in different time zones are different time points By default, the Decision Server Insights solution uses the time zone of the server where it is deployed
Durations	The distance between two time points Example: 5 years (start point: December 3, 2014, 1:30:00 PM; end point: December 2, 2019, 1:30:00 PM)
Time periods	An interval that is defined by a start time and an end time Examples: today, this month, 2 hours before departure, the year period between the start of 2000 and the start of 2010

Designing Decision Server Insights solutions

Figure 2-12. Understanding "time" in Decision Server Insights

The concept of time is paramount to the behavior and operations of Decision Server Insights. You must understand the units of time, what a time point is, and how you can use periods and durations to refine which events you act upon.

IBM Training

Location-based reasoning

- Evaluate geographical locations or measure distances
- Static geometries
  - The location of an entity or an event that does not move over time
- Moving geometries
  - The location of an entity that moves over time and has a location at a specific time
- Objects that define geographic locations are represented by these spatial geometries:
  - Geometry
  - Point
  - Line string
  - Linear ring
  - Vertex
  - Polygon
- Composed geometries
  - A group of one or more geometries

Designing Decision Server Insights solutions

Figure 2-13. Location-based reasoning

Decision Server Insights supports the handling of geographic, geometric, and topological data.

In an agent, you can write operations on entities and events that have geographic locations. For example, you can update the location information of an entity, calculate the distance between two points, or send alerts based on the location of an entity.

Before you use geospatial computations in your solution, you must be aware that in special cases the results might be imprecise. Imprecisions in geospatial positioning are caused by the following factors.

- Shape of the Earth
- Computation of floating points

2.2. Planning for development and test environments

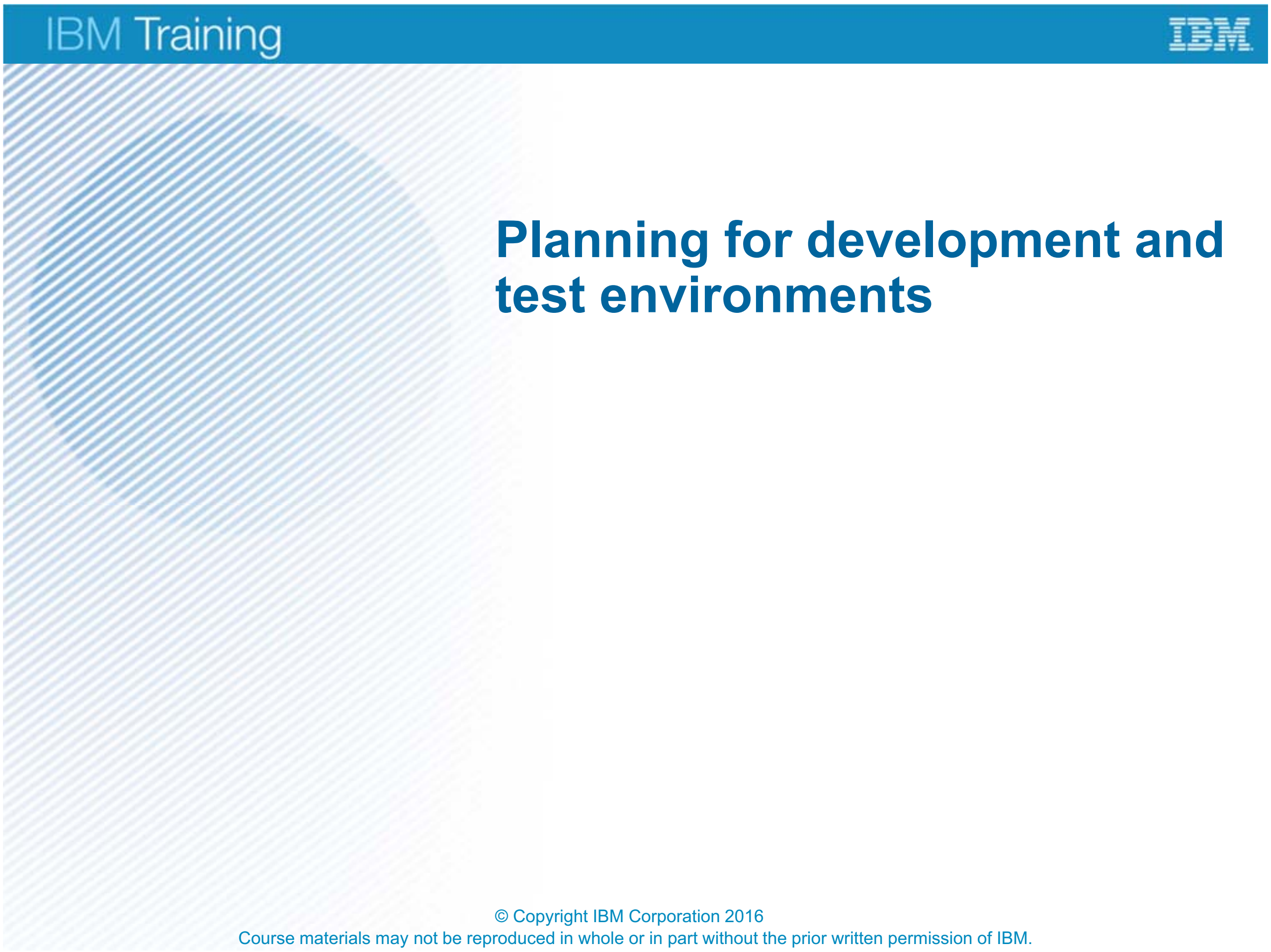


Figure 2-14. Planning for development and test environments

IBM Training

Setting up the development and test environments

- Decision Server Insights relies upon WebSphere Liberty profile and WebSphere eXtreme Scale for its runtime environment
  - Integration with WebSphere eXtreme Scale is paramount to the overall performance
  - In the design phase, you need to define the integration points of your solution
  - Become familiar with WebSphere eXtreme Scale configuration settings that affect capacity and performance
- During the design phase:
  - Document the solution architecture and the functional specification that is related to the data level
  - For security considerations, specify the location of the Java keystore
  - Include anticipated metrics to assess the performance goals

Designing Decision Server Insights solutions

Figure 2-15. Setting up the development and test environments

Before you start developing a solution, youwant to make sure that you set up your development and testing environments properly to ensure productivity and mitigate risk.

In the design phase, you need to define the integration points of your solution, document the solution architecture, and functional specifications, along with metrics to assess performance goals.

For example, the design might include metrics for transaction speed performance requirements before deployment, and the architecture might be a series of block diagrams that show networks, service components, and network connection elements.

You should also become familiar with the extreme scale configuration settings that might affect capacity and performance.

Then, you can set up the hardware and other infrastructure resources that are required for development and testing activities.

IBM Training

Design concerns

- Data capacity
  - Consider how many physical machines and processors you need and how many container servers and partitions are needed to host your object maps
  - How much data you plan to store on each server
  - The amount of data that is stored in the maps is one factor in determining the appropriate number of partitions
- Caching topologies
  - Data persistence can be configured to write system data to a back-end database, either synchronously or asynchronously
- Connectivity
  - All inbound and outbound binding is defined in the solution connectivity definition so you must know all of the sending and receiving endpoints and the type of binding that the messages require
- Global aggregation
  - A solution that uses global aggregates can require large amounts of memory and computer processing capacity

Designing Decision Server Insights solutions

Figure 2-16. Design concerns

Before you begin the development of your solution, you want to set up your development and testing environments. Having these environments properly setup can maximize team productivity and mitigate risk.

Establishing a development environment involves setting up hardware and other infrastructure resources, such as the project structure, that the development activities require. Similar to the development environment, the testing environment must be designed to emulate the production environment as closely as possible.

This slide lists some points to consider during the design phase.

For example, when you plan data capacity, consider how many physical machines and processors you need and how much data you plan to store.

For caching topologies, you need to decide whether data should be written to a backend data store synchronously or asynchronously.

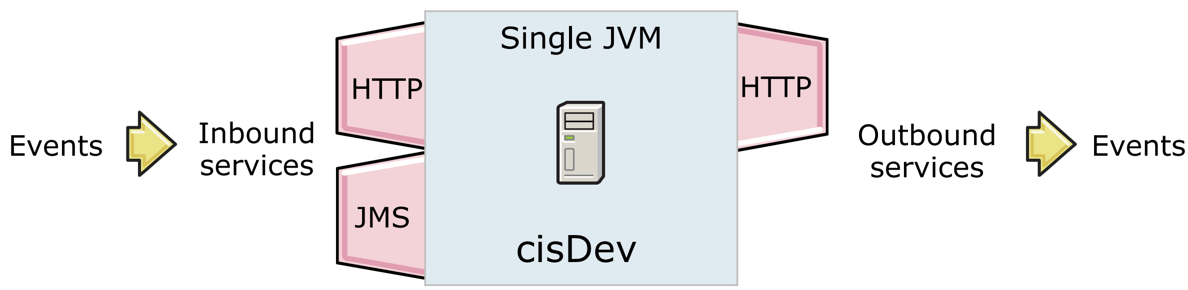
For connectivity, what are all the endpoints that you need to receive and send events to? Will you use both JMS and HTTP for inbound and outbound events, or which endpoints require which type of binding?

Global aggregation can require large amounts of memory and computer processing capacity, so the frequency of global aggregate calculation needs to be considered as part of the design.

IBM Training

Planning development and production environments

- Development and test environments should emulate your production environment



Designing Decision Server Insights solutions

Figure 2-17. Planning development and production environments

A Decision Server Insights configuration consists of one or more servers that are connected by LAN and WAN network links.

For development, you can set up a single-server topology on a single computer. However, development and test environments should emulate your production environment.

For early development work, you might need to use a Java test client to submit events and test the execution of your solution. However, the final testing environment must use the message providers that you are going to use in production.

For example, you can use both HTTP and JMS for both inbound and outbound events. But, if in production, you expect to receive inbound events over HTTP and JMS, and you choose to use only HTTP to emit outbound events, then your development environment should reflect that.

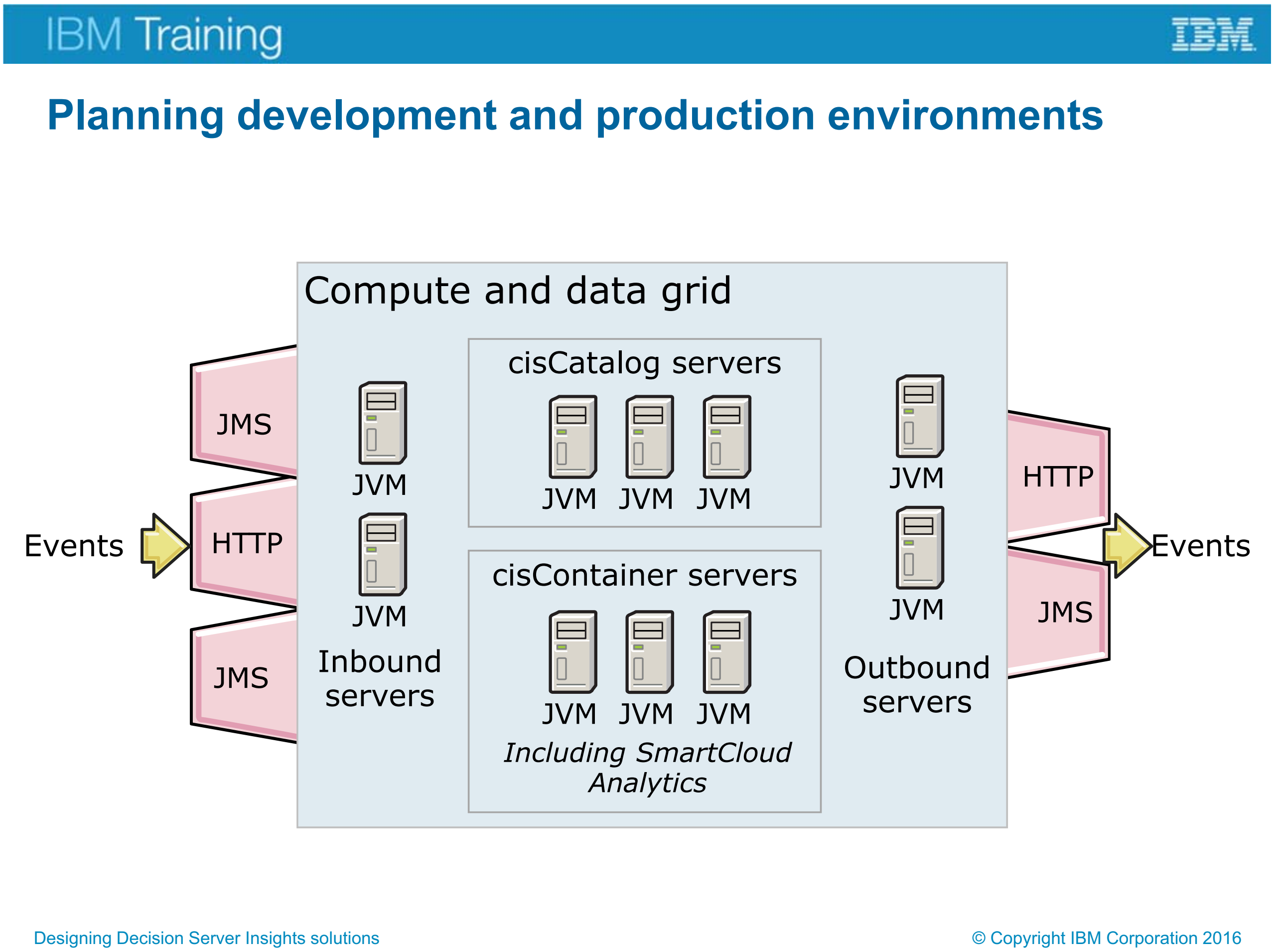


Figure 2-18. Planning development and production environments

Multiple server topology for production is designed to meet availability needs and to ensure minimal downtime if you have a disruption in the IT system. You can also use this topology to add more servers when you need more memory to store the entities.

You use the provided server templates to create the appropriate number of server types with synchronization and backup replicas. Availability and performance can be scaled up horizontally by adding more servers.

2.3. Creating solutions

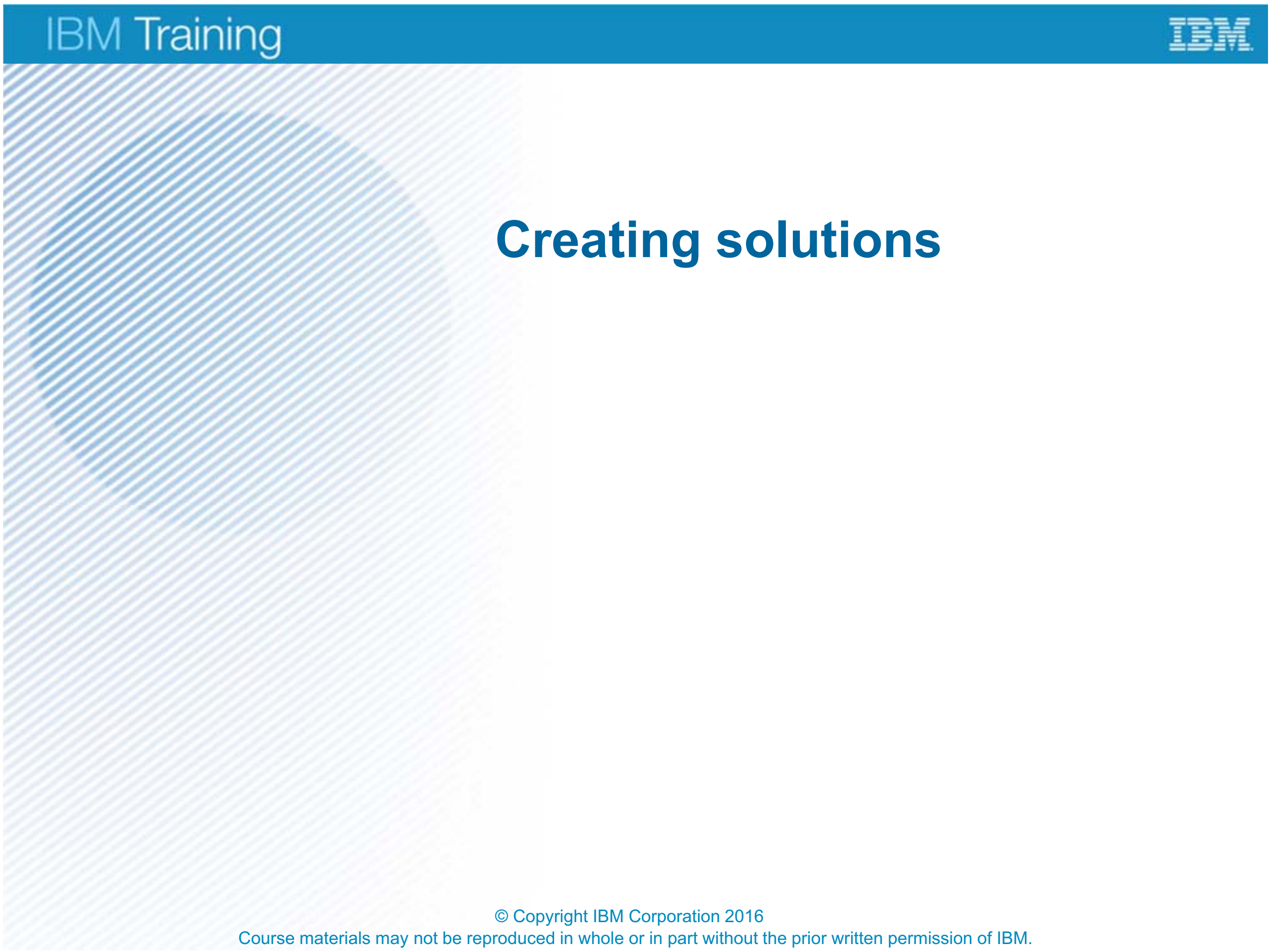


Figure 2-19. Creating solutions

IBM Training

Steps of building a Decision Server Insights solution

- To build your solution:
  1. Create a solution project
  2. Create or import definitions (BMD)
  3. Generate the vocabulary
  4. Set relationships between the objects of the model
  5. Create a rule agent project
  6. Write rule agent business logic
  7. Create an Insight Server (or use the cisDev server)
  8. Deploy the solution
  9. Define connectivity or use the Java API to send an event payload to an Insight Server
  10. Run the application to submit events and see the agents execute

Designing Decision Server Insights solutions

Figure 2-20. Steps of building a Decision Server Insights solution

To start building a Decision Server Insights solution, you work in Insight Design and create a solution project. After you have a solution project, you define your business model in a business model definitions file. In this definition file, you model your entities, events, other concepts, and the relationships between them.

If you already have an XML model for your domain, you can import your event and entity types from that XSD.

The model determines the vocabulary for you to start creating rule agents, and implementing the business logic in rules.

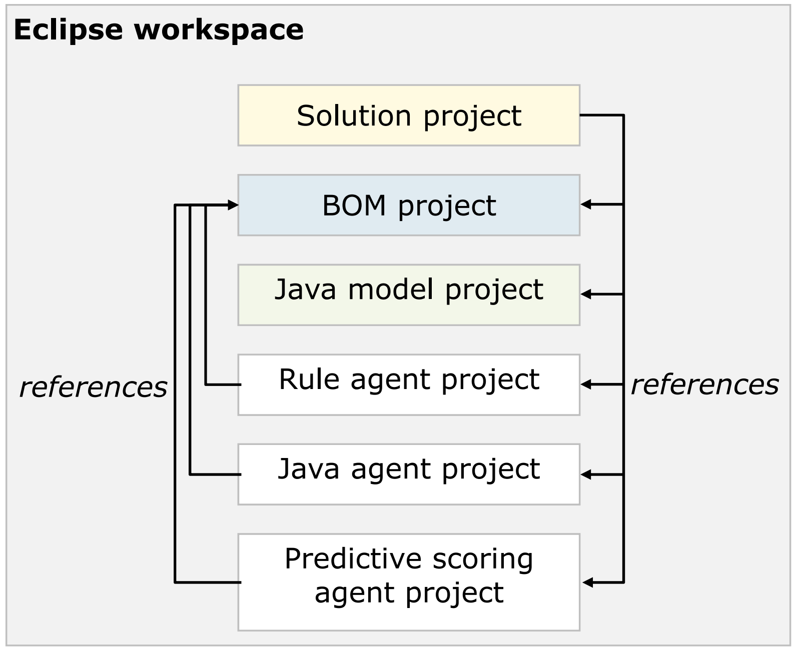
If you are not using the sample cisDev server, you also create an Insight Server. To test your solution, you deploy the solution to the server, and create and initialize your entities. Finally, you submit events to test the agent execution. You use a Java test client or define connectivity to submit events.

You walk through these steps during the labs.

IBM Training

Getting started on an Insights solution

- The solution project is the starting point for adding rules to agents, adding agents to solutions, and adding solutions to the enterprise-wide runtime
- The solution project is the main project of the solution
- The BOM, rules, and other artifacts are stored separately in referenced projects



Designing Decision Server Insights solutions

Figure 2-21. Getting started on an Insights solution

If you are familiar with ODM Rules, you know that the rule project is the container for all rule artifacts in a rule solution. With Decision Server Insights, the solution project contains many, but not all solution artifacts. An insight solution involves several projects that reference each other. This type of container organization makes it easier to manage artifacts.

A complete solution contains the projects that you listed here. The solution project is the main project. When you create a solution, you start off with the solution project, BOM project, and Java model project, which are all generated by default.

The BOM project contains your business model definitions. The Java Model project is a container for the executable object model that is generated from the business object model.

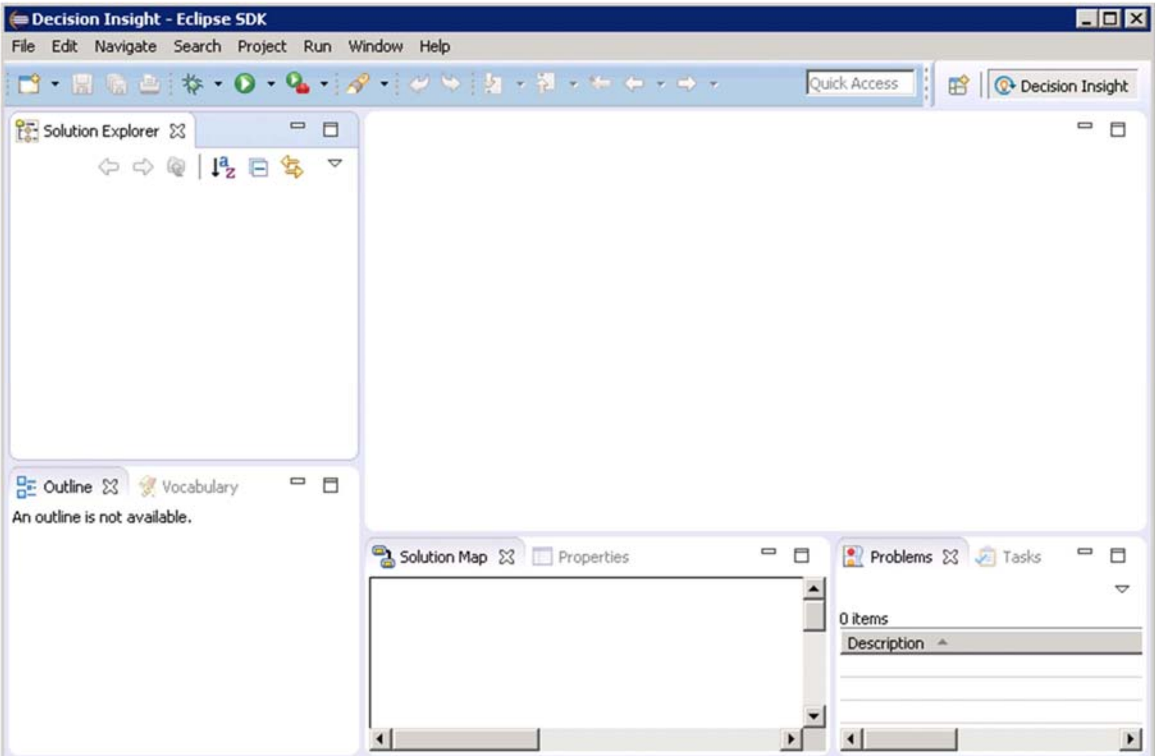
After you add entity and event definitions to the BOM model, you can start creating agent projects. Each agent is stored in a separate project.

Rule agent projects contain the rules that define the business logic. The agent project also contains a descriptor that you must complete to define which entity the agent should monitor and what events the agent should listen for or subscribe to. All agent types, rule agents, Java agents, and predictive scoring agents use descriptors for this purpose.

The **Java agent project** contains Java code. The **Predictive scoring agent project** contains the predictive scoring agent, which calls an external SPSS scoring service.

IBM Training

Decision Insight perspective



Designing Decision Server Insights solutions

Figure 2-22. Decision Insight perspective

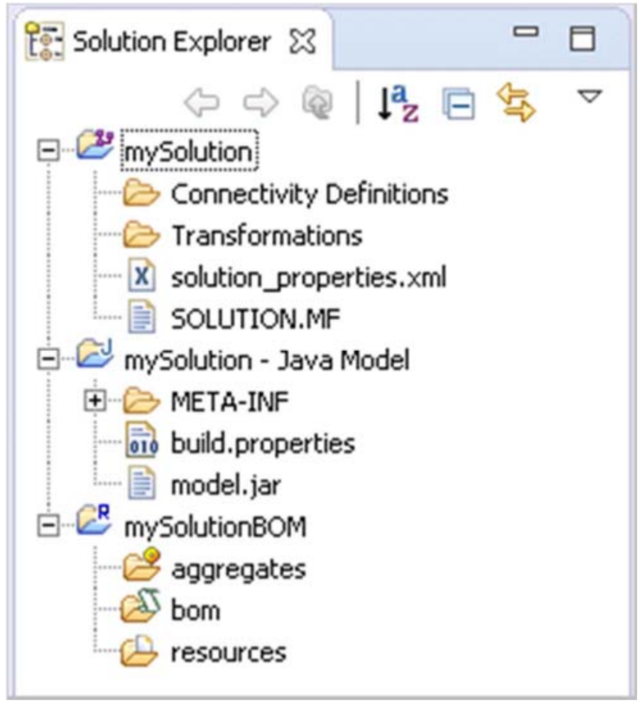
When you create your solution in Insight Designer, you work in the Decision Insight perspective, as you see here.

This perspective contains the menus and tools that you need to create the solution, including the Solution Explorer, which lists any projects that are referenced for the solution.

IBM Training

Solution project folders and files

- When you create a solution, three projects are automatically generated:
  - Solution project
  - BOM project
  - Java Model project



Designing Decision Server Insights solutions

Figure 2-23. Solution project folders and files

As you see in the example, when you create a solution, three projects are generated: the solution project, the BOM project, and the Java model project.

The BOM project that contains the following artifacts:

- **bom** folder: Stores the business model definition files (`.bmd`) where you define in plain English the entities, events, concepts, and others
- **aggregates** folder: Stores the global aggregate definitions (`.agg`)

The solution project folder contains these folders and files:

- **Connectivity Definitions** folder: Stores the connectivity definition (`.cdef`) files that define the inbound and outbound bindings and endpoints for the solution.
- **Transformations** folder: Stores XSL transformation files for converting unrecognized inbound messages.
- **solution_properties.xml** file: Contains the custom properties of the solution project.
- **SOLUTION.MF** file: Contains the version information of the solution, and the solution symbolic name.

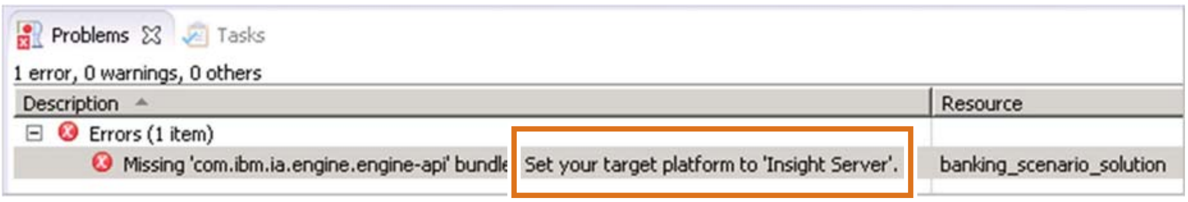
Note

You learn more about the business model and aggregates in later units.

IBM Training

Setting up the workspace

- To compile Java code for Java agents in Insight solutions, you must set Insight Server as the target platform



Designing Decision Server Insights solutions

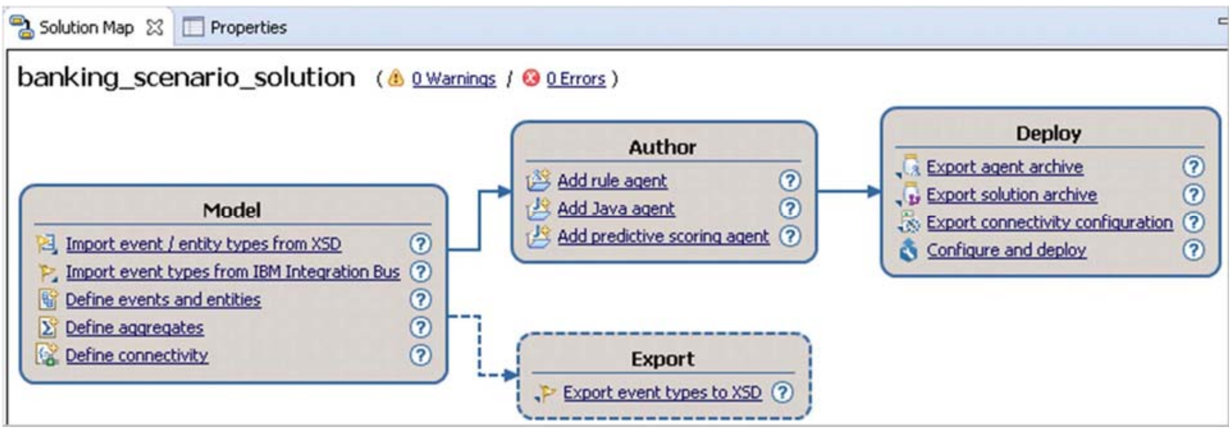
Figure 2-24. Setting up the workspace

When you develop an Insights solution, you need to set the target platform to use the Insight Server. Java agents are plug-in projects, so to compile the Java code for these plug-in projects, you must set the Insight Server target platform, which is a set of OSGi bundles.

IBM Training

Solution Map view in Decision Insight perspective

- Use the Solution Map to guide you through development tasks
- Solution Map tasks are grouped in boxes in the map by their overall goal: Model, Author, Export, and Deploy



Designing Decision Server Insights solutions

Figure 2-25. Solution Map view in Decision Insight perspective

The Decision Insight perspective also includes the Solution Map, which guides you through the tasks of developing a solution.

You start with modeling, which includes the business model, aggregates, and connectivity definitions. You learn more about the Model task later.

The solution map is context-sensitive to each solution that you have in your workspace.

The map displays the solution project name and an accumulated count of the warnings and errors from all of the projects that are referenced by the solution.

When you click a task in the map, Insight Designer opens the relevant dialog box or wizard to start that task. Tasks that appear in gray in the map are disabled because they are not currently applicable to the status of the selected solution.

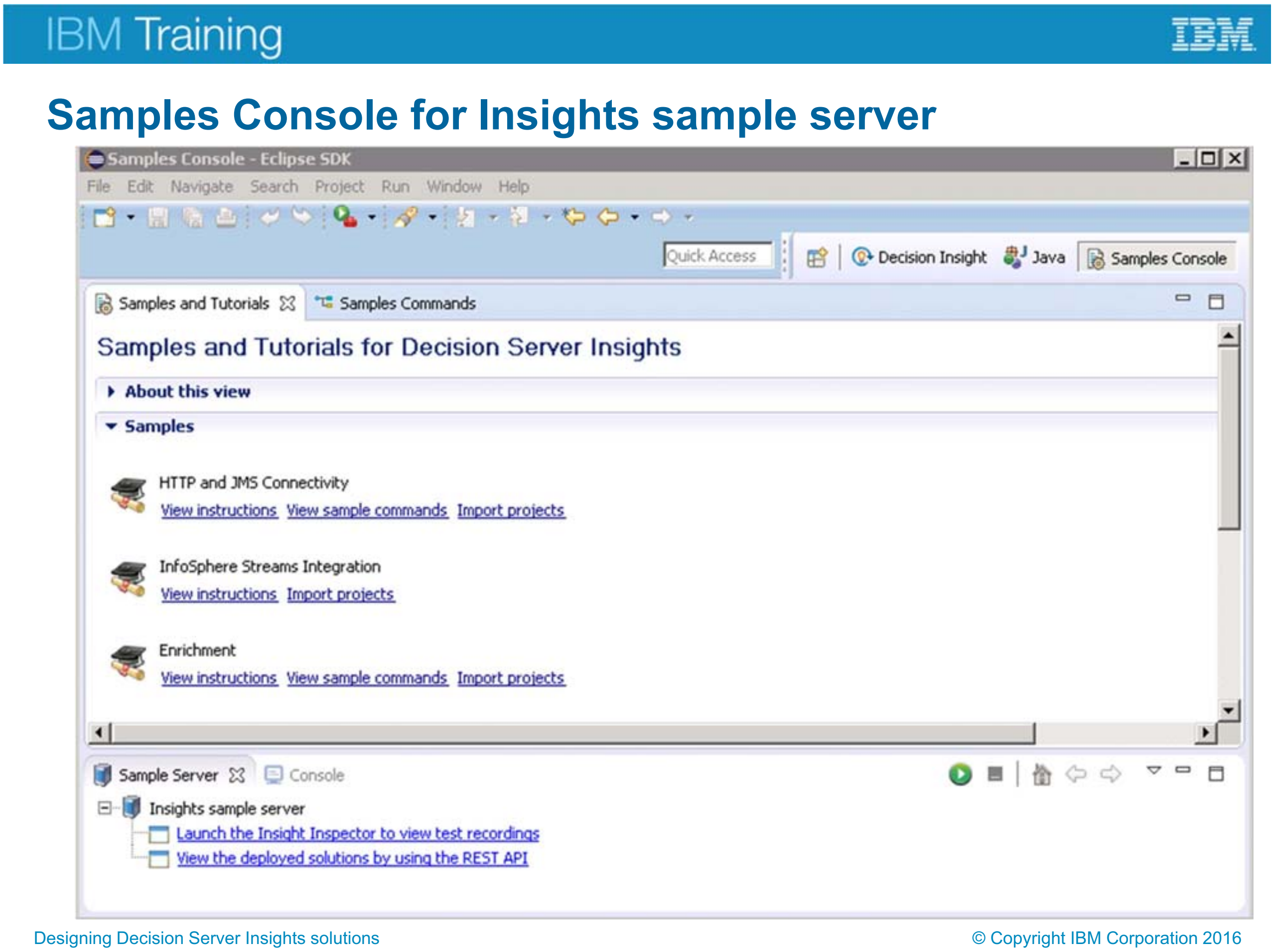


Figure 2-26. Samples Console for Insights sample server

While most of the solution development tasks are done in the Decision Insight Perspective, you also work in the Java perspective and the Samples Console perspective.

For example, to work with the sample server and other tools that are provided by Decision Server Insights, you can use the menus and tools in the Samples console perspective. You can start and stop the Insights sample server from a command prompt window or from the Samples Console. The Insights sample server is called cisDev.

IBM Training

Unit summary

- Model a solution
- Outline design factors
- Describe the solution project

Designing Decision Server Insights solutions

Figure 2-27. Unit summary

IBM Training

Review questions

1. True or false: An outbound event from one agent cannot be an inbound event for another agent.

2. True or false: The spatial geometries feature is built in Decision Server Insights.

3. Solution projects are an Eclipse project that includes references to which of these projects? Select all that apply.
   a. Agent projects
   b. Java test client projects
   c. Business object model project

Designing Decision Server Insights solutions

Figure 2-28. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. True or false: An outbound event from one agent cannot be an inbound event for another agent.
   Answer: False. An agent can consume events that were emitted by other agents.

2. True or false: The spatial geometries feature is built in Decision Server Insights.
   Answer: True.

3. Solution projects are an Eclipse project that includes references to which of these projects? Select all that apply.
   a. Agent projects
   b. Java test client projects
   c. Business object model project

   Answer: a (agent projects) and c (business object model projects).

Designing Decision Server Insights solutions

Figure 2-29. Review answers

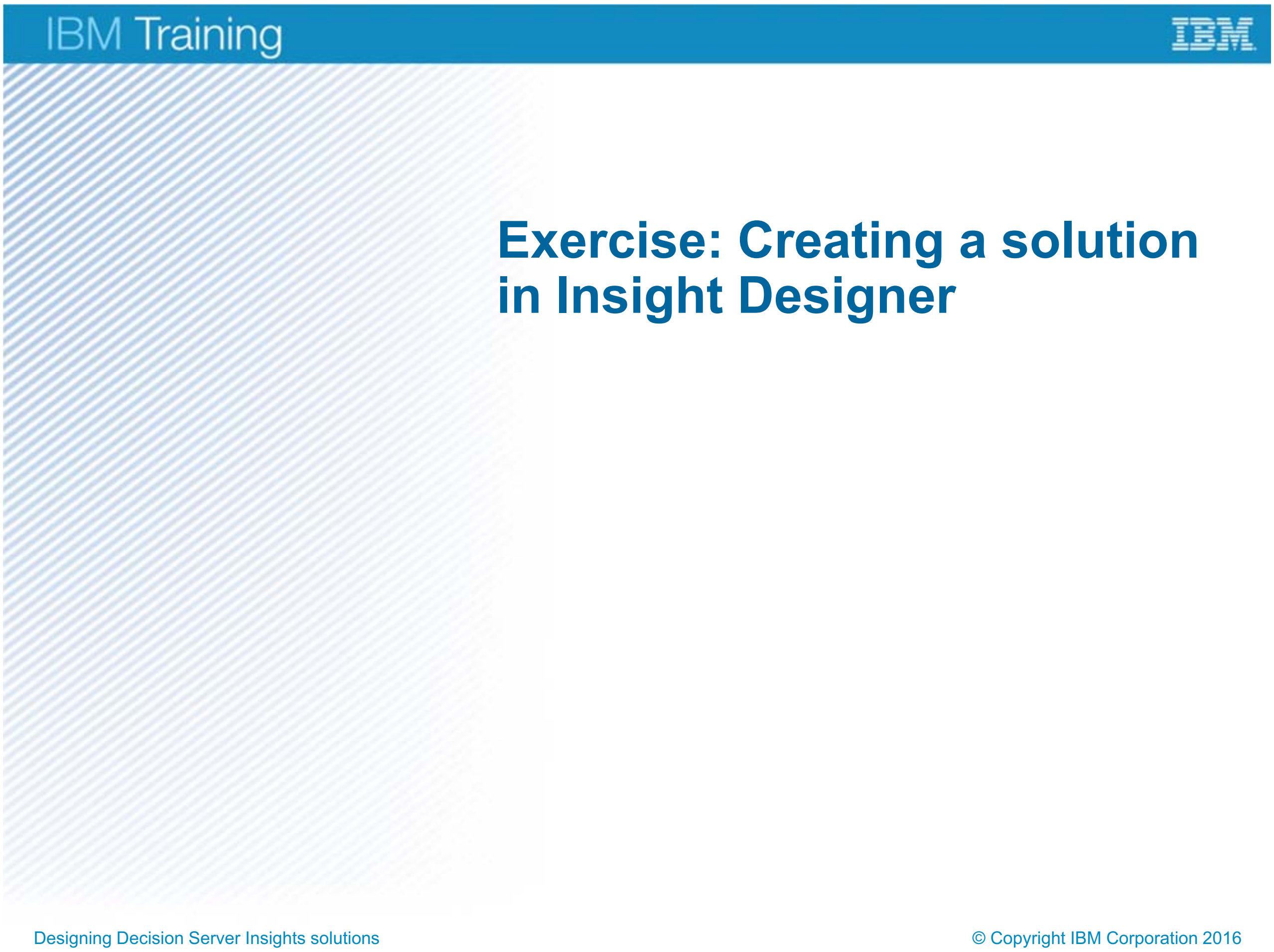


Figure 2-30. Exercise: Creating a solution in Insight Designer

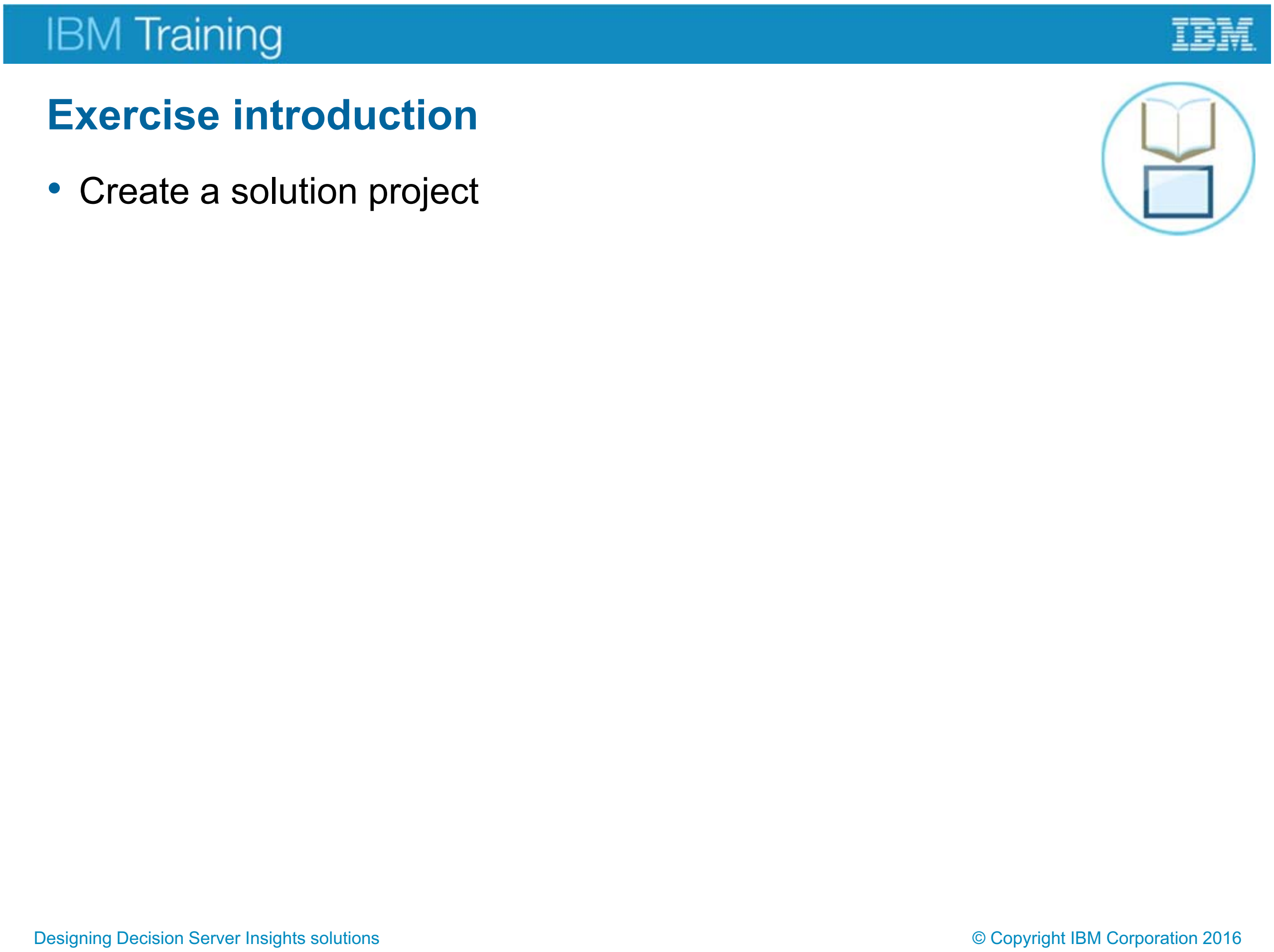


Figure 2-31. Exercise introduction

Unit 3. Creating the business model

Estimated time

01:00

Overview

This unit teaches you how to create the business model definition file.

How you will check your progress

- Review
- Exercise

IBM Training

Unit objectives

- Describe the elements of a business model
- Translate a UML diagram into a business model definition

Creating the business model

Figure 3-1. Unit objectives

IBM Training

Topics

- Overview of the business model
- Writing the business model

Figure 3-2. Topics

3.1. Overview of the business model

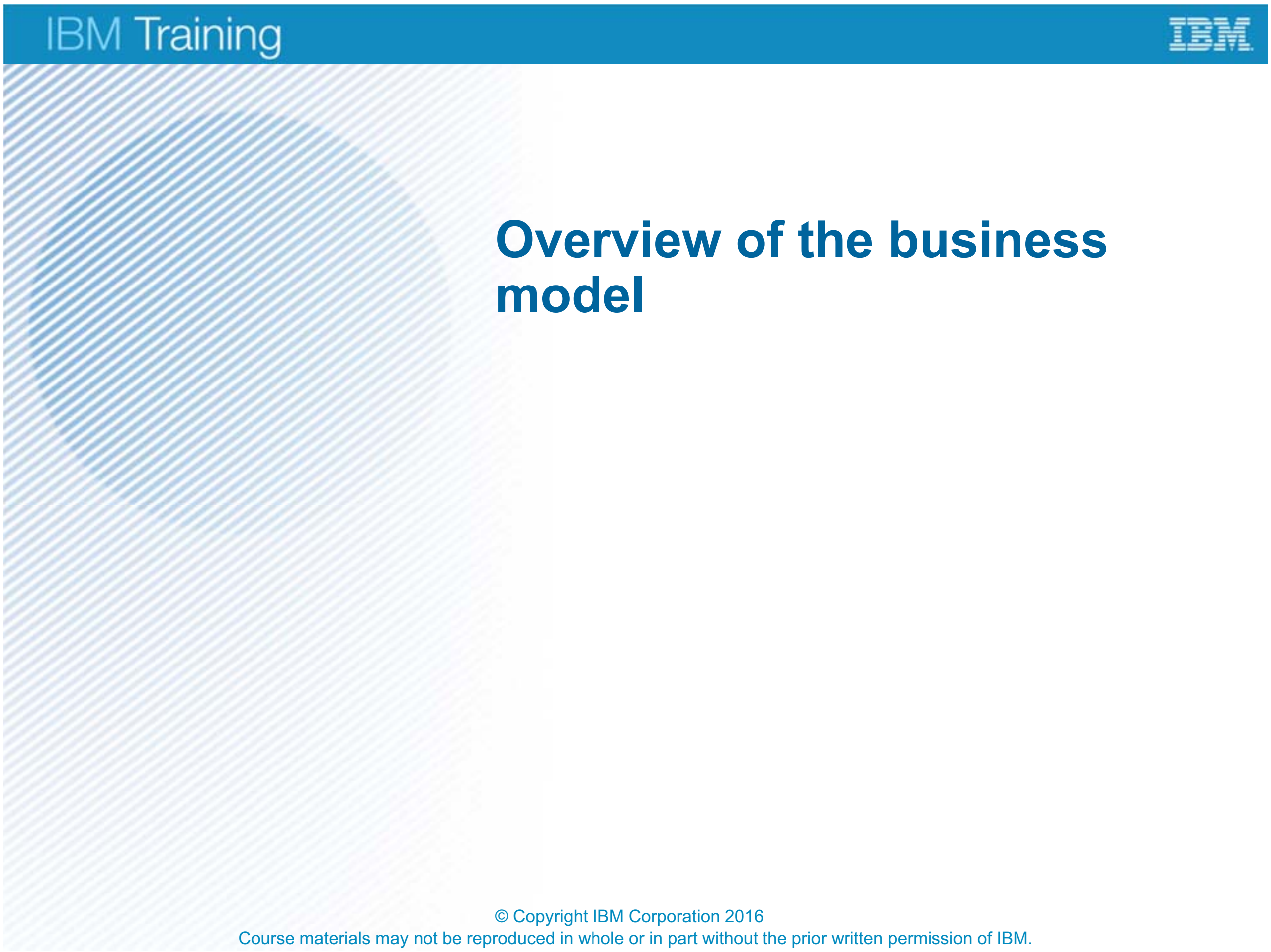


Figure 3-3. Overview of the business model

IBM Training

Modeling for a Decision Server Insights solution

- Before you implement the business logic, you must build the business model
  - Business model definitions determine agent and rule vocabulary
- Business model includes definitions for:
  - Entity types
  - Event types
  - Attributes
  - Relationships
  - Global aggregates
- You can use modeling tools, such as UML diagrams to model entities, events, relationships between them, and temporal logic

Creating the business model

Figure 3-4. Modeling for a Decision Server Insights solution

Modeling the entities and events in your business domain is the first step of developing an insight solution.

The business model includes definitions for entity types, event types, attributes, relationships, global aggregates, and others.

If you used modeling tools, such as UML diagrams to model your entities, events, relationships, or temporal logic, you can use those diagrams as a starting point for your model.

The language that is used in the business model definition is natural language not code. It integrates well with source code control. All the vocabulary that is used by your agents and your business rules comes from the business model definition file.

IBM Training

Elements of a business model

- **Concepts**: A simple data structure, such as an address
- **Entities**: Concepts that have an identifier and a lifecycle
- **Events**: Concepts that have a time stamp and can relate to one or more entities
- **Relationships**: Defines the inheritance or reference to other entities, events, and concepts
- **Enumerations**: Concepts that include a list of possible values
- **Attributes**: An attribute is a characteristic of an entity, event, or concept
- **Derived attributes**: Value is calculated from the value of another attribute
- **Enriched attributes**: Value is supplied by a data provider
- **Data providers**: A service that accepts inputs and that returns outputs
- **Facets**: Location or time attributes that can be added to a concept

Creating the business model

Figure 3-5. Elements of a business model

Here you see a list of elements that you can define in your business model.

For example, a concept is the most basic data structure that you can include in your business model.

Next, you see entities, which are also considered concepts, but they have an identifier and lifecycle. As discussed earlier, they are the things that are important or relevant to your business domain. The events that occur in your domain happen to these entities or affect the entities.

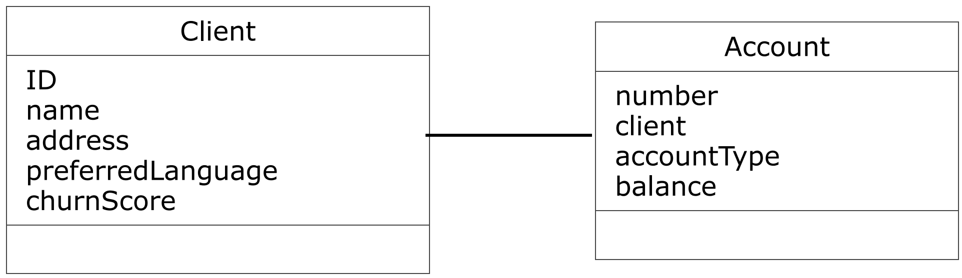
Events are also considered concepts, but they are concepts with a time stamp and they can relate to one or more entities.

For more information about other elements of a business model, see the product documentation for Decision Server Insights.

IBM Training

Example entity model

- UML class diagram for an entity of type Client and of type Account



- Business model definition

a client ***is a business entity identified by*** an ID.

a client ***has a*** name,
an address,
a preferred language,
a churnScore.

an account ***is a business entity identified by*** a number.

an account ***is related to*** a client.

an account ***has*** an account type,
a balance.

Creating the business model

Figure 3-6. Example entity model

Here you see a UML class diagram that is used to model an entity of type *client* and an entity of type *account*.

The entity type is composed of a set of attributes and relationships to other entity types. Each entity has an attribute that acts as an identifier for entity instances.

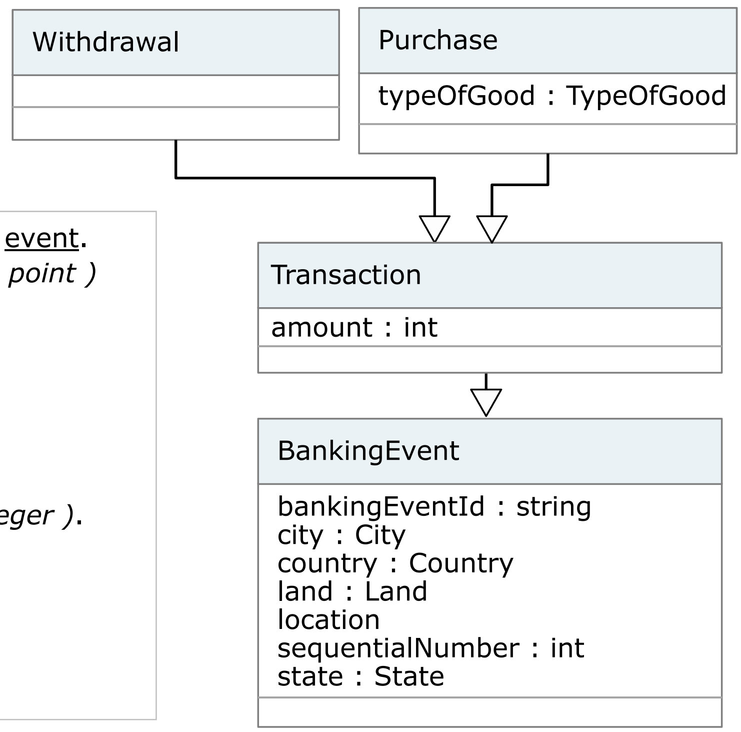
In this example, as you can see by the business model definition, the identifier for client is the *ID*. The identifier for the account is *number*.

Also, in the business model definition, you see that to define the relationship between entities by using the “is related to” construct. In this example for account, an account is related to a client.

IBM Training

Example event model with relationships

- UML diagram of event inheritance relationships



- Business model definition

a banking event ***is a*** client related event.
a banking event ***has a*** *location (a point)*
used as the default *geometry*
a banking event ***has a*** *country*.
a banking event ***has a*** *state*.
a banking event ***has a*** *city*.

a transaction ***is a*** banking event.
a transaction ***has an*** amount *(integer)*.

a withdrawal ***is*** a transaction.

a purchase ***is a*** transaction.
a purchase ***has a*** type of good.

Creating the business model

Figure 3-7. Example event model with relationships

Here, you see a UML diagram of events in the inheritance relationships between events.

An event type describes the shape of an event. It is also composed of a set of attributes, and a set of relationships to other entity types. An event has a time stamp attribute that represents a date and time. An event can also have a location, such as the location of a withdrawal. Decision Server Insights provides special constructs in the rule language to facilitate the handling of location.

As you see in the model definition, you can model relationships between event types. The type of construct that you use determines the type of relationship. For example, the business model defines a **banking event** with its attributes, and the definition for **transaction** is that the transaction is a banking event, and it also has an amount. Also, transaction is in the definition for **withdrawal**: a withdrawal is a transaction.

So here, you see the inheritance relationships that are depicted in the UML diagram, which are translated into the business model definition.

IBM Training

Example enumerations

- Enumerations define a list of possible values

<enumeration> AccountType
Savings: Account
Checking: Account
High_Interest: Account

<enumeration> City
Beijing: City
Kuala_Lumpur: City
New_York: City
Paris: City
San_Francisco: City

<enumeration> TypeOfGood
COMPUTING: TypeOfGood
TRAVEL: TypeOfGood
MUSICAL: TypeOfGood

Creating the business model

Figure 3-8. Example enumerations

On this slide, you see examples of enumerations. Enumerations define a list of possible values.

An entity type like account might have an attribute account type, which is an enumeration and limits the list of possible values.

IBM Training

Facets

- Add a location or time facet to a concept so that you can use time-based or location-based reasoning in your rules

- Location facets
  - Location facets are used for space-driven logic
  - Example:
    `a banking event` ***`has a`*** `location` *`( a point )`* ***`used as the default`*** `geometry.`

- Time facets
  - In Decision Server Insights, every event is assigned a time stamp, or time ***facet*** of type date and time
  - Time facets are used for time-based reasoning
  - Example:
    `a client related event` ***`has a`*** `transaction time` *`( a time )`* ***`used as the default`*** `time stamp.`

Creating the business model

Figure 3-9. Facets

In Decision Server Insights, facets are used when you work with location-based or time-based reasoning.

The term “facet” refers to an aspect or element of something. A “location facet” means an element of location, such as a geometrical point. Or, with the time facet, it refers to an element of time, such as the date or an exact time.

When you define facets in your model, these facets can help you streamline your rules so that the rules are easier to write and read.

For example, consider the banking event definition on this slide, which has “a point” defined as the default geometry. This point is the location facet. With this facet, you can compare the distance between geometrical points where banking events occur.

So when you write the rule about banking events, you do not need to write out the full term:

`the distance between the location point of a BankingEvent1 compared to the location pointed BankingEvent2`

You can write a simpler rule that tests:

`the distance between BankingEvent1 and BankingEvent2 in miles`

The rule language automatically implies a test between geometrical points for these events.

The same principal applies to the time facet that is defined for client-related event definition that is shown here. You can write a rule that tests:

`if ClientRelatedEvent1 is` ***`before`*** `ClientRelatedEvent2`

The language implies that the times for these events are to be compared.

So, the facets are a powerful tool for implied business logic and also improve readability.

3.2. Writing the business model

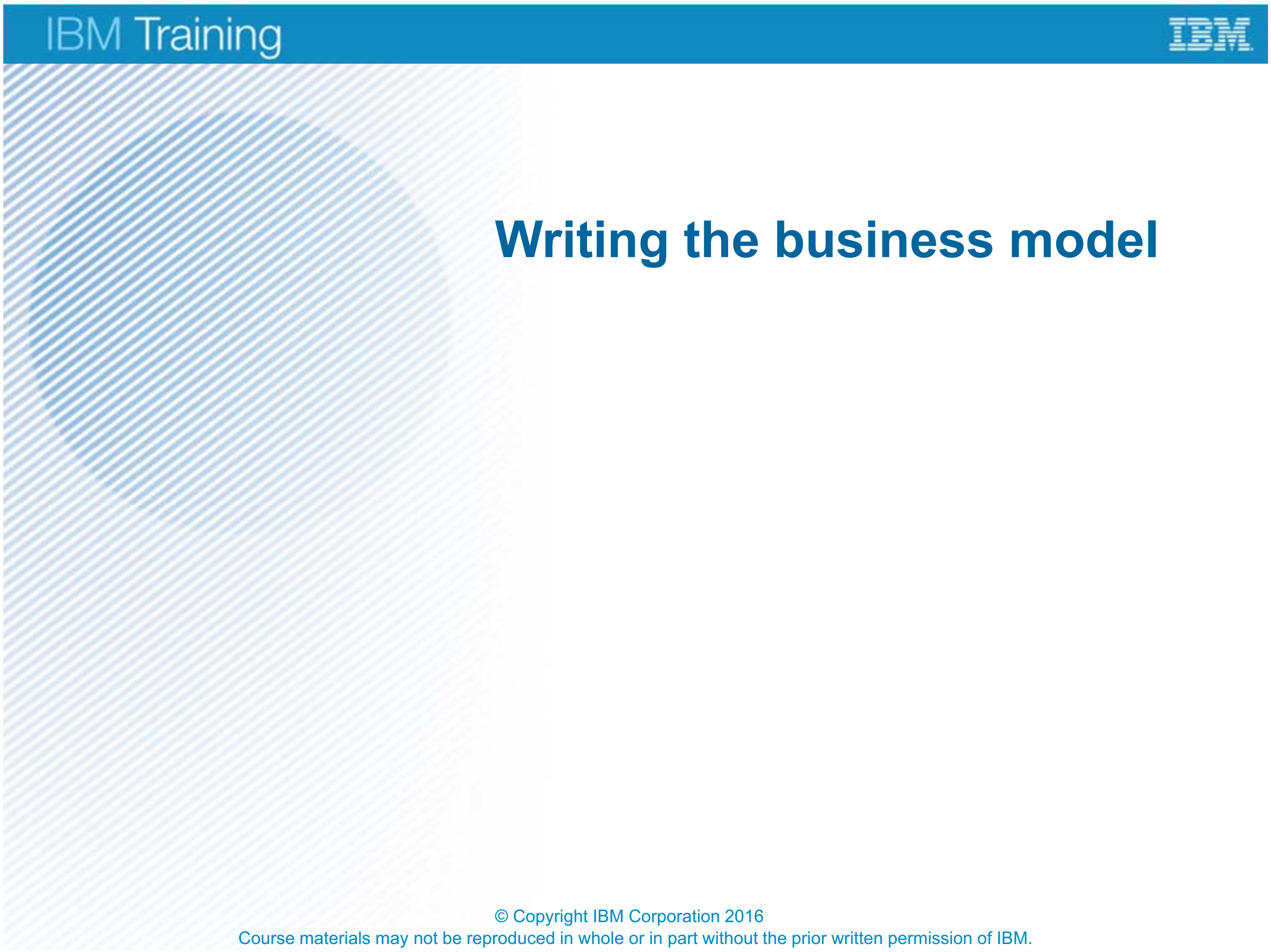


Figure 3-10. Writing the business model

IBM Training

Definition structure

- A definition contains one or more phrases
  - First phrase defines the nature of the concept as an entity, event, or simple concept
  - Phrases that follow specify the attributes and relationships to other concepts
  - Each phrase starts with the name of the concept and ends with a period (.)

- Example definition for a ticket entity
  - Identifier: ***ticket number***
  - Attribute: ***price***, which is a number
  - Relationships to: ***trip*** and ***customer*** entities, ***seating category***, and ***status*** enumerations

```
a ticket is a business entity
   identified by a ticket number.
a ticket has a price (numeric).
a ticket is related to a trip.
a ticket is related to a customer.
a ticket has a seating category.
a ticket has a status.
```

Creating the business model

Figure 3-11. Definition structure

So how do you start writing the definition? When you open the editor, you see an empty page. But as you start typing, the completion menu prompts you with a list of possible syntax options or constructs that are listed in a Content Assist box. You can either choose an option from the Content Assist box, or you can type freely, or even copy and paste from another file. As you add more entity and event definitions, these terms are added to the vocabulary, and also become available in the Content Assist box. So later definitions in that file allow you to choose from earlier definitions, which is useful when you want to define a relationship for example.

As you write your first definition, keep in mind the syntax. A definition contains one or more phrases to identify the type of concept, either an entity, an event, or just a simple concept. You also include phrases to specify attributes, and phrases to identify relationships to other concepts. Each phrase starts with the name of the concept (including the article, “a” or “an”) and ends with a period.

On this slide, you see the definition for a ticket entity. Each phrase starts with lowercase “a ticket” and ends with a period. The ticket definition also includes an identifier: ***ticket number***

So, the first line of this definition is:

`a ticket is a business entity identified by a ticket number`

The definition also includes attributes, which are identified by the ***has a*** construct. So a ticket ***has a*** price, and in brackets, price is of type **numeric**. Also, a ticket ***has a*** seating category and a ticket

has a status. Seating category and status might be defined as enumerations, which would limit the possible values for this attribute.

The ticket definition also includes phrases to identify the relationships. A ticket *is related* to a trip and a ticket *is related* to a customer.

So the **trip** entity type, the **customer** entity type, the **seating category**, and **status** enumerations all need to be defined in this file so that the ticket entity type definition has no errors.

IBM Training

Defining attributes

- Keywords to define attributes:
  - ***with***
  - ***has***
  - ***can be***

an address ***is a concept with*** a street, a town, a zip code ***and*** a country.

a customer ***is a business entity identified by*** an email ***with*** a first name, a last name, and an address.
a customer ***has*** a mobile number.
a customer ***has*** a gender.

- Attributes
  - Must have a type, and the instance of this type belongs to the instance of the object
  - Attributes must be separated by commas, except for the last attribute in the list, which is preceded by the ***and*** connector
  - Introduced by an indefinite article: **a**, **an**, **some**

Creating the business model

Figure 3-12. Defining attributes

Entities, concepts, and events can have attributes.

On this slide you see the keywords that are part of the predefined syntax to define attributes you use the words: ***with***, ***has***, or ***can be***.

For example, in this snippet from a business model definition, an **address** is defined with four attributes that addresses the concept. It is defined with the following attributes: ***street***, ***town***, ***zip code***, and ***country***. Notice that you do not need to have a separate phrase for every attribute; you can use a comma-separated list.

The next example is the **customer**, which is a business entity that is identified by an email with a first name, a last name, and an address. The additional attributes **mobile number** and **gender** are also included for the customer by using the ***has*** keyword.

When you define an attribute, it must have a type. For example, a first name and a last name would be of type **string** or gender would be an enumerated list. Also, notice that the attributes were introduced by an indefinite article you can use: ***a***, ***an***, or ***some***.

IBM Training

Entity definition

- Entities have an identifier and their own lifecycle

a car ***is a business entity identified by*** a vin ***with***
a make,
a model,
a year ***(integer)***.

a car ***is related to*** a policy.

a policy ***is a business entity identified by*** an id.
a policy ***has*** a start ***(date & time)***.
a policy ***has*** an end ***(date & time)***.
a policy ***is related to*** a car.
a policy ***is related to*** a customer.
a policy ***has*** a fraud status.

Creating the business model

Figure 3-13. Entity definition

This slide shows two more examples of entity type definitions.

All the entity types that you defined for your domain have a lifecycle, which means that when you have an entity of that type it is created with a certain set of initial values. As events arrive in the Insights runtime, those events have some impact on the lifecycle of the entities.

In general, you define entities that have a long lifespan within your domain. On this slide, if your domain is car insurance and your car entity has a lifecycle that begins when the car is first insured. That lifecycle lasts for the duration of the policy. The policy also is an entity with a lifecycle. In the business model definition snippet here, the lifecycle of the policy entity is defined by the start and end date.

IBM Training

Event definition

- An event is a concept with a time stamp
  - Define events to represent the things that happen in your business
- Each event has a time of occurrence

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

a client related event ***is related to*** a client.

Creating the business model

Figure 3-14. Event definition

Here, you see an example of an event type definition.

The definition uses the predefined syntax term: ***a business event time-stamped by a timestamp***

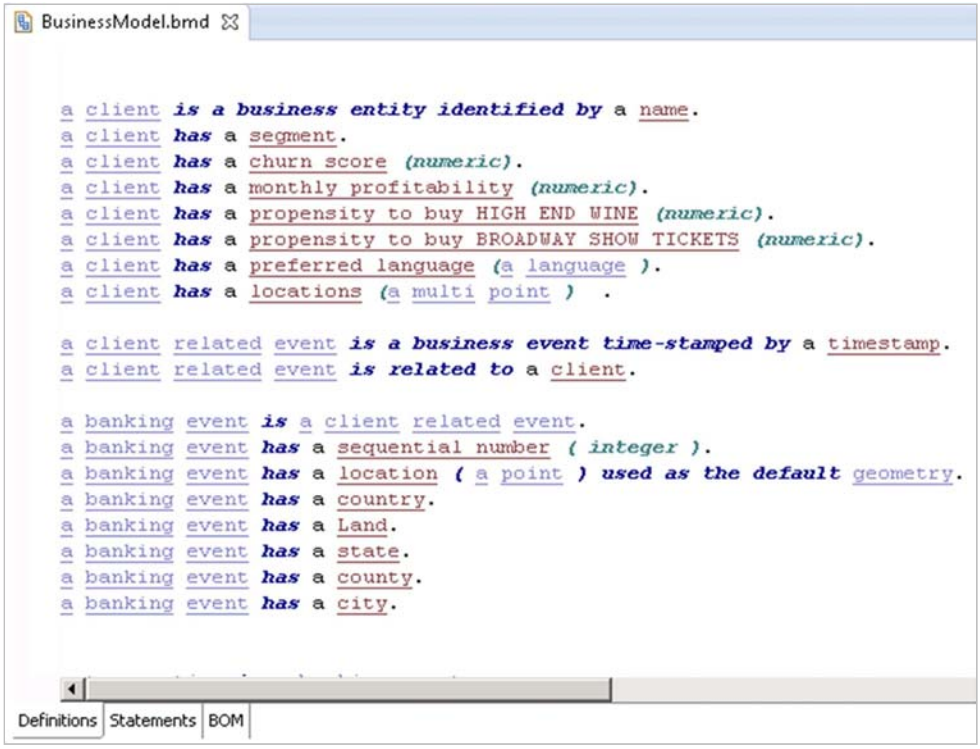
Every event that arrives in the Decision Server Insights runtime receives a time stamp to determine how it is processed.

As you can see from the definition, this event type is related to a client entity type so that relationship is defined by the ***is related to*** construct.

IBM Training

Business model editor (1 of 3)

- **Definitions** tab
- Model is written with natural language in the BMD



Creating the business model

Figure 3-15. Business model editor (1 of 3)

Here, you see a screen capture of the business model definition in the business model editor.

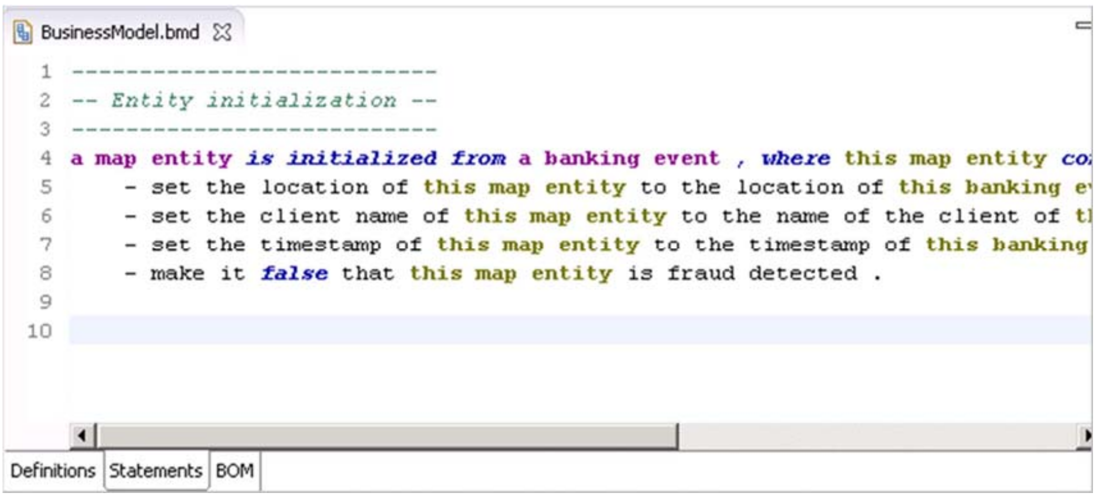
The predefined constructs that are provided as part of the model definition language are italicized and in bold. Notice that the end of every phrase has a period.

On the **Definitions** tab of the editor, you can type freestyle. You can also press Ctrl+Space to use the editor and see the list of syntax options that are available in the Content Assist box. You can even copy and paste definitions from another file.

IBM Training

Business model editor (2 of 3)

- Use **Statements** tab: In the Statements editor, you write behaviors for your business model definitions
  - Initialize entities from an event
  - Enrich attributes
  - Derive attributes
- Example: Definition of initializing a “map entity”



Creating the business model

Figure 3-16. Business model editor (2 of 3)

A data provider is a piece of Java code that calculates the value of an entity's attribute at runtime. A data provider typically fetches data in a database or calls a web service.

The screen capture on the slide shows the **Statements** editor. On the **Definitions** tab, if you define a type of entity that requires methods, you use the **Statements** tab to complete those methods.

For example, on the **Definitions** tab, if you define a language provider, then, on the **Statements** tab you define how that language provider should behave.

IBM Training

Business model editor (3 of 3)

- Use **BOM** tab to view and verify code implementation

Creating the business model

Figure 3-17. Business model editor (3 of 3)

As a developer, you might find the format of your model definitions easier to read and review in the BOM view of the editor. By clicking the **BOM** tab, you can see the implementation of your model.

IBM Training

Unit summary

- Describe the elements of a business model
- Translate a UML diagram into a business model definition

Creating the business model

Figure 3-18. Unit summary

IBM Training

Review questions

1. True or False: Entity and event types are defined in UML.
2. True or False: BMD supports inheritance between entity types and between event types.
3. True or False: Every entity has a unique identifier.
4. True or False: Every event has a timestamp.

Figure 3-19. Review questions

Write your answers here:

1.

2.

3.

4.

IBM Training

Review answers

1. True or False: Entity and event types are defined in UML.
Answer: False: Entity and event types are defined in the BMD.

2. True or False: BMD supports inheritance between entity types and between event types.
Answer: True. The BMD supports inheritance between entity types and between event types through the "is a" construct.

3. True or False: Every entity has a unique identifier.
Answer: True.

4. True or False: Every event has a timestamp.
Answer: True.

Creating the business model

Figure 3-20. Review answers

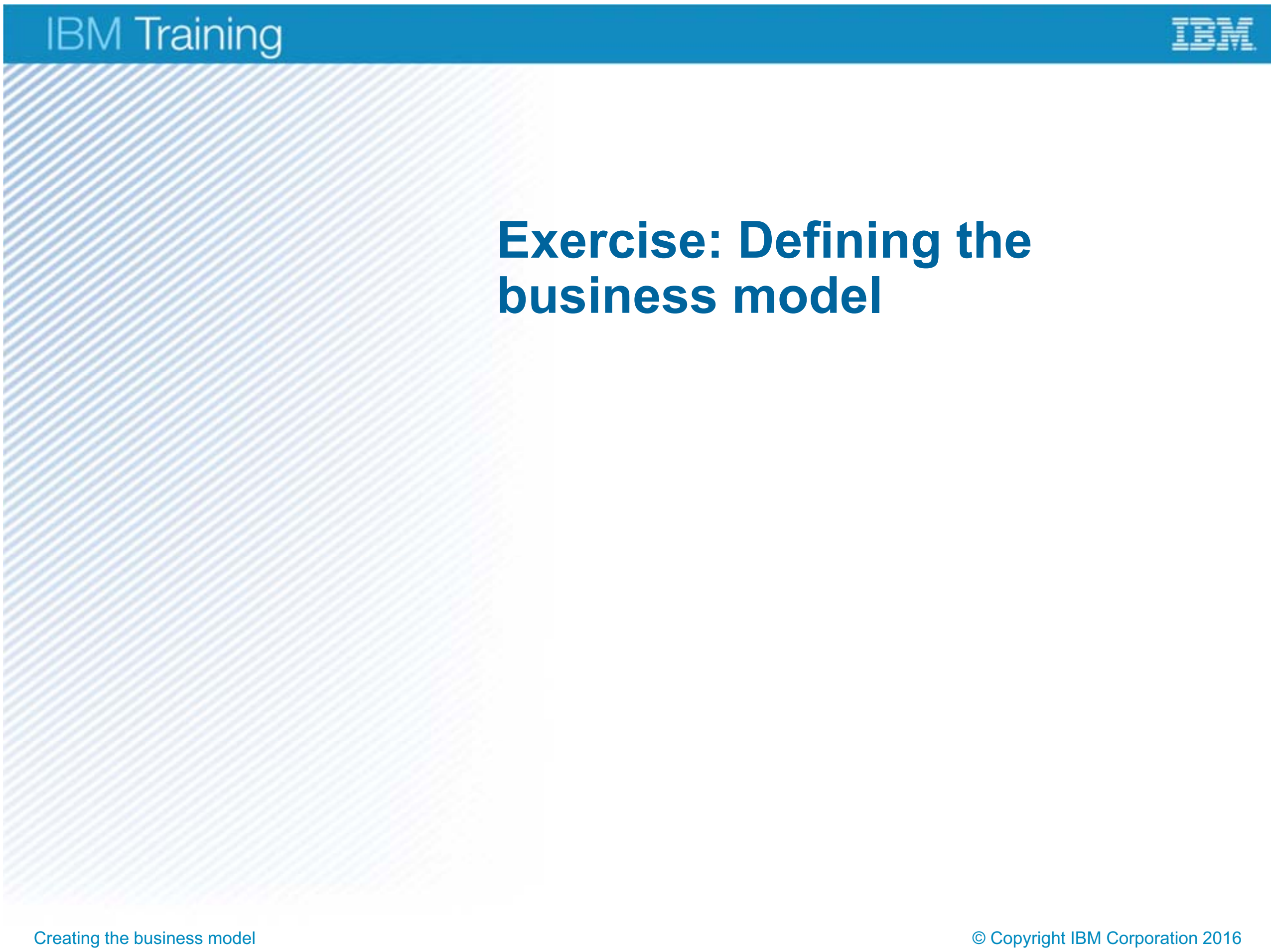


Figure 3-21. Exercise: Defining the business model

IBM Training

Exercise objectives

- Create a business model definition file

Creating the business model

Figure 3-22. Exercise introduction

Unit 4. Authoring the business logic

Estimated time

01:30

Overview

This unit teaches you how to implement the business logic with rule agents.

How you will check your progress

- Review
- Exercises

IBM Training

Unit objectives

- Describe the structure of rule agents, Java agents, and predictive scoring agents
- Implement business logic with rules
- Explain how to implement time-based reasoning
- Describe location-based tests

Authoring the business logic

Figure 4-1. Unit objectives

IBM Training

Topics

- Implementing business logic with agents
- Rule agents
- Java agents and predictive scoring agents

Authoring the business logic

Figure 4-2. Topics

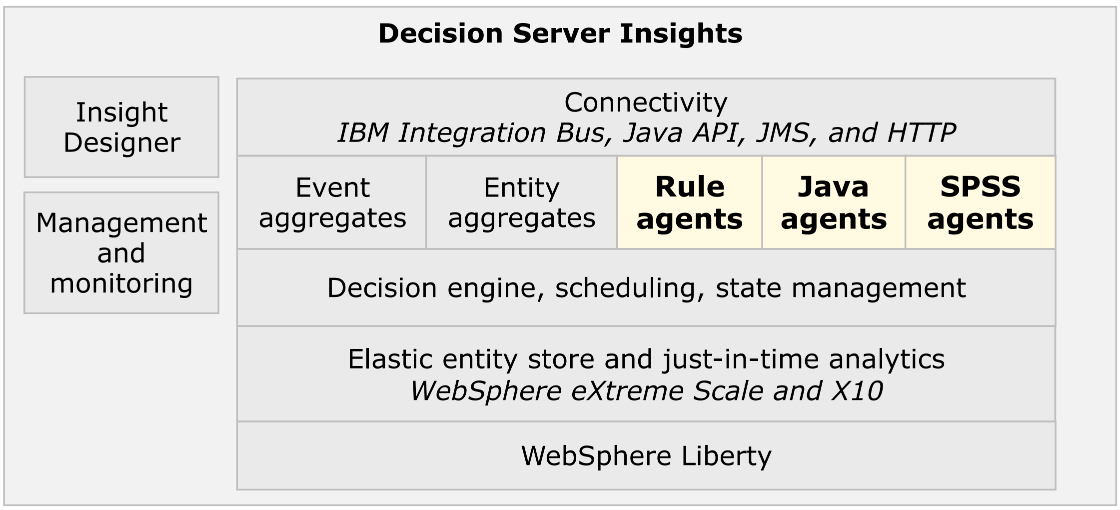
4.1. Implementing business logic with agents

Figure 4-3. Implementing business logic with agents

IBM Training

Runtime architecture: Agents

- Three types:
  - Rule agents
  - Java agents
  - Predictive scoring agents



Authoring the business logic

Figure 4-4. Runtime architecture: Agents

Decision Server Insights uses three types of agents: rule agents, Java agents, and SPSS or predictive scoring agents.

The agents are managed in the runtime tier, where they have direct access to the entities that they monitor.

As events arrive through the connectivity tier, they are routed to the agents that are listening for or subscribed to those events. The agents are bound to an entity, and when the event arrives, the agent processes the event for that entity. So if an event affects that entity, the agent handles it.

The Decision Server Insights runtime automatically and transparently maintains the state that is required for temporal and stateful computations by the agent.

IBM Training

Agents

- Use agents to define the business logic that binds incoming events to entities
  - Define routing logic between events and entities
  - Define binding logic that determines which entity to target
  - Define function logic that defines how to process the event
- An agent detects incoming events, and can also generate new events
- Agents can either be triggered by the arrival of events or be scheduled for execution

Authoring the business logic

Figure 4-5. Agents

Agents define the routing logic between events and entities. The agent also defines the binding logic that determines which entity to monitor and they define the function logic that determines how to process the event.

You use your agents to detect incoming events or event patterns and also to emit new events. Agents can be triggered by the arrival of events or you can schedule them for execution.

IBM Training

Agents and entities

- An agent is bound to only one entity but can listen to several events
- Several agents can be bound to the same entity
- The binding to the entity is defined at the agent level, in an agent descriptor
- If an event is dispatched to several agents that target the same entity, you can define priority for the agent to determine the runtime invocation order

Authoring the business logic

Figure 4-6. Agents and entities

An agent can be bound to only one entity but it can listen to several events and it can emit new events. An agent can create, update, or remove its bound entity. Agents also have read-only access to other non-bound entities.

You can have multiple agents that are bound to the same entity. The binding is defined at the agent level in agent descriptor. In the descriptor, you can also define a priority for the agent to determine the runtime invocation order. So if multiple agents subscribe to the same event and are bound to the same entity the priority determines which agent processes the event first.

IBM Training

Agent types (1 of 2)

Rule agent:

- Rule agents act on entities through rules
- Rules can generate events and emit them into and out of the system
- The vocabulary within the rule agent has the following access rights:
  - Read and write on all attributes of the bound entity
  - Read-only on attributes of entities that are referenced through relationships (remote entities)
  - Read-only on event attributes

Authoring the business logic

Figure 4-7. Agent types (1 of 2)

As mentioned, Decision Server Insights uses three types of agents: the rule agent, the Java agent, and predictive scoring agent or SPSS agent.

The predictive scoring agent is a Java agent that uses SPSS scoring configurations to access external scoring servers. You learn more about SPSS features later in this course, during the unit on integration.

The rule agent is the agent that makes the decisions, so this agent is where you implement your business logic.

Agents access their bound entity in various ways. The rule agents act on entities through rules. The rules are written in natural language by your business experts, who can read, review, and even update themselves.

IBM Training

Agent types (2 of 2)

Java agent:

- A Java agent processes events by using Java code
- The agent has access to the entity model and can establish a Java coded logic
- A Java agent does not have to be bound to an entity
- Predictive scoring agent:
- A predictive scoring agent is a Java agent that can use an SPSS scoring configuration that is hosted externally from Decision Server Insights

Authoring the business logic

Figure 4-8. Agent types (2 of 2)

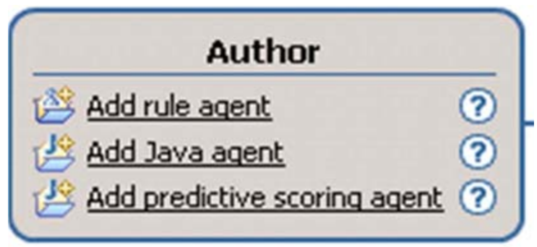
Java agents process events by using Java code. Because Java agents can be developed to do many things that do not require access to a particular entity, you can create a Java agent with no bound entity.

In the descriptor, you omit the clauses that bind the agent to an entity.

IBM Training

Creating rule agents

- After you create a solution project and business model, you create agents
- Agents describe the bound entity and subscribe to events of interest
- To develop an agent:
  1. Create the agent project
  2. Write the descriptor
  3. Write the rules
- You can use the links in the **Author** goal of the **Solution Map** to create agents



Authoring the business logic

Figure 4-9. Creating rule agents

Before you can create an agent that is bound to an entity or that subscribes to an event, those entities and events need to be defined in your business model.

To create an agent, you can use the links in the Solution Map of the Decision Insight perspective. For example, to create a rule agent, click the **Add rule agent** link to open the wizard, which guides you through the steps of creating the agent.

IBM Training

Agent projects

- Agents are represented as:
  - Rule agent projects
  - Java agent projects
  - Predictive scoring agent projects
- An agent project contains an agent descriptor and the business logic

Authoring the business logic

Figure 4-10. Agent projects

All the agents are represented as agent projects so each rule agent is stored in a separate project and the project contains the agent descriptor and the business logic or code.

IBM Training

Agent descriptors

- Descriptors define the agent signature, which includes:
  - The events that the agent wants to process
  - The event field that is the key for the entity
  - The routing logic
- Agent signature
  - Determines the authoring context to help rule authors write the rules
  - Defines the applicability condition

Authoring the business logic

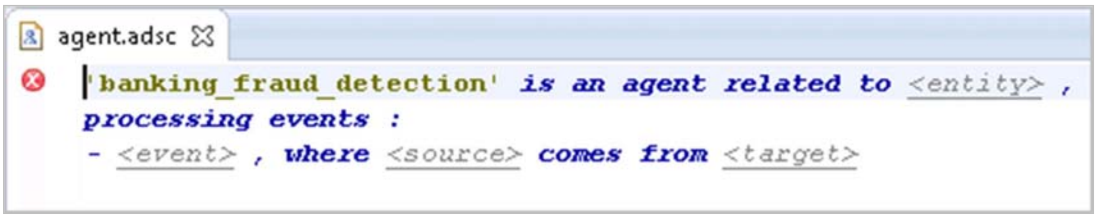
Figure 4-11. Agent descriptors

Agent descriptors were mentioned earlier. The agent descriptor is what defines the agent signature, including which events the agent wants to listen to or subscribe to, which event field is used as the key for the entity, and the routing logic. This agent signature determines the authoring context to help rule authors write the rules and it defines the applicability condition.

IBM Training

Writing an agent descriptor

- Descriptor is stored in the `.adsc` file
- When you create the agent project, the `.adsc` file automatically opens in the editor



- All agents are written with business vocabulary
  - Before you can define an agent, you must define the entities and events in the business model (`.bmd` file)
  - Vocabulary to build the agent comes from the `.bmd` file

Authoring the business logic

Figure 4-12. Writing an agent descriptor

A descriptor is written and stored in a **`.adsc`** file. When you create an agent project, a descriptor file is automatically generated and opens in the editor. Some initial text is already in place, including the name of your agent. You need to choose which entity to bind the agent to and which events to subscribe to.

IBM Training

Example descriptor

- In this example, the agent is bound to “a client” entity and is subscribed to banking events

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

Authoring the business logic

Figure 4-13. Example descriptor

In this example, you see a rule agent that is named ***banking fraud detection***, which is related to a client entity type. This agent processes all banking events that affect its client entity to detect fraud patterns. The rules that are attached to the agent implement the business logic for fraud detection.

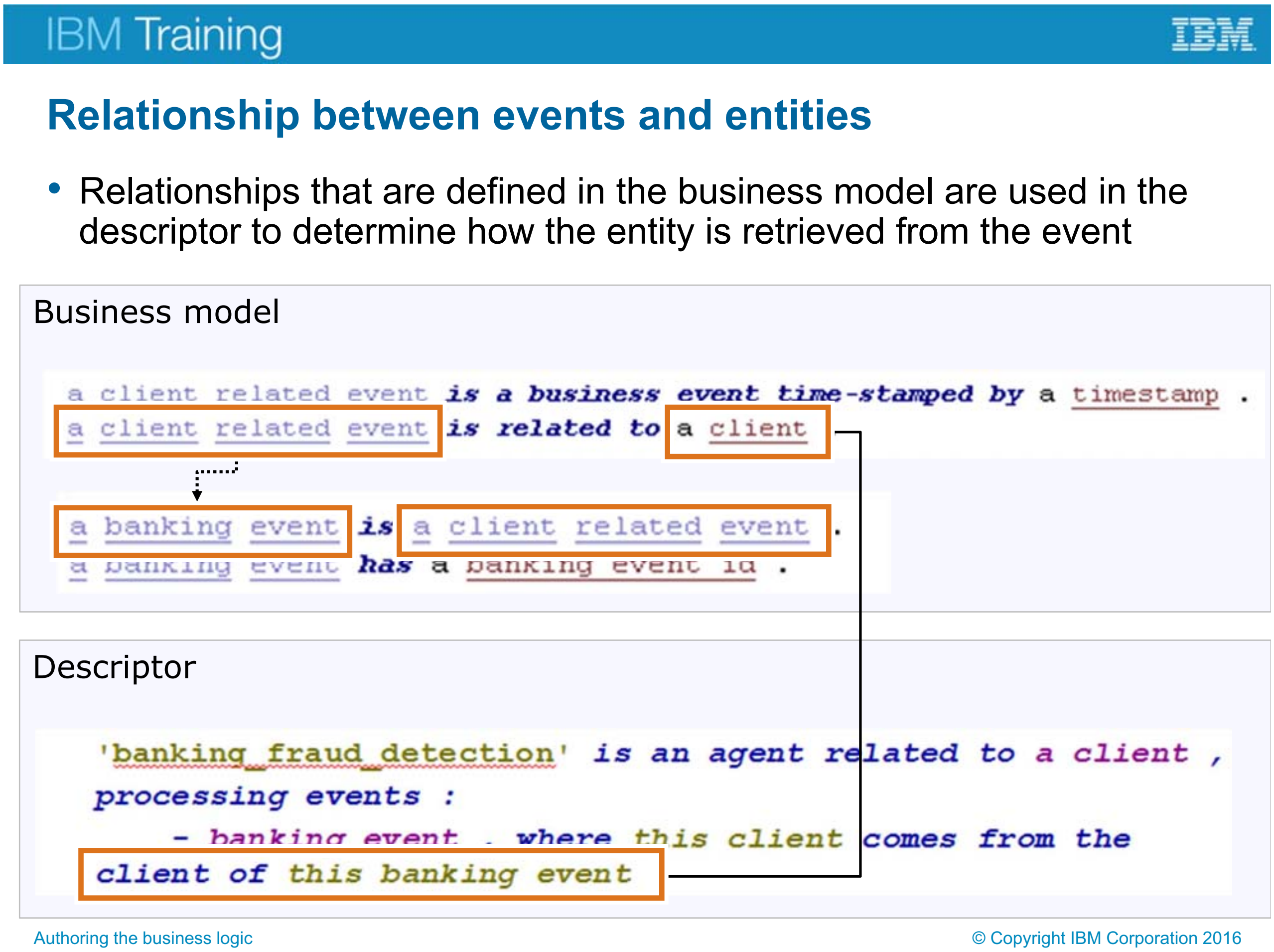


Figure 4-14. Relationship between events and entities

When you complete your descriptor file, the completion editor prompts you with a list of possible options. But where do those options come from? They come from your business model.

You cannot define the banking fraud detection agent that is being bound to a client entity unless you have a client entity that is already defined in the model. Nor can you listen to a banking event unless the banking event is also defined in your model.

As you see on this slide, the vocabulary that was used to build the descriptor comes directly from your business model definition.

You also see the significance of inheritance relationships. According to the business model definition, a banking event ***is*** a client-related event, and a client-related event ***is related to*** a client. Therefore, a banking event is automatically related through inheritance to a client.

When your agent subscribes to a banking event, that banking event is automatically routed to the correct agent, the one monitoring the client entity that is related to this banking event.

4.2. Rule agents

IBM Training

Rule agents

Figure 4-15. Rule agents

IBM Training

Creating rule agents

- A rule agent encapsulates the business logic that updates the state of its bound entity
- You write rules to define the business logic of the agent
- The rules define the action to be taken when an agent receives an event
- An agent can listen to several events, but is always associated with a bound entity
  - The binding to the entity is defined at the agent level
  - The rule references the entity through the authoring context that is defined by the agent descriptor
- By using the rules, rule agents can emit new events

Authoring the business logic

Figure 4-16. Creating rule agents

You create a rule agent to make decisions within the insights solution.

The rule agent acts as a container for the rules that implement the business logic that is required by that agent.

IBM Training

Structure of an action rule

- A rule can comprise some or all of the following parts

when	The *when* part defines the processing of an event
definitions	The *definitions* part defines variables that can be used within the rule
if	The *if* part defines the condition upon which an action is executed
then	The *then* part defines the action to be taken if the condition is true
else	The *else* part defines the action to be taken if the condition is false

Authoring the business logic

Figure 4-17. Structure of an action rule

A rule defines the specific actions to take when certain conditions are met. A rule is composed of several parts, as you see here, that define the conditions and actions.

A basic rule associates a condition with an action. If the condition tests true, take a particular action. The vocabulary available in the rules comes from the elements that are defined in the business model and the agent descriptor. The agent descriptor provides the context for your rule authoring because it limits which vocabulary is available to you as you write the rule.

As outlined in this table, a rule can comprise some or all of the following parts. The parts that you include depend on the type of rule that you want to write.

The ***when*** part defines the processing of an event. For rules that are used to apply logic to event processing, you always include the ***when*** part.

The ***definitions*** part is where you can define variables that are used in the rule. If you define a variable in the ***definitions*** part, the scope of that variable is the rule itself. That variable is not visible or accessible outside the rule.

In the ***if*** part, the ***if*** part of the rule defines the condition tests. When you define a condition, the action is only executed when the condition tests true.

The ***then*** part of the rule defines the action to take when the condition tests true. The ***else*** part can also be used to define what action to take when the condition tests false.

All these parts of the rule are optional except the ***then*** part. You can have a rule that includes only an action, in which case, that rule would always execute every time that it is invoked because it is not dependent on a condition.

IBM Training

Defining the “when” part of a rule

- To process an event immediately when it arrives in the agent, use:
  `when <event> occurs`

- When an event is received, the rule is instantiated for this event
  - The rule instance is applicable only when this event is processed and is no longer applicable when later events arrive

- You can add filters to the ***when*** part by adding a ***where*** clause

```
when a banking event occurs
  where the state of this banking event is CA
```

Authoring the business logic

Figure 4-18. Defining the “when” part of a rule

Rules that are used to process events start with the ***when*** part.

The ***when*** part and the ***then*** part are mandatory.

The ***when*** part can be written to execute a rule as soon as the event arrives or it can be written to wait or postpone the event processing. You learn more about postponing event processing in the next slide.

To process an event as soon as the agent receives it, use the clause: `when event occurs`

When the event that is specified in this clause is received, a rule is instantiated for this event. How does that work?

If you are familiar with Decision Server Rules, you know how the rule engine and working memory are used to manage from instances. For Decision Server Insights, the rule agent has a working memory. When a rule agent receives a new event, the agent tests the event against all its rules and sees whether the event matches the when clause for any of the rules. If it makes a match between an event and a rule, the agent creates a rule instance for every match.

The next step is for the agent to select a rule instance from the list, based on some specific criteria such as priority, and then execute that rule.

To execute a rule instance, means to apply the action statement. The execution can modify the state of the entity that the agent is monitoring and might affect the results the next time that rule agent tries to match events to rules. After a rule instance is executed, the rule instance is deleted so that it does not execute any more. So a rule instance applies only while it's being evaluated to events that it matched.

You can write rules that do not have a ***when*** part. Those rules would be evaluated either every time a new event arrives or by an internal scheduling mechanism. An instance of rule without a when part might be applied several times.

IBM Training

Using “when” to postpone event processing

- To postpone processing an event, use:
  `when <event> has occurred`
- To indicate the delay period, you use a time expression with an implicit reference to `now`
  - 1 day ago
  - 2 hours ago
- Use postponed processing if:
  - An event depends on other events
  - The order in which the events are processed is important
  - An action must happen after a certain duration
  - You want to test the presence or absence of events

Authoring the business logic

Figure 4-19. Using “when” to postpone event processing

To postpone the processing of an event, you use the phrase: **`when event has occurred`**

To indicate how long the delay period should be, you use a time expression such as ***1 day ago***, or ***2 hours ago***. This expression implicitly compares the delay to “now” which represents the time stamp of the event.

This type of construct to postpone or schedule an event is useful when you have an event that depends on other events. It is also useful in the following situations:

- When the order in which events are processed is important
- When an action must happen after a certain duration
- When you want to test for the presence or absence of events

IBM Training

Example: Testing for the absence of an event

- This rule postpones processing by 30 minutes:

```
when a fraud alert has occurred 30 minutes 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 ;
```

Authoring the business logic

Figure 4-20. Example: Testing for the absence of an event

This example shows how to postpone event processing, and how to test for the absence of an event.

The ***when*** clause says, “when a fraud alert has occurred 30 minutes ago” (meaning 30 minutes before now, which represents the time stamp of the event).

To test for the absence of an event, the ***if*** statement says “if there is no confirmation from the client.” This statement means that if no notification was received within that 30-minute delay period, then the action emits a new notification event.

IBM Training

Defining rule variables

- Define the variable in the ***definitions*** part of a rule
- Variables can be set to:
  - An expression
  - An event type
  - A constant
  - A collection of values
- Example: `'RECENT TRANSACTIONS'` is set to a collection of withdrawals

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

Authoring the business logic

Figure 4-21. Defining rule variables

To implement the business logic, you might need to define variables. The variables that you define in the ***definitions*** part of the rule are only accessible or visible within the scope of the rule.

A variable can be set to an expression, an event type, a constant, or a collection.

For example, you might define a variable called ***recent transactions***, which is sent to a collection of withdrawal transaction events.

As you see in this example, the syntax to define the variable is:

```
set recent transactions to all withdrawals during the last period of 50 days
```

IBM Training

Calling global aggregate variables in rules

- Use global aggregates in your rules in the same way that you use other rule variables
- Example: Set a variable to the value of a global entity aggregate

```
definitions
 set AVG_CHURN to 'Average churn for GOLD' ;
if
 the churn score of 'the client' is at least 1.5 *
AVG_CHURN
then
 emit a new gift where
  the code is COUPON 100 USD ;
```

Authoring the business logic

Figure 4-22. Calling global aggregate variables in rules

Within your rules, you can also refer to global aggregates in the same way that you would use other variables.

The scope of global aggregates makes them accessible to all agents within your solution.

So in the example that is shown here, a local rule variable is set to the value of a global entity aggregate. The definition says:

`set AVG_CHURN to Average churn for GOLD`

Where `Average churn for GOLD` is the name of your global entity aggregate.

You learn more about global aggregates later in this unit but for now it's important to know that you can use the global aggregates throughout your rule.

However, because global aggregates are defined and calculated by different process than rule variables, you cannot create a global aggregate within a rule. The scope of the global aggregate is the entire solution, whereas the scope of rule variable is just the rule.

IBM Training

Defining conditions

- Use the ***if*** part to state under which conditions to carry out the rule actions that are defined in the ***then*** and ***else*** parts

- Use condition tests to:
  - Compare statements
  - Test for a number
  - Test if an object belongs to a set

- Negate conditions
  - `it is not true that`
  - `none of the following conditions are true:`

Authoring the business logic

Figure 4-23. Defining conditions

The ***if*** part of the rule is used to state the conditions under which to complete the rule actions that were defined in the ***then*** or in the ***else*** parts of the rule.

Earlier, you saw that the rule agent performs a pattern matching, where it tests the events against all the rules to see whether the event matches any of the ***when*** clauses in any of the rules. The agent also tests the event against the ***definitions*** part and the ***if*** parts of the rule, looking for a match.

Even if the rule does not have a ***when*** part, the rule agent still creates a rule instance when an event matches the ***definitions*** or the ***if*** parts of the rule.

In the ***if*** part, you create your condition tests to compare statements, to test for a number, or test whether an object belongs to a set.

You can also use negative statements to negate conditions, such as: “if it is not true that” or “none of the following conditions are true.”

Remember that conditions are not mandatory in a rule. If you have a rule with no conditions, that rule always executes, which can be useful when you need a rule to always be evaluated.

IBM Training

Defining actions

- Rule actions have at least one action phrase in the ***then*** part of the rule that executes when the ***if*** part of the rule test true
  - Optionally, you can include ***else*** statements to execute when the ***if*** part of the rule is false
- You can use action phrases to create bound entities
  - Set the rule to the highest priority and test for null entities and create them before other rules in the agent execute
  - Example: Create an account from event data

```
 when a new account event occurs, called 'the event'
if
    'the account' is null
then
    set 'the account' to a new account where
        the customer is the customer of 'the event',
        the opening date is 'the event',
        the status is NEW;
```

Authoring the business logic

Figure 4-24. Defining actions

The final part of the rule structure to discuss is the actions part of the rule.

Actions can be defined in the ***then*** part of the rule, and optionally in the ***else*** part.

The ***then*** part is mandatory for all your rules, and they must have at least one action phrase.

If your rule has condition statements, and the condition tests true, the action phrases in the ***then*** part are executed. If you have an ***else*** part in your rule and the conditions test false, then the action statements in the ***else*** part are executed.

Generally, it is not as easy to maintain rules that have an ***else*** statement. For more guidelines on when to use an ***else*** statement, see the product documentation.

The action statements that you define in the ***then*** part of the rule can be used to modify entity data or to even create an entity.

For example, if you need to initialize entities with values, you can set a rule with the highest priority to test for all null entities and create them before other rules in the agent execute. By initializing entities, you ensure that you do not cause null pointer errors.

In the example that is shown here, you see that the action statement creates an account with initial values. After testing whether the account is null, the then statement says:

```
set the account to a new account where …
```

Then, it provides some initial values for the account entity.

IBM Training

Time-related tests (1 of 2)

- Time operators can be used to calculate useful values, such as:
  - An elapsed time
  - A time window
  - A window of opportunity
- To use the time of the current event or identify when events arrive out of sequence, you can use the time reference: `now`
  - The `now` value is set from the time stamp of the event
  - Example: Compare the calendar date to `now` as set by the banking event time stamp

```
when a banking event occurs
if
  now is after 1/1/2016
then
  print "A banking event occurred after 1/1/2016"
```

Authoring the business logic

Figure 4-25. Time-related tests (1 of 2)

For event processing logic, time-related tests are essential.

Time operators can be used to calculate values, such as an elapsed time, a time window, or a window of opportunity.

To use the time of a current event, or identify when events arrive even when they are out of sequence, you can use the time reference: **now**

Events have a time stamp that is defined by the event source. In most cases, you see a delay between the time stamp of the event and the moment when the event arrives into the system to be processed by the agent.

An event that happened first might be processed *after* another event that happened later. When an event is processed out-of-order, its time stamp is considered before **now**.

In this example, you can compare the calendar date to **now** as set by the banking event time stamp. So in the condition test, “if **now** is after January 1, 2016” means if the time stamp of the banking event is after January 1, 2016, then apply the action.

IBM Training

Time-related tests (2 of 2)

- Time points
  - An instant of time that is measured in a time scale of a specific time zone
  - Can be specified by a date, an event, or an attribute of type date or time

- Operators to compare time points:
  - `is after` and `is after or the same as`
  - `is before` and `is before or the same as`
  - `is at the same time as`

- Operators to compare a time point and a duration:
  - `<duration> before <date>` | `<duration> after <date>`

- Operators to compare a time point and a time period:
  - `before` | `after`
  - `during`
  - `includes`
  - `starts at` | `ends at`

Authoring the business logic

Figure 4-26. Time-related tests (2 of 2)

IBM Training

Location-related tests

- You can write rules to reason over location-aware entities or events and perform the following operations:
  - Detect whether some locations contain or are contained by other locations
  - Calculate the distance between two locations
  - Define a perimeter in which locations are contained
  - Check whether a location intersects a path or an area
  - Find the nearest location to another location
  - Add a location to a path or route
- Use <*a geometry*> to reference any type of geometry that is defined in the business model, such as a point

Authoring the business logic

Figure 4-27. Location-related tests

Objects that have a geographic location must define a spatial geometry. A spatial geometry can be static or moving.

IBM Training

Static and moving geometry types

- Static geometries
  - Represent the location of an entity or an event that does not move over time
- Moving geometries
  - Represent the location of an entity that moves over time
  - A moving geometry has a location at a specific time
- Use a static geometry to model location-related objects, such as a building, a field, a county, or a state
  - Example: Business office location (static)

```
a business has a location (a geometry) used as the
default geometry.
```

- Use a moving geometry to model an entity whose location that changes over time
  - Example: Car location (moving)

```
a car has a location ( a moving geometry ) used as the
default moving geometry .
```

Authoring the business logic

Figure 4-28. Static and moving geometry types

Objects that have a geographic location must define a spatial geometry. A spatial geometry can be static or moving.

You use a **static** geometry to model location-related objects, such as a building. You use a **moving** geometry to model an entity whose location changes over time. The time that is needed to access the value of a moving geometry and the memory that is required depend on the size of the history limit that you define.

IBM Training

Geospatial attributes and operators

- You can define how long to track movement when you initialize the location of a moving entity
  - `set <attribute> of [attribute of]* <entity> to a new moving geometry [`**`tracked for <calendar period>`**`]`
  - Default history limit is 24 hours

- Geospatial operators for moving geometries include:
  - `the maximal | minimal | average speed of <moving geometry> in <speed unit> over <time period>`
  - `<moving geometry> is leaving <geometry>observed location at timestamp`
  - `<moving geometry> is approaching <geometry>`
  - `<moving geometry> is accelerating`
  - `<moving geometry> has been located in <a polygon> during <time period>`

Authoring the business logic

Figure 4-29. Geospatial attributes and operators

You define geometries and moving geometries as attribute types in your model. Static geometries are defined as “a geometry” and moving geometries are defined as “a moving geometry.”

Geospatial operators for moving geometries include calculations of the maximal, minimal, or average speed of a moving geometry over a time period. Or you can check a moving geometry’s location at a specified time point. You can also check whether the distance between entities is decreasing or increasing, such as when two objects are approaching or leaving.

4.3. Java agents and predictive scoring agents

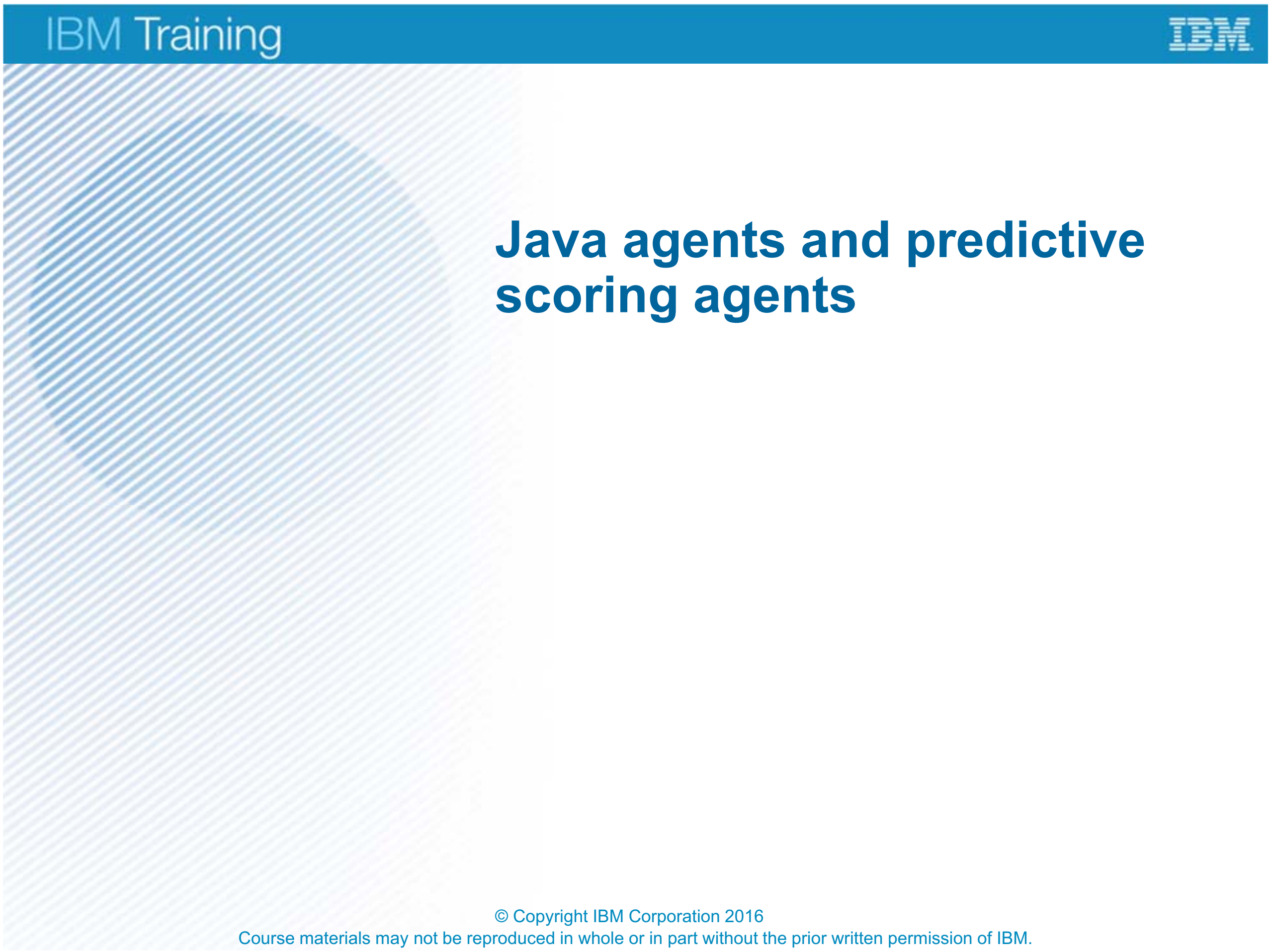


Figure 4-30. Java agents and predictive scoring agents

IBM Training

Setting the target platform in Insight Designer

- In the Decision Insight perspective, set the target platform to Insight Server before creating Java agents or predictive scoring agents

1. In Insight Designer, use the **Window > Preferences** menu
2. Select **Plug-in Development > Target Platform**
3. Add the Insight Server template
4. Select the Insight Server template as the target

Authoring the business logic

Figure 4-31. Setting the target platform in Insight Designer

When you develop Java agents and predictive scoring agents in Insight Designer, you must first set the target platform to build the Java code.

If the target platform is not set, you see errors on the agent projects.

During the lab, you see how to do this task.

IBM Training

Java agents

- Use Java agents for complex computations or to call a web service
- The Java agent project contains the descriptor and a `.java` class file
- You can schedule Java agents to run at a specific time by using the com.ibm.ia.agent.Agent API

```
public String schedule( int delay, TimeUnit unit, String cookie
)
```


Figure 4-32. Java agents

You can use Java agents any time you need to write code, for example, for complex computations, or to call a web service, or to create some type of data provider. The reasons for using Java code are innumerable.

One thing to keep in mind is that the Java agent is not designed to make the decision. All business logic should be handled by the rule agent.

Like a rule agent, a Java agent contains a descriptor that binds it to an entity. However, you can also create Java agents that are not bound to an entity.

When you create a Java agent, a Java class file is automatically generated that you need to complete.

Java agents can also be scheduled to run at a specific time by using the agent API that is provided with Decision Server Insights.

IBM Training

Predictive agents

- To use SPSS predictive models in decisions, you create a predictive scoring agent
- Agent invokes the scoring server and updates entity state with score value
- Steps for creating a predictive scoring agent
  1. Define the agent descriptor
  2. Complete the agent `.java` class
  3. Create the `solution_properties.xml` file
  4. Add solution properties for scoring service endpoints

Authoring the business logic

Figure 4-33. Predictive agents

To work with SPSS, you use predictive scoring agents in your solution.

The predictive scoring agent contains connection details so that it can call a running SPSS server and have access to predictive scoring models.

Predictive scoring agents are like Java agents. They also contain a descriptor and have a Java class that you need to complete. When you create a predictive scoring agent, you also add solution properties for the scoring service endpoints.

IBM Training

Unit summary

- Describe the structure of rule agents, Java agents, and predictive scoring agents
- Implement business logic with rules
- Explain how to implement time-based reasoning
- Describe location-based tests

Authoring the business logic

Figure 4-34. Unit summary

IBM Training

Review questions

1. True or false: An agent is bound to only one entity, and several agents can be bound to the same entity.

2. The rule agent can postpone processing an event by using which construct?
   a. Postpone <event> for <time duration>
   b. When <event> has occurred
   c. When <event> occurs

Authoring the business logic

Figure 4-35. Review questions

Write your answers here:

1.

2.

IBM Training

Review answers

1. True or false: An agent is bound to only one entity and several agents can be bound to the same entity
   Answer: True

2. The rule agent can postpone processing an event by using which construct?
   a. Postpone <event> for <time duration>
   b. When <event> has occurred
   c. When <event> occurs

   Answer: b. When <event> has occurred

Authoring the business logic

Figure 4-36. Review answers

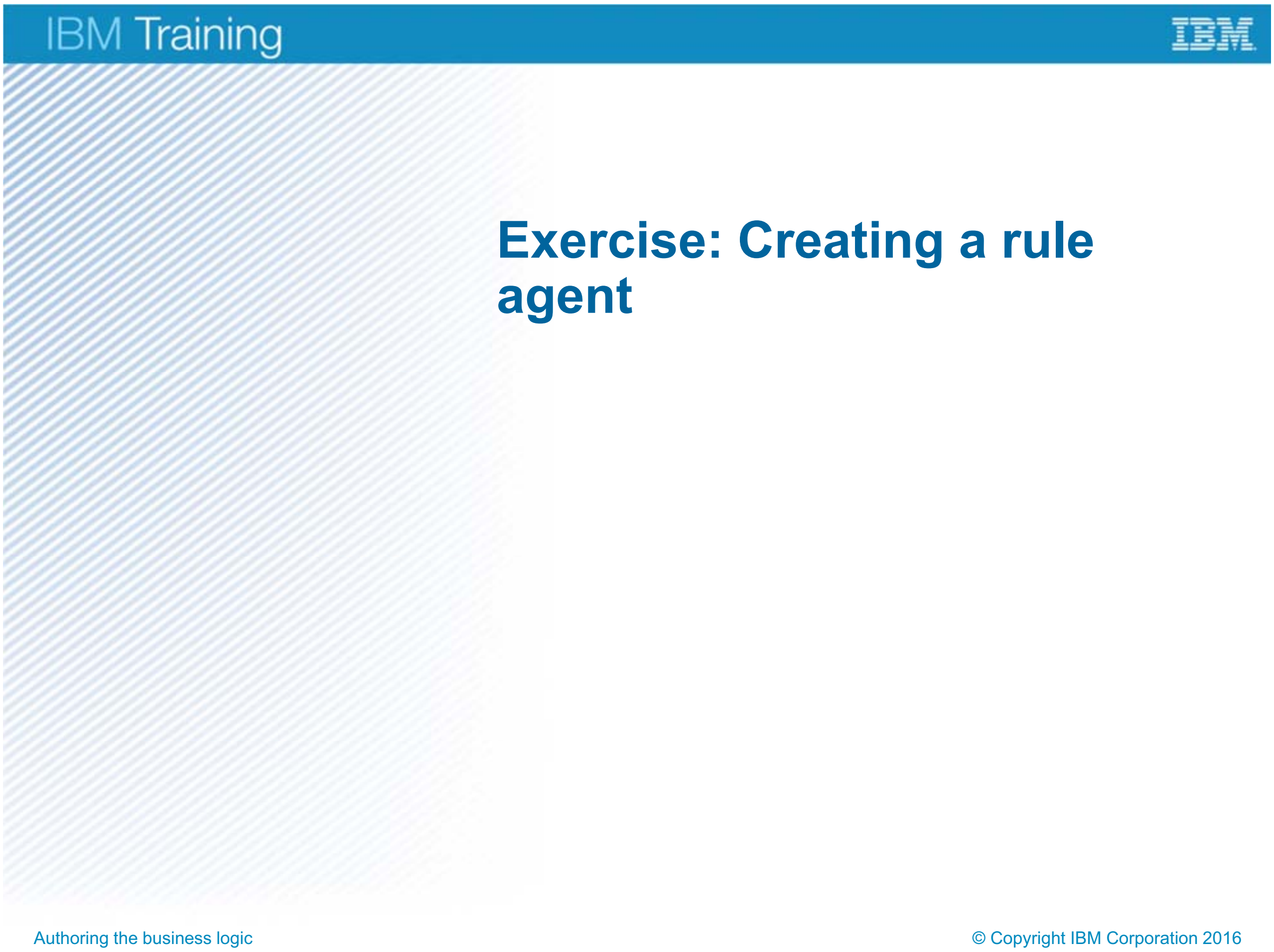


Figure 4-37. Exercise: Creating a rule agent

IBM Training

Exercise introduction

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

Authoring the business logic

Figure 4-38. Exercise introduction

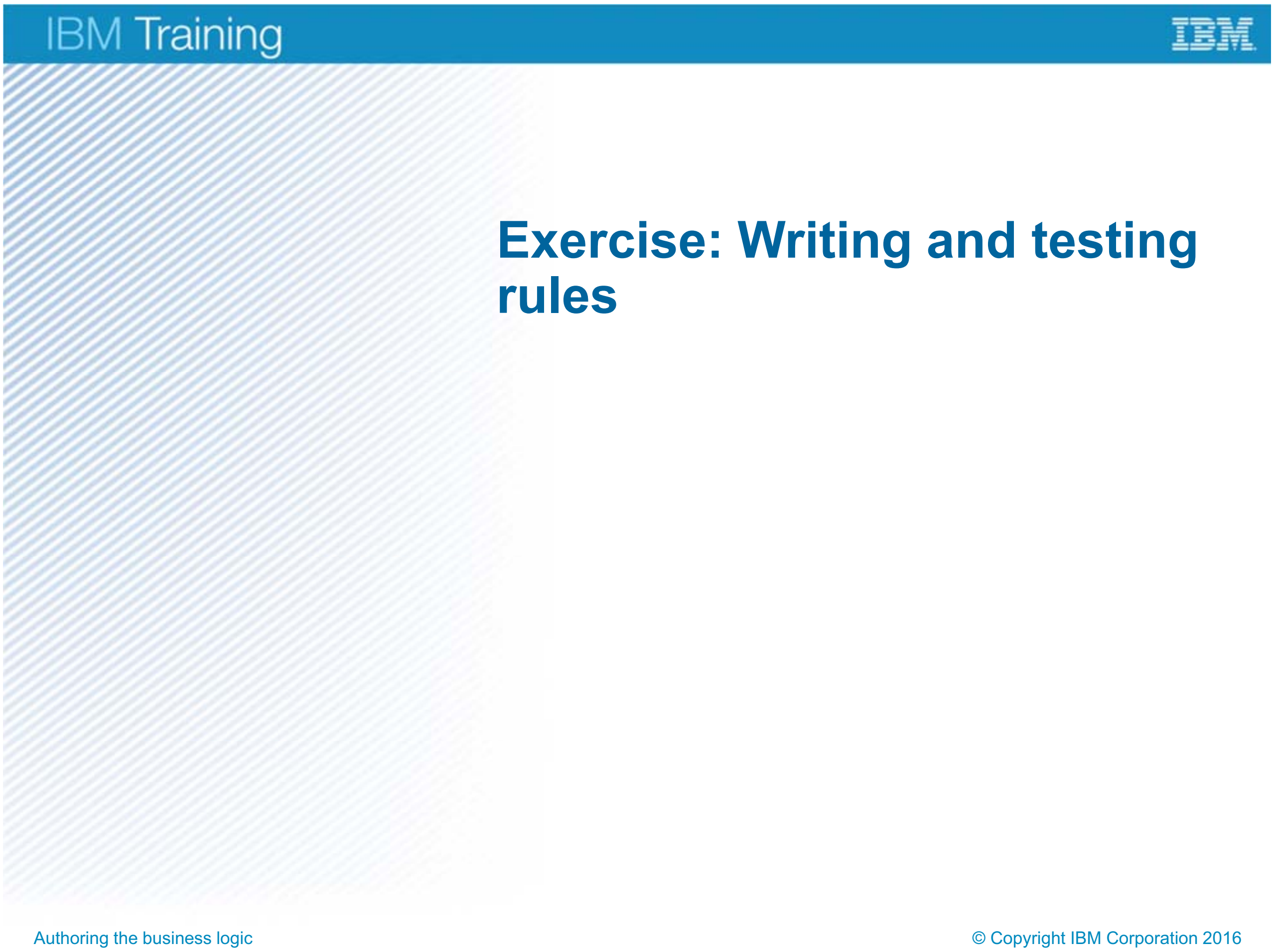


Figure 4-39. Exercise: Writing and testing rules

IBM Training

Exercise objectives

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

Authoring the business logic

Figure 4-40. Exercise introduction

Unit 5. Working with aggregates

Estimated time

00:45

Overview

This unit teaches you how to implement analytics in your business logic by using local, global, and shared aggregates.

How you will check your progress

- Review
- Exercises

IBM Training

Unit objectives

- Define global aggregates
- Define shared aggregates

Working with aggregates

Figure 5-1. Unit objectives

IBM Training

Topics

- Global aggregates
- Shared aggregates

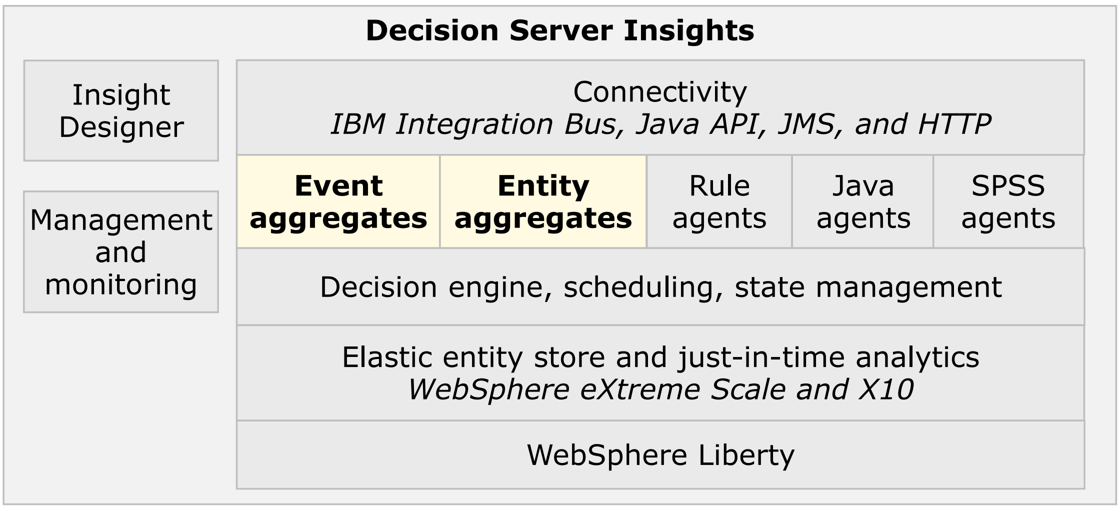
Working with aggregates

Figure 5-2. Topics

IBM Training

Runtime architecture: Aggregates

- Event aggregates
- Entity aggregates



Working with aggregates

Figure 5-3. Runtime architecture: Aggregates

Decision Server Insights includes both a **local** and a **global** programming model for entities and events that allows for global calculations and analysis of all events and all entities in the solution.

Global aggregates are defined as part of a solution in Insights Designer. After deployment, the runtime manages the aggregates.

5.1. Global aggregates

Figure 5-4. Global aggregates

IBM Training

What is a global aggregate?

- Globally calculated values that are derived from a population of entities or events
  - Not about an individual event or entity

- Event aggregates
  - A global event aggregation is evaluated automatically when a new event of that type occurs
  - Calculate the total, minimum, maximum, or average value for a collection of events
- Entity aggregates
  - A global entity aggregate is evaluated on a schedule or when explicitly recalculated by an agent, an MBean, or a script
  - Calculate totals, minimums, maximums, or averages on values for entity attributes over an entire population of entities

- Scalar values
  - What is the average age of all customers?
  - How many accounts were opened last year?

Working with aggregates

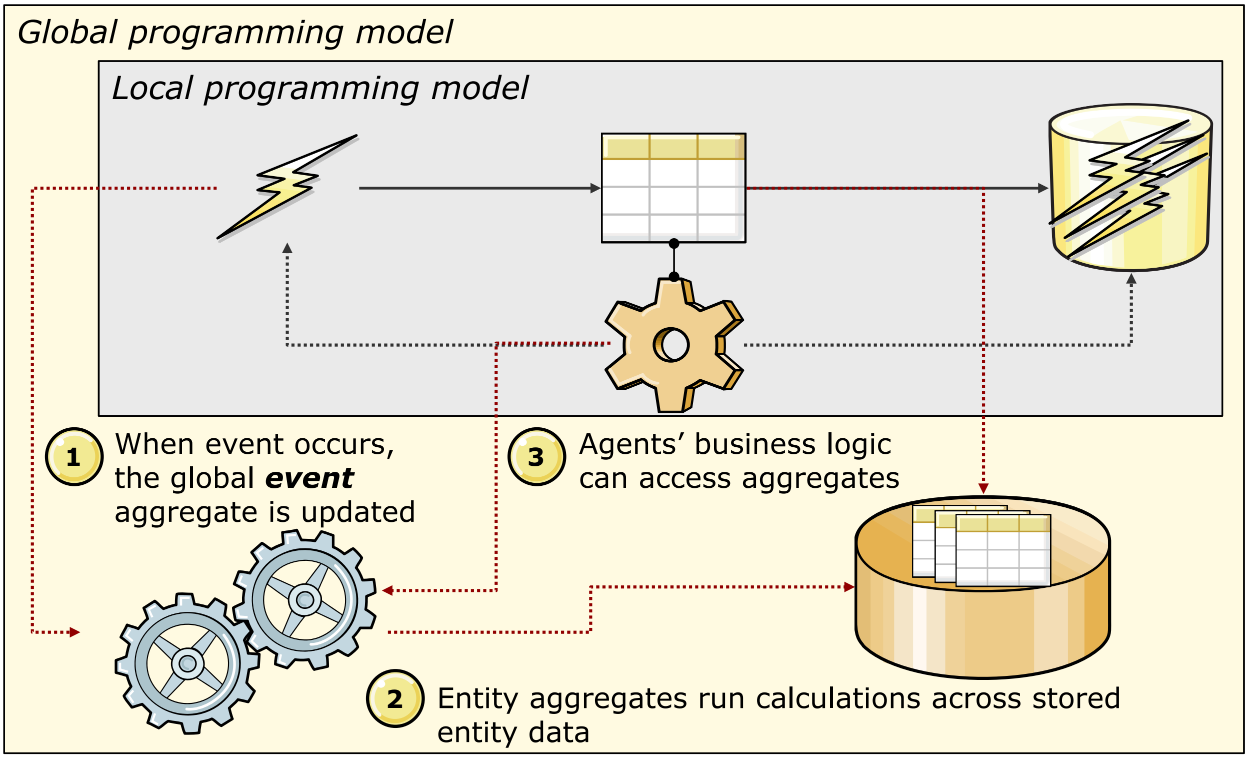
Figure 5-5. What is a global aggregate?

What is a global aggregate? Global aggregates are globally calculated values that are derived from a population of entities or events.

For example, you can look at an entire population of entities, such as all your customers and calculate an average age. Or, you can look at all your accounts to determine how many accounts were opened last year. These types of scalar values can be useful in assessing what types of strategies to pursue or in analyzing your success metrics.

IBM Training

Aggregates in the programming model



Working with aggregates

Figure 5-6. Aggregates in the programming model

Global event aggregates run continuously as part of a deployed solution. The graphic depicts both the local and global programming model. For example, in the local programming model, you can aggregate event information for all events that are delivered to a particular entity (as opposed to all entities). So a local aggregate can calculate the 30-day average transactions for an account entity. However, a global aggregate can calculate the 30-day average transactions of all accounts.

The first step of calculating an event aggregate is that when an event occurs, the global event aggregate is updated. A global event aggregate is calculated every time that an event occurs. So if you have an event aggregate defined for all banking transactions, every time a banking transaction occurs, that banking transaction aggregate is recalculated, and its stored value is updated.

In point number two: global entity aggregates calculate values across all stored entity data. So when an event occurs to an entity the agent monitoring that entity processes the event, and that event data gets stored in the entity context. Entity aggregates have access to all stored context information.

Global aggregate values are accessible to all agents in your solution.

IBM Training

Global entity aggregate

- You define a global entity aggregate by specifying an aggregate expression, an evaluation schedule, and an optional default value
- Use entity aggregates to run, on a schedule, batch jobs that perform global entity calculations

- To define the entity aggregate, specify:
  - The aggregate name, which is the same name as the file
  - The aggregation operator or *the number of* construct
  - The attribute of an entity
  - The evaluation schedule

- Example:

```
define 'Average churn for PLATINUM and GOLD' as the
 average churn score of all clients ,
  where the segment is one of { PLATINUM , GOLD } ,
  evaluated every day at 3:00 PM
```

Working with aggregates

Figure 5-7. Global entity aggregate

To define a global entity aggregate, you specify an aggregate expression and an evaluation schedule. Entity aggregates recalculate their values on a schedule, such as daily or weekly.

In the lab, you see the syntax for an aggregate definition, which includes the aggregate name, the operator, the attribute of the entity that you want to calculate, and the evaluation schedule.

For example, as you see here, to calculate the average churn, the aggregate value is calculated across a subset of client entities. The calculation is only for clients where the segment attribute is set to either platinum or gold. This calculation is run every day at 3 PM.

IBM Training

Global event aggregates

- You define a global event aggregate by specifying an aggregate expression, an optional time filter, and an optional default value
  - Can find **total**, **minimum**, **maximum**, or **average** value of a collection of events, or the number of events
- Global events run continuously as part of a deployed solution
  - Continuously performs calculations on incoming events
  - Calculated every time one of the events occurs
- To define the event aggregate, specify:
  - The aggregation operator or *the number of* construct
  - The attribute of an event
  - Example:

```
define 'average_delay' as the average delay of all flight_delay
events, where the delay of each flight_delay event is more than
10
```

Working with aggregates

Figure 5-8. Global event aggregates

To define a global event aggregate, you again specify the aggregate expression, and an optional time filter.

The aggregate can calculate totals, minimums, maximums, or average values for collection of events or a specific number of events.

The global event aggregates run continuously to perform calculations on the incoming events. They are recalculated every time one of the specified events occurs.

Again during the lab, you see how to define an event aggregate by following the syntax that you see here. For example, the average to the aggregate is calculated as the average delay of all flight delay events where the delay of each flight delay event is more than 10. The events that are calculated are filtered according to the delay attribute.

IBM Training

Global aggregate functions

- **Count**
  - *Define 'transaction count' as the **number** of Transactions*
- **Sum**
  - *Define 'transactions total' as the **total** amount of all Transactions*
- **Average**
  - *Define 'average transaction amount' as the **average** amount of all Transactions*
- **Min**
  - *Define 'min amount of all transactions' as the **minimum** amount of all Transactions*
- **Max**
  - *Define 'max amount of all transactions' as the **maximum** amount of all Transactions*

Working with aggregates

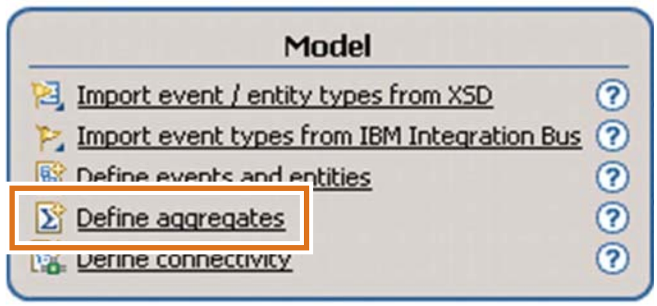
Figure 5-9. Global aggregate functions

Here, you see the predefined aggregate functions that you can use when defining your global aggregates.

IBM Training

Defining aggregates

- You create global aggregate definition (`.agg`) files to define how the aggregate should work
- The name of the aggregate is the same as the name of the `.agg` file
- Aggregate files are created in the **aggregates** folder of the BOM project
- Use the link in the **Model** goal of the **Solution Map** to create an aggregate



Working with aggregates

Figure 5-10. Defining aggregates

You define a global event aggregate by composing an expression in the `.agg` file that has the following syntax:

```
define '<same_name_as_file>' as <expr>
```

If events occurred for too short a time, a global event aggregate might not have a meaningful value. Therefore, you can specify a default value and the condition for when the default applies. The default value applies unless enough event history exists for the aggregate calculations, and any rules that are based on them, to have meaningful results. You might need to test and experiment with the default value and the condition to obtain optimal results for your particular solution.

The event history is the elapsed time after the first event of any type occurs in a deployed solution. If you change the definition of the global event aggregate and deploy a new version of the solution, the elapsed time is reset to zero and restarts after the first event of any type occurs.

To create a global aggregate, you can use the **Define aggregates** link in the Model goal of the Solution Map.

When you create a global aggregate, a `.agg` file is generated for you to define how the aggregate should work.

The name of the aggregate is the same as the name of the `.agg` file. Aggregate files are created in the aggregates folder of the BOM project.

IBM Training

Retrieving global aggregate value

- Accessible from rules

```
definitions
 set AVG_CHURN to 'Average churn for GOLD' ;
if
 the churn score of 'the client' is at least 1.5 * AVG_CHURN
```

- Accessible from Java agents through the Java agent API

```
getGlobalValue(aggregateName)
```

- Accessible from REST
  - To display all aggregates:

    `https://<host>:<port>/ibm/ia/rest/solutions/<solution>/aggregate`

  - To display a particular aggregate:

    `https://<host>:<port>/ibm/ia/rest/solutions/<solution>/aggregate/<aggName>`

Working with aggregates

Figure 5-11. Retrieving global aggregate value

Aggregate names are in the BOM project. You can also open the `globalQueries` variable set in the **aggregates** folder to see the list.

Your global aggregate values are visible to all the agents in the solution, so you can access the aggregate values in your rules. You can also access the aggregate in a Java agent by using the Java agent API. In addition, you can use the REST API to view your aggregates and their values.

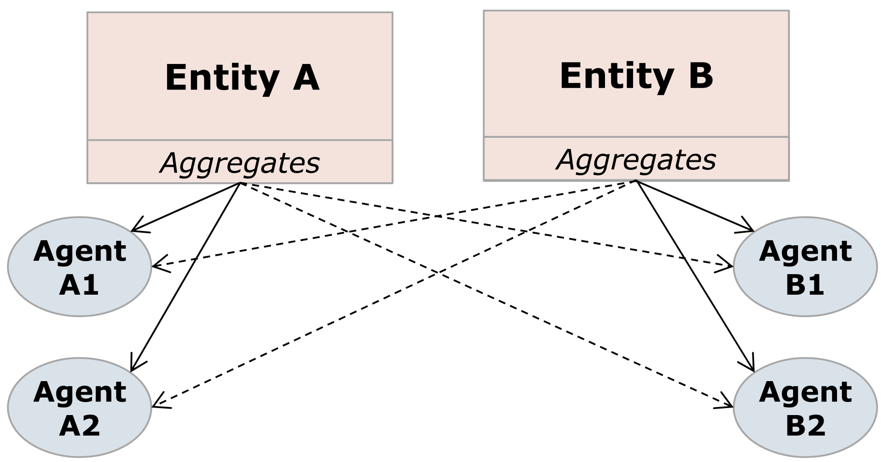
5.2. Shared aggregates

Figure 5-12. Shared aggregates

IBM Training

What is a shared aggregate?

- Aggregates events that are associated with an entity
- Usable from any agent, local or remote to the entity
  - Events subscribed to independently from agents
- Can use the same aggregate to obtain values in multiple timeframes
- Sharing results in increased performance



Working with aggregates

Figure 5-13. What is a shared aggregate?

Shared aggregates are used to aggregate events that are associated with a particular entity.

You define a shared aggregate in the business model as a normal entity attribute. You use the business model statements to define how the aggregation should be evaluated.

The aggregate is generated and shared with any agent, either local or remote to that entity. As shown in the graphic, Agents A1 and A2 are defined for Entity A. Agents A1 and A2 can access the shared aggregates that are defined on both Entity A and Entity B. The agents are also defined for Entity B as long as they have a reference to Entity B.

A shared aggregate is somewhat like an index of a database; it can be used to access quickly any value within the range of the horizon. Getting a value at a specified moment in time (not necessarily the current moment) can be used to detect when a change in the value occurs over time.

IBM Training

Referenced with an argument

- Uses a time argument when referenced in a rule
- Eliminates the need for separate aggregates for each time frame
- Allows for comparisons across time frames
- Argument can be a time point or time period
- Time periods cannot be used in aggregates with time filtering
- Time points default to "now" of the rule

Figure 5-14. Referenced with an argument

Unlike other aggregate types, shared aggregates are referenced with a time argument when they are used in a rule. The argument can be either a time point or a time period. By default, if no time point is specified, the time point defaults to “now”.

The ability to use a single-shared aggregate for multiple time frames eliminates the need to define aggregates for each time frame. You can easily use the aggregate to compare values from different time frames, for example, comparing a total from “last month” to the same month from the previous year.

IBM Training

Defining shared aggregates

- Declared as an attribute of an entity in the business model
  - Example:
    `a train` *`has a`* *`total seven day delay`* *`(numeric)`*.
- Can aggregate average-of, maximum-of, minimum-of, number-of, and total-of
- Defined in statements on **Statements** tab of editor
  - Example:

```
the total seven day delay of a train is aggregated from train
delays,
  where this train comes from the train of each train delay as
the total delay of all train delays during the last period of 7
days
available for 1 month.
```

Working with aggregates

Figure 5-15. Defining shared aggregates

Here you see an example of how to work with shared aggregates.

First, the shared aggregate is defined in the same way that you define any attribute on an entity. In this example, the train entity has the shared aggregate: `total seven day delay`

This value is aggregated from all *train delay* events. The *total* operation sums the *delay* attribute values.

The argument that is passed to the aggregate is the time point “now” and the value is calculated from the 7 days previous to the time point.

The system keeps the data for this aggregate available for one month.

IBM Training

Time filtering

- Any explicit or implicit reference to "now" is time filtering
  - With time filtering:

```
the longest monthly delay of a train is aggregated from train
delays,
   where this train comes from the train of each train delay
as the maximum delay of all train delays during the current
month
available for 1 year.
```

  - Without time filtering

```
the average weekday delay of a train is aggregated from train
delays,
   where this train comes from the train of each train delay
as the average delay of all train delays,
   where the day of week of each train delay is not one of {
Saturday, Sunday }
available for 1 year.
```

Working with aggregates

Figure 5-16. Time filtering

Time filtering is an implicit or explicit reference to “now” in the definition of the aggregate.

In the first example, the term “during the current month” is an implicit reference to now. According to the rule language, “the current month” is the month that contains “now”.

A time period argument cannot be used with time filtering.

The second example does not have time filtering. The term “where the day of week” is a reference to the time of the event. The rule is looking for an event on a weekday, which is not based on “now”.

The use of time is not equivalent to time filtering, which is always in reference to “now”. The time period that is passed as an argument replaces time filtering.

IBM Training

Rules with time points

- Use “at” to specify the query time point
- The query time point defaults to "now" in the rule
  - "Now" in the aggregate definition is the query time point

```
when a train delay occurs
if
    the total seven day delay of 'the train' is more than
    the total seven day delay of the train at the time of 1 day
before the departure time of the train
then
    emit a new alert where
    the message is "The delays are trending the wrong way!" ;
```

Working with aggregates

Figure 5-17. Rules with time points

Here, you see how to the aggregate is referenced in a rule.

This rule references the aggregate twice. No time reference is passed to the first reference, which means the aggregate calculation is based on the default time point, “now”.

In the second reference, the time period “1 day before the departure time of the train” is passed as the time argument.

IBM Training

Flexibility with time periods

- Use “over” to specify a time period
  - "Now" does not apply to time period queries

```
when a ticket purchase occurs
definitions
  set 'average delay last month' to the average weekday delay of 'the train'
  over the calendar month of 1 month before now;
  set 'average delay last year' to the average weekday delay of 'the train'
  over the calendar year of 1 year before now;
  set improvement to (1 - ('average delay last year'
  / 'average delay last month') * 100);
if
  'average delay last month' is less than 'average delay last year'
then
  emit a new email where the message is "Thank you for your purchase. We're
  proud to have achieved " + 'improvement' + "% improvement in commuter
  service over last year.";
else
  emit a new email where the message is "Thank you for your purchase.";
```

Working with aggregates

Figure 5-18. Flexibility with time periods

Time period arguments replace time filtering.

In the example here, the same aggregate is used over two different time periods to compare the last calendar month to last calendar year.

IBM Training

Resolution

- Increase performance with long event histories
- Result is approximate
- Not available with time filtering

```
the average weekday delay of a train is aggregated from train
delays,
   where this train comes from the train of each train delay
as the average delay of all train delays,
   where the day of week of each train delay is not one of {
Saturday,
Sunday }
available for 4 years with a resolution of 24 hours.
```

Working with aggregates

Figure 5-19. Resolution

Resolutions are defined in the `available-for` statement and cause the system to internally bundle aggregate data. By bundling the data, the system maintains much less data.

Including a resolution clause is useful when the aggregate runs over long time periods. Resolutions are a tradeoff between performance and accuracy.

In this example, the aggregate runs over a period of several weeks. The `available-for` period includes a 24-hour resolution. The resolution causes the system to bundle all the events in the past 24 hours, so the bundling means that the accuracy is plus or minus a day.

IBM Training

Default values

- If you do not specify a default value, the default value of shared aggregates is null
  - If there are fewer than the minimum quantity data points at the requested time parameter, a null value is returned
  - For `number-of` operations, an actual numeric value is returned even if empty (0)
- When the oldest point of an aggregate query is older than the horizon, the value returned is null
- Null values can cause rules that use the shared aggregate to not fire
- Specify a default value to avoid null pointer errors
  - Example: Default is 5

```
the longest monthly delay of a train is aggregated from train
delays,
   where this train comes from the train of each train delay
as the maximum delay of all train delays during the current
month
defaulting to 5 if there are less than 3 events
available for 1 year.
```

Working with aggregates

Figure 5-20. Default values

The default value of a shared aggregate is null. Null values might cause a rule that uses the shared aggregate to not fire or execute.

To avoid null value issues, you can specify a default value. Default values do not affect the `number-of` operator, which always returns a numeric value, even if the aggregate is empty.

In the example here, the default value is set to 5.

IBM Training

Unit summary

- Define global aggregates
- Define shared aggregates

Figure 5-21. Unit summary

IBM Training

Review questions

1. True or false: In Java agents, a separate API accesses event and entity aggregates.

2. True or false: A global entity aggregate can perform calculations across entire populations of entities.

Figure 5-22. Review questions

Write your answers here:

1.

2.

IBM Training

Review answers

1. In Java agents, a separate API accesses event and entity aggregates
Answer: False. You use the same API to access both event and entity aggregates.

2. A global entity aggregate can perform calculations across entire populations of entities
Answer: True.

Figure 5-23. Review answers

IBM Training

Exercise: Using global aggregates in rules

Working with aggregates

Figure 5-24. Exercise: Using global aggregates in rules

IBM Training

Exercise objectives

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

Working with aggregates

Figure 5-25. Exercise introduction

Figure 5-26. Exercise: Using event aggregates in rules

IBM Training

Exercise objectives

- Use event aggregates and shared aggregates in rules

Working with aggregates

Figure 5-27. Exercise introduction

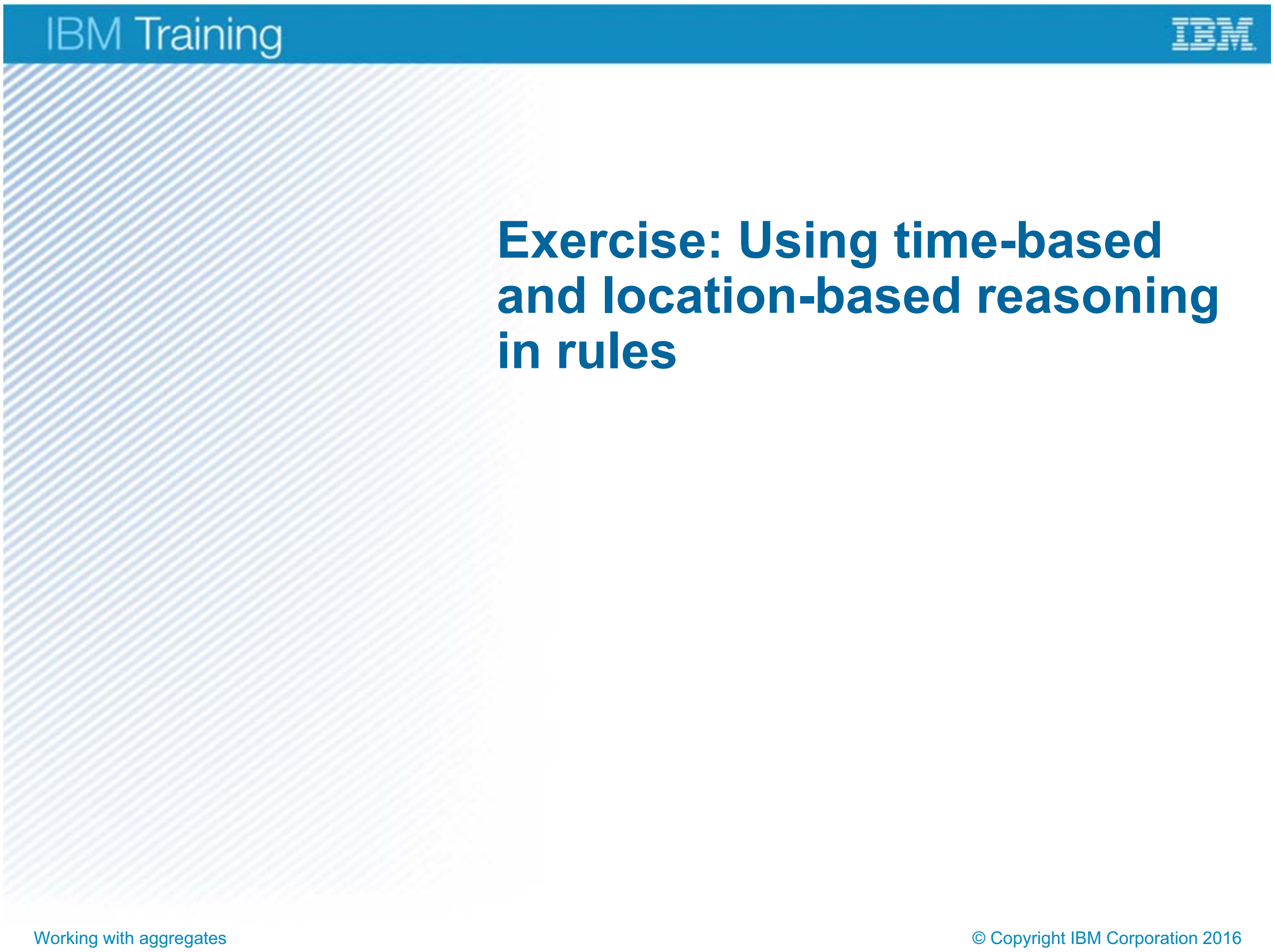


Figure 5-28. Exercise: Using time-based and location-based reasoning in rules

IBM Training

Exercise objectives

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

Working with aggregates

Figure 5-29. Exercise introduction

IBM Training

Exercise: Testing for the absence of events

Working with aggregates

Figure 5-30. Exercise: Testing for the absence of events

IBM Training

Exercise objectives

- Test for the absence of events

Working with aggregates

Figure 5-31. Exercise introduction

Unit 6. Testing solutions

Estimated time

01:00

Overview

This unit teaches you how to test the implementation of your business logic.

How you will check your progress

- Checkpoint

IBM Training

Unit objectives

- Test solutions with the TestDriver API
- Test solutions with the Test Client
- Work with the REST API
- Analyze event processing with Insight Inspector
- Work with the Map Viewer

Testing solutions

Figure 6-1. Unit objectives

IBM Training

Topics

- TestDriver API
- Test Client
- REST API
- Insight Inspector
- Insight Map Viewer

Testing solutions

Figure 6-2. Topics

6.1. TestDriver API

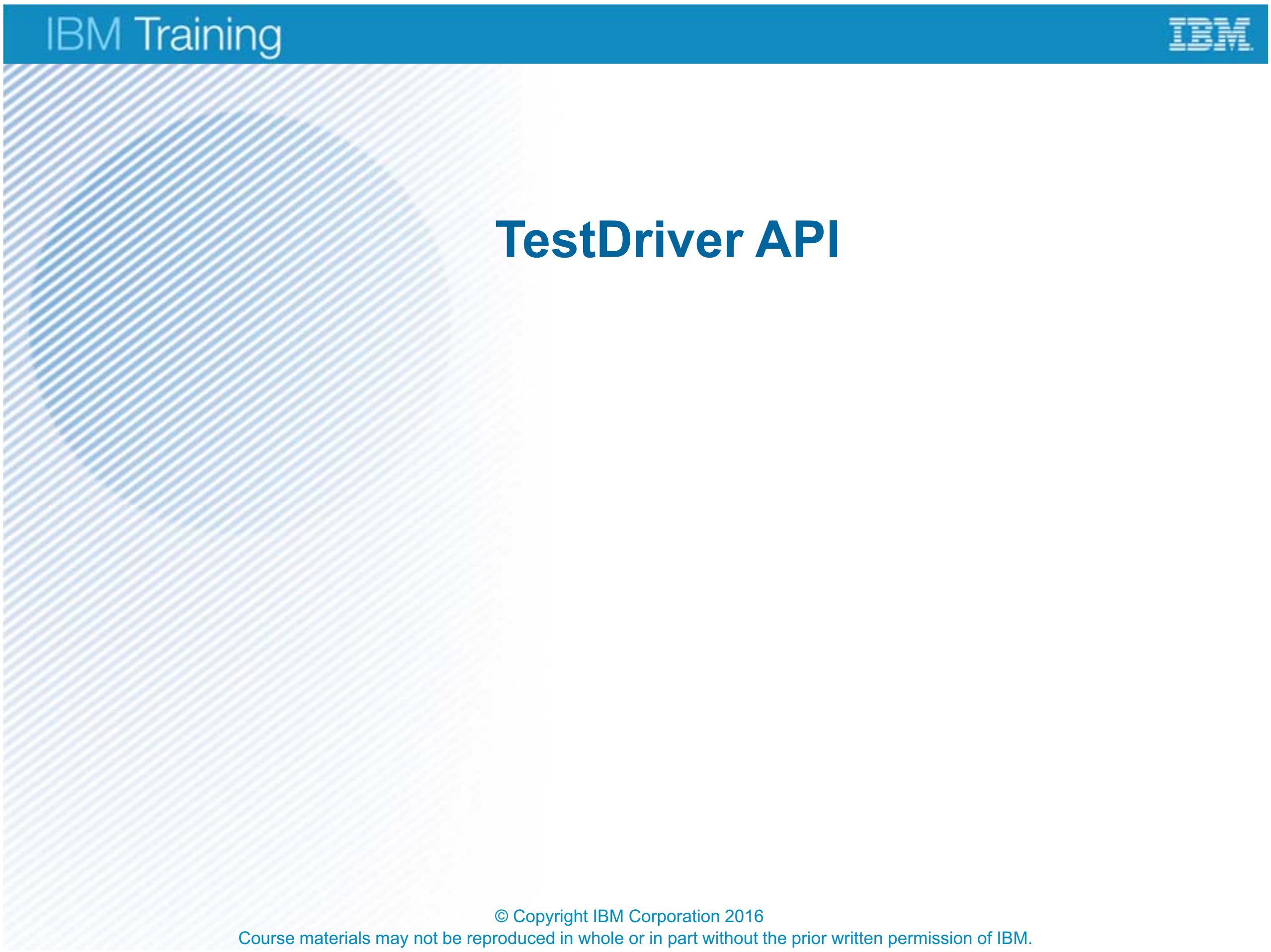


Figure 6-3. TestDriver API

IBM Training

TestDriver API

- Java TestDriver API provides all methods that are required to test a solution
  - Test and debug solutions
  - Manage entities
  - Determine the system and solution status
  - Create and submit events to Insight Server
- The TestDriver class is the main entry point for testing a solution
- Use model factory methods such as getConceptFactory and getEventFactory to create events, entities, and concepts
- Use TestDriver API to capture events for debugging and problem determination

Testing solutions

Figure 6-4. TestDriver API

You can test your Insights solutions in several ways, such as by using the TestDriver API to load test entities into the grid and submit test events for processing.

A test driver helps you run solution tests and debugging, manage entities, determine the system and solution status, and create and submit events to Insight Server. TestDriver provides model factory methods, such as **getConceptFactory** and **getEventFactory** to create events, entities, and concepts.

The TestDriver API provides the **DebugReceiver** interface to help you capture outbound information from test events for debugging and problem determination.

TestDriver properties

- Configured within a Java Properties object that contains property/value pairs for the various properties
- Use a `testdriver.properties` file to
  - Connect the test driver instance to a server
  - Create and add properties to modify the behavior of the test driver

Figure 6-5. TestDriver properties

To use TestDriver, you create a Java project and a test driver instance.

To ensure that a test driver instance can connect to a server, you create a `testdriver.properties` file and add properties to configure the behavior of the test driver.

The **`testdriver.properties`** file contains connectivity and server properties to set up and run the test driver.

The **`testdriver.properties`** file must be located either in the directory where the Java application starts, or in a location that is defined by the **`TESTDRIVER_HOME`** environment variable.

IBM Training

Example testdriver.properties

```
solutionname=MyTestDriverProject
catalogServerEndpoints=localhost:2809
host=localhost
port=9443
debugServers=localhost:6543
username=tester
password=tester
trustStoreLocation=<InstallDir>\\runtime\\wlp\\usr\\servers\\
cisDev\\resources\\security\\key.jks
trustStorePassword=tester
disableSSLHostnameVerification=true
```

Testing solutions

Figure 6-6. Example testdriver.properties

Here, you see the `testdriver` properties and some example values.

The **`solutionName`** property is a required property and specifies the solution where the test driver instance runs. The other properties are optional or use default values.

The **`username`** and **`password`** properties must match the **`username`** and **`password`** properties in the **`server.xml`** file.

The **`trustStoreLocation`** property uses the default installation path for the keystore location. If you share projects and test drivers with users on other computers, they might need to modify this property to match their installation path.

IBM Training

Insert entities and submit events (1 of 2)

- In your class, create a TestDriver instance and code a connection to your solution

```
TestDriver testDriver = null;
testDriver = new TestDriver();
testDriver.connect();
```

- Define a method to insert your entities to the grid

```
private void loadCustomer() throws Exception {
Customer customer =
testDriver.getConceptFactory(ConceptFactory.class).createCustomer
("Smith");
customer.setFirstName("Jack");
customer.setLastName("Smith");
customer.setLoyaltyCardOwner(true);
testDriver.loadEntity(customer);
}
```

Testing solutions

Figure 6-7. Insert entities and submit events (1 of 2)

To insert entities and submit events to your solution, you first create a TestDriver instance and define a connection to your solution by using the code that you see here.

Next, you define a method to insert your entities to the grid and this method would create the entity and initialize it with some values.

IBM Training

Insert entities and submit events (2 of 2)

- Create an event by using the EventFactory API

```
CreateCartEvent createCartEvent =
testDriver.getEventFactory().createEvent(CreateCartEvent.class);
createCartEvent.setShoppingCartId(shoppingCartId);

Relationship<Customer> customerRel =
testDriver.getEventFactory().createRelationship(Customer.class,
customerId);

createCartEvent.setCustomer(customerRel);
```

- Send the event in a `TestDriver.submitEvent` method

```
testDriver.submitEvent(createCartEvent);
```

Testing solutions

Figure 6-8. Insert entities and submit events (2 of 2)

The next step is to create an event by using the EventFactory API. Finally, you submit the event by using the TestDriver **`submitEvent`** method.

IBM Training

Receiving and storing debug information (1 of 2)

- Run the `propertyManager set` command to configure the server debug port property, or range of ports
  - For example:

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

- Use the `DebugReceiver` interface to create a debug receiver instance
  - The interface defines an entry point for `DebugInfo` by using the `addDebugInfo` method
  - The `addDebugInfo` method is called when the server sends debug information to the test driver client
  - For example:

```
addDebugInfo( DebugInfo info, String sourceAgent)
```

  - In the example, the `info` parameter represents the debug information and `sourceAgent` is the name of the agent that generated the information

Testing solutions

Figure 6-9. Receiving and storing debug information (1 of 2)

After testing event processing by submitting events, the TestDriver API also helps you monitor and capture event debug information. You can use the TestDriver to collect the debug information from the server in memory or write it to a file.

As you saw earlier, the **`testdriver.properties`** file contains connectivity and several properties to set up the TestDriver instance and control solutions and information. You can also identify which agents should send information by using the debug agent list property. You set this property to a comma-separated list of your agent names. By default, you get debug information from all the agents in the solution.

During the first lab of this course, you set the debug port as described on the slide. The TestDriver uses the host port value to receive the debug information.

In your TestDriver instance, you use the debug receiver interface to capture outbound information from the events for debugging and problem determination.

IBM Training

Receiving and storing debug information (2 of 2)

- Start with the sample implementation `IADebugReceiver` and then customize the debug receiver to work in your environment
- Add the debug receiver instance to the test driver client by calling the `addDebugReceiver` method in the client
- For example:

```
TestDriver testDriver = null;
testDriver = new TestDriver();
DebugReceiver r = new IADebugReceiver();
testDriver.addDebugReceiver( r );
testDriver.connect();
```

Testing solutions

Figure 6-10. Receiving and storing debug information (2 of 2)

When the connection to the server is established, the server detects all events that are emitted by specific solutions and agents that match the **`solutionname`** and **`debugagentlist`** properties in the **`testdriver.properties`** file.

The server then sends debug information from the events to the TestDriver instance. The TestDriver instance calls the debug receiver, which captures the information and caches it in memory or stores it in a file.

You can start with a sample implementation of the **`IADebugReceiver`** and then customize the receiver for your environment. To add the debug receiver instance to a TestDriver client you call the **`addDebugReceiver`** method in the client, as you see in this example.

6.2. Test Client

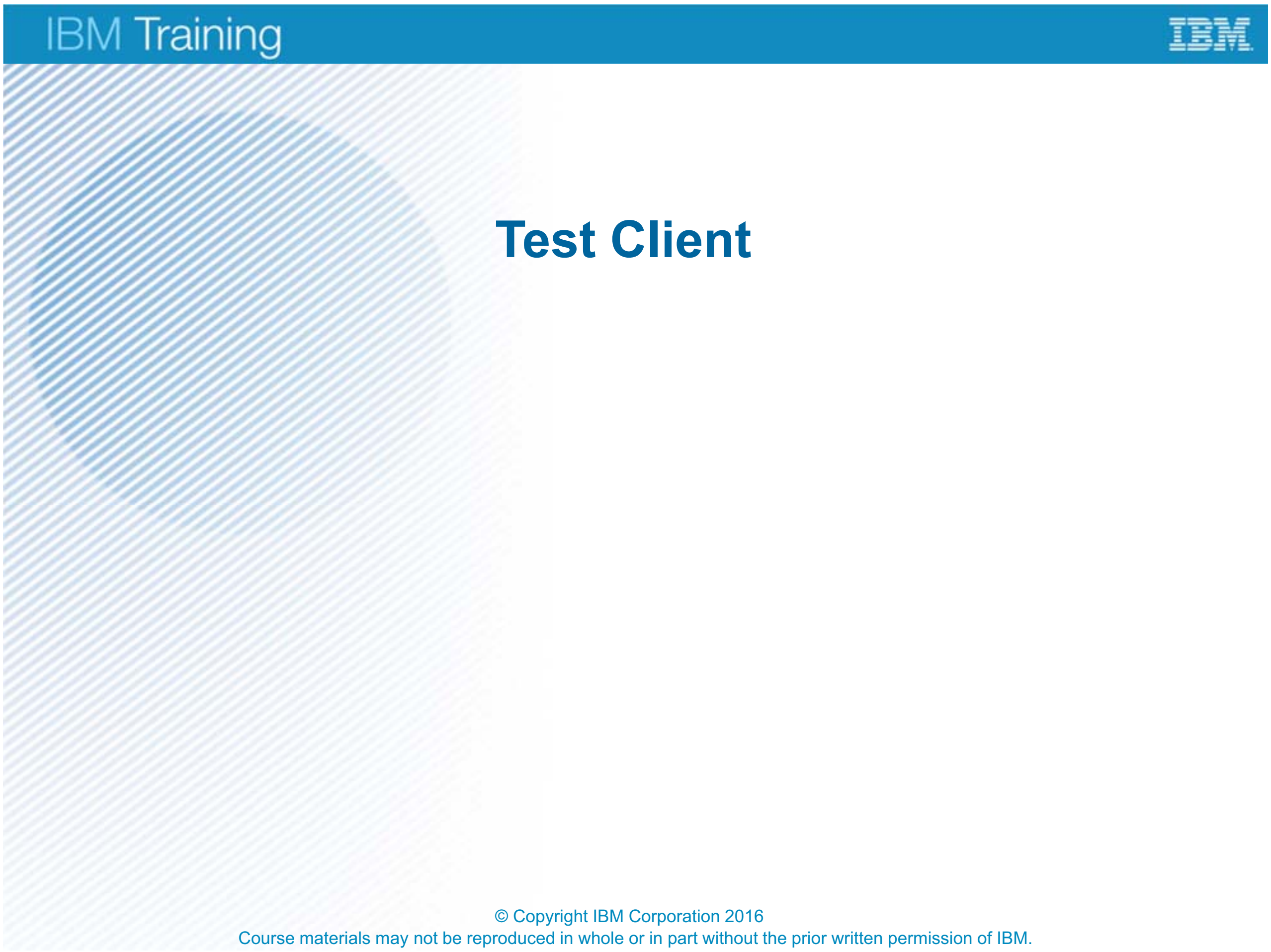


Figure 6-11. Test Client

IBM Training

Using a test client

- Use test clients to:
  - Define and load entities
  - Create and submit a sequence of events
  - Write and run assertions about the state of the entities
- No Java required

Testing solutions

Figure 6-12. Using a test client

The test client provides solution developers with simplified tools to verify the behavior of their solution.

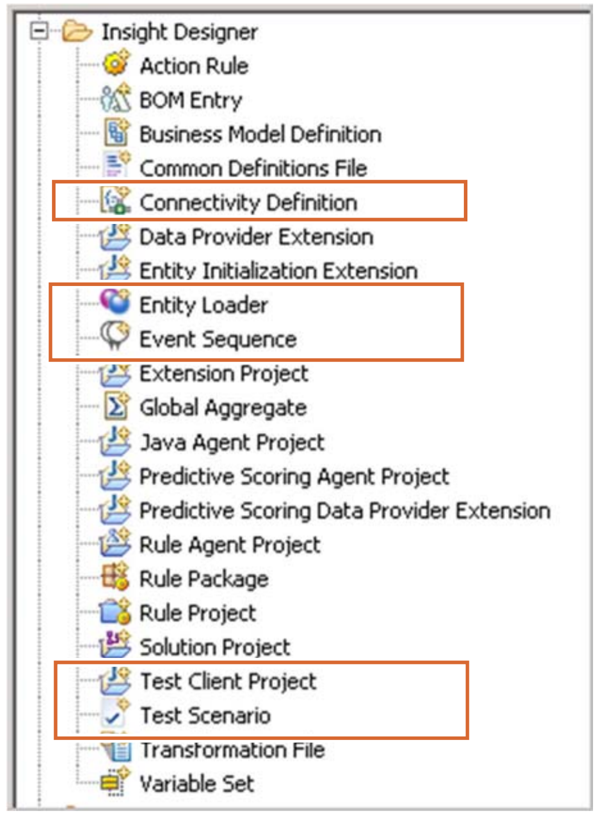
Writing test cases with Java and XML provides power and flexibility, but can be difficult for quick experimentation for non-technical users.

The test client provides an easy way to quickly load entities and submit some events. You can then use the test client to verify the state of your system as a result of those actions, and know whether your solution is working.

IBM Training

Test Client artifacts

- Test artifacts include:
  - Common definitions
  - Entity loaders
  - Event sequences
  - Test scenarios
- Artifacts are stored in a **Test Client Project**



Testing solutions

© Copyright IBM Corporation 2016

Figure 6-13. Test Client artifacts

Insight Designer provides various artifacts to enable users to test their solution.

- **Entity loaders** allow the user to define entities to load.
- **Event sequences** allow the user to define an ordered sequence of events.
- **Test scenarios** allow the user to co-ordinate their entity loaders, event sequences and to write assertions about the state of their entities.
- **Common definitions** files allow the user to create definitions that can be used throughout the other testing artifacts.
  - The languages for all of these assets are based around a subset of the rule language.
  - Entity loaders, event sequence, and test scenarios can all be run on a development server, by using Eclipse run configurations, without requiring the user to write any Java code.
  - If recording is enabled in the run configuration, Insights Inspector can be used to review any submitted events. The REST API can be used to review the loaded entities.

IBM Training

Creating a test client project

- All testing artifacts stored in a separate Test Client project
  - Use **File > New > Other > Insight Designer** menu
- Connection details for the server to run the test artifacts on can be found in the `testdriver.properties` file within the Test Client project
  - Modify properties to match the development server
- Run a testing artifact by right-clicking on the file or editor and select “**Run as**”
  - Modify your run configuration later by using the Run Configurations window

Testing solutions

Figure 6-14. Creating a test client project

All artifacts that you need to test the solution are stored together in a separate test client project.

To run test artifacts, you right-click your file and select **Run as** with the appropriate menu option.

IBM Training

Entity loaders

- Use the “load” keyword to load entities into a development server.
- Define the initial attributes of the entity, so it is in the state that you require at the beginning of your test.

```
load a new customer where
  the customer id is "123" ,
  the name is "Betty" ,
  the address is a new address where the house number is "21" ,
  the street name is "Hursley Park Road" ,
  the city is "Winchester" ;
```

Testing solutions

Figure 6-15. Entity loaders

You can load multiple entities in the loader. In the example here, the entity loader defines the customer entity and required attributes.

IBM Training

Event sequences

- Use the “emit” keyword to emit events in the sequence that you would like them to be submitted to the server
- After you emit an event with a time-stamp, subsequent events can be emitted with timestamps set relative to the previous event

```
emit a new transaction where the amount is 50 , the customer is
the customer "Fred", time-stamped 10/01/2015;

emit a new transaction where the amount is 100, the customer is
the customer "Fred" , time-stamped 1 week later;
```

Testing solutions

Figure 6-16. Event sequences

The emit keyword emits the events in the sequence you specify. You can include time stamps along with relative times set relative to the previous event.

IBM Training

Test scenarios

- Coordinate loading entities and running event sequences
- Allow you to write assertions about the state of the entities in the system at various time points
- Can simulate the passing of time
  - Ask the system to continue processing until a particular point in time
  - Supports testing scheduled rules and time conditions

```
load entities from "Customers" ;

check that the customer "Fred" exists ;

check that for the customer "Fred" :
-   the customer id of this customer is
    "123"
-   the name of this customer is "Fred"
-   the status of this customer is BRONZE ;

submit events from "Fred transactions" ;

check that for the customer "Fred" :
-   the status of this customer is SILVER ;

continue processing until 1/10/2016 ;

check that for the customer "Fred" :
-   the status of this customer is BRONZE ;
```

Testing solutions

Figure 6-17. Test scenarios

You can use the test scenarios to coordinate the event sequences and write assertions about entities at various time points.

Test scenarios can simulate the passing of time so that you can test scheduled rules and time conditions.

IBM Training

Common definitions

- Definitions that are defined in a Common Definitions file can be used in any of the other test artifacts
  - Use Common Definitions file to define complex structures, such as lists
  - Use the definitions in the entity loader, event sequence, and test scenario artifacts

```
define 'Fred transaction' as a new
transaction where
  the customer is the customer Fred ;

add a new order where
  the order id is "abc" ,
  the description is "Item 1"
to the orders of 'Fred transaction' ;

add a new order where
  the order id is "abc" ,
  the description is "Item 2"
to the orders of 'Fred transaction' ;
```

- Access all the definitions by including this statement:

```
using definitions from <common definitions filename> ;
```

```
using definitions from "Fred definitions" ;

emit 'Fred transaction', time-stamped 10/04/2015 ;
emit 'Fred transaction', time-stamped 1 day later;
```

Testing solutions

Figure 6-18. Common definitions

You can use a common definitions file to define IDs or complex structures that you do not want to repeatedly define, such as lists of concepts.

To include these definitions in other files, you include a statement at the beginning of your other files that makes the definitions accessible.

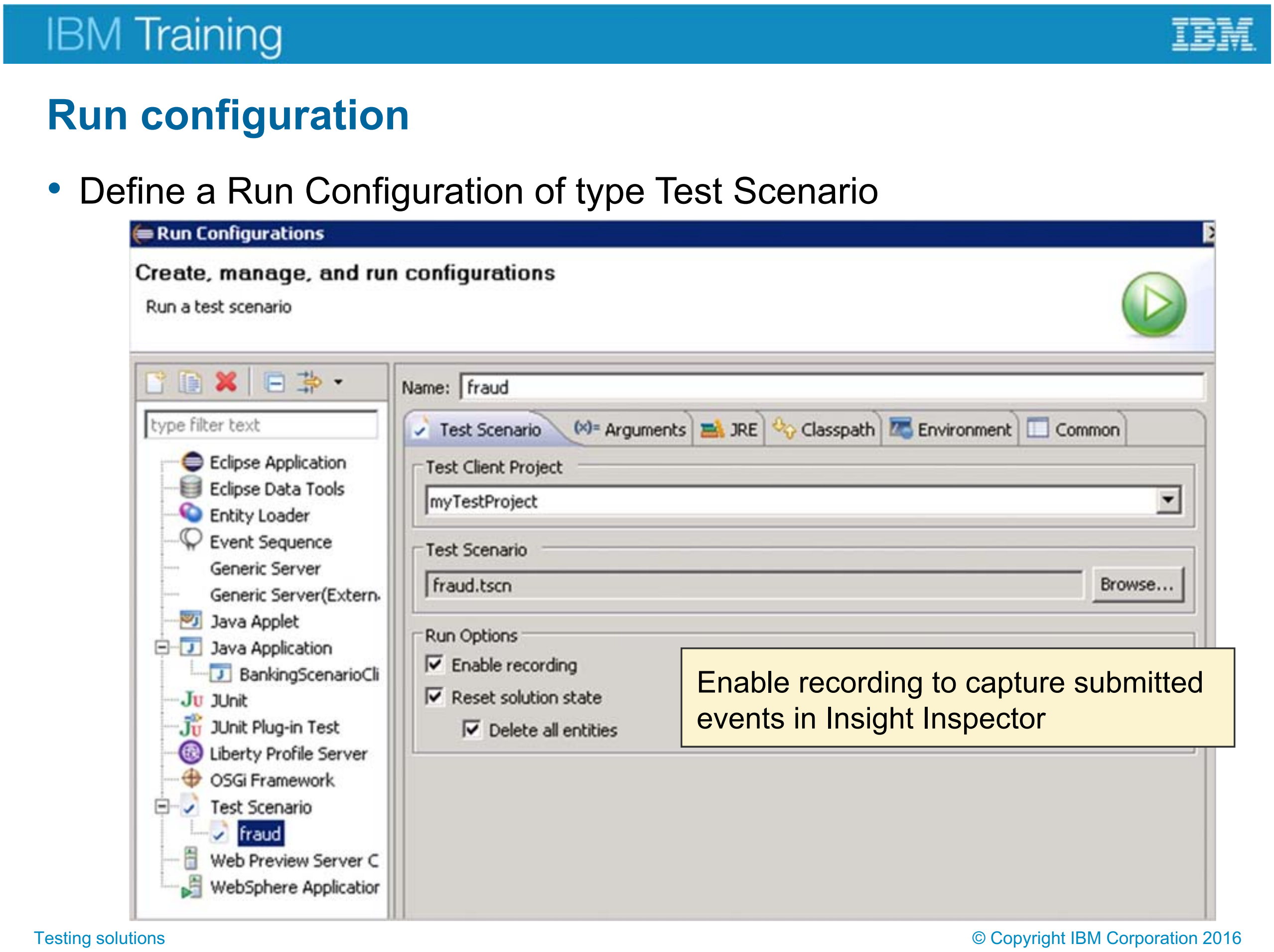


Figure 6-19. Run configuration

The testdriver.properties file within the Test Client project is used to connect to the server when you run the Test Scenario.

In the **Run Options**, you can enable a recording of the test run to analyze in Insight Inspector. The recording is automatically started and stopped for you. After the test is done, a link to the recording is provided in the console.

6.3. REST API

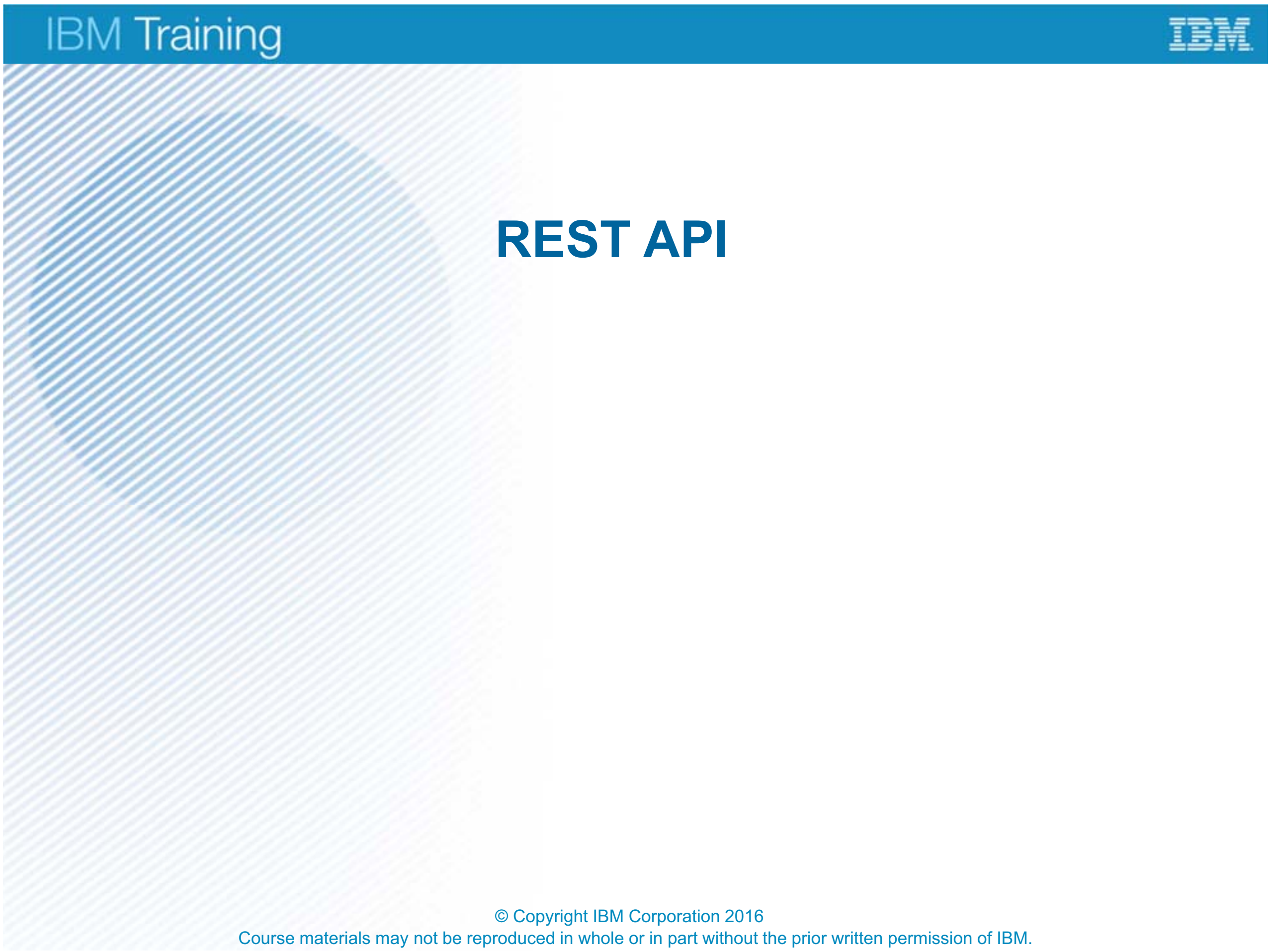


Figure 6-20. REST API

IBM Training

Using REST API

- Use a browser to call REST API methods
- Verify connections to servers
  `https://[hostname]:[port]/IBMJMXConnectorREST`
- Verify deployment of solutions to servers
  `https://[hostname]:[port]/IBMJMXConnectorREST`
- Get values for testing
  - Check whether an entity is created
  - Verify the current value of a global aggregate

Testing solutions

Figure 6-21. Using REST API

You can use the REST API as a quick way to verify connections to servers, deployment of solutions, or get values, such as entity instances or aggregate values.

IBM Training

Using the REST API to access deployed solutions

- List all deployed solutions:
  `http://[hostname]:[port]/ibm/ia/rest/solutions`

- List all entity types that are managed by your solution:
  `http://[hostname]:[port]/ibm/ia/rest/solutions/MySolution/entity-types`

- List all aggregates in your solution:
  `http://[hostname]:[port]/ibm/ia/rest/solutions/MySolution/aggregate`

- Retrieve a global aggregate value:
  `http://[hostname]:[port]/ibm/ia/rest/solutions/MySolution/aggregate/aggregate_name`

Testing solutions

Figure 6-22. Using the REST API to access deployed solutions

On this slide, you see URL examples that you can use to validate that your solution is deployed and to see which entities are created. You can also check for the aggregates in your solution and their values.

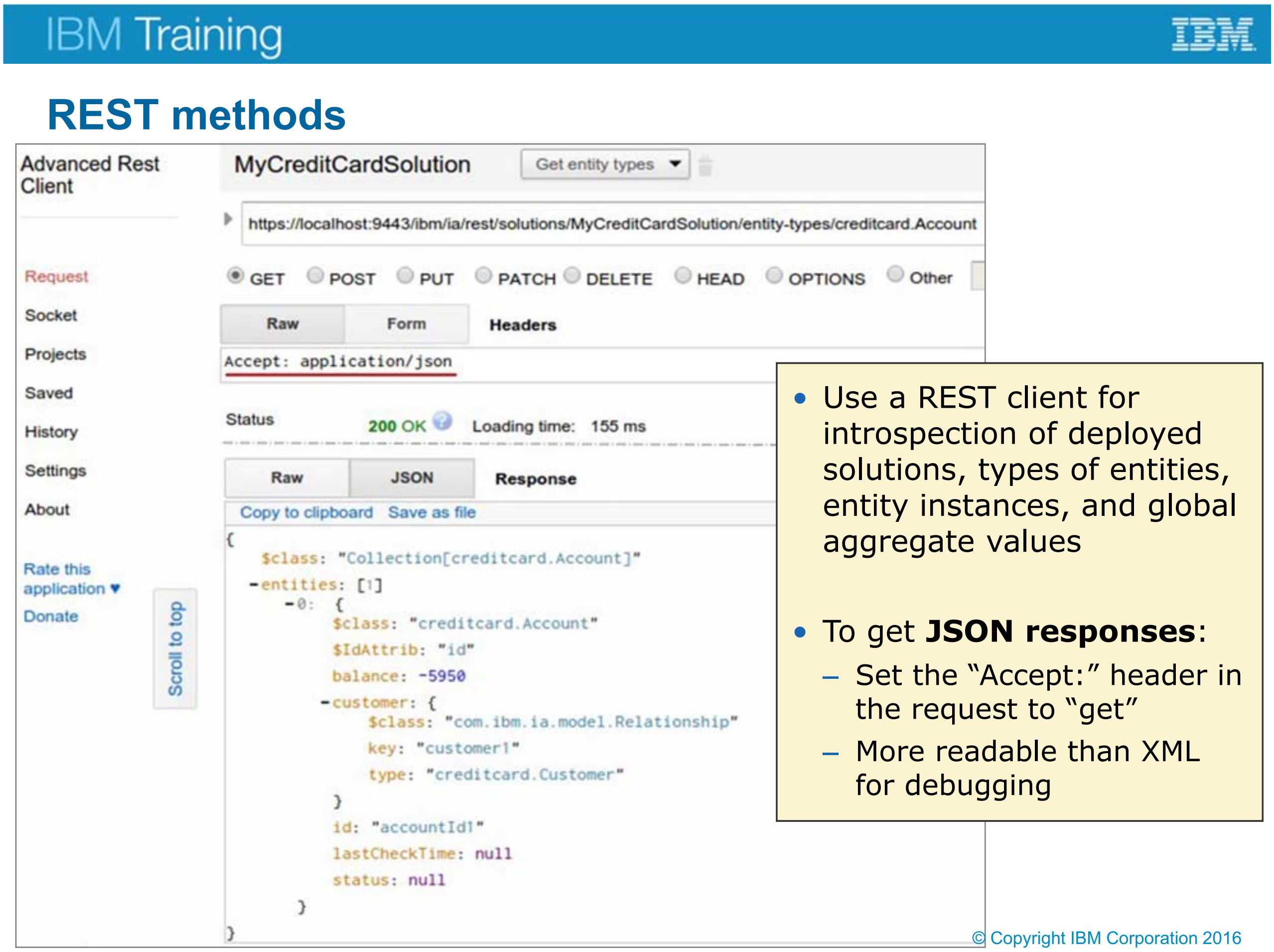


Figure 6-23. REST methods

After you run a test project, you can use the REST API as a quick way to check that entities are inserted into the grid.

You can use a REST client to view deployed solutions, types of entities, entity instances, and global aggregate values. If you want JSON responses, you need to set the `accept-header` in the request to get. JSON is more readable than using XML for debugging.

6.4. Insight Inspector

Figure 6-24. Insight Inspector

IBM Training

Overview

- Web application to review and analyze solutions during development
- Record a test run of a solution
- Review and inspect the events and entities that are associated with each solution agent
- Troubleshoot failures and errors
  - For example: No emitted events, rules are not fired, unexpected results
- Insight Inspector is a component of the Insights Runtime feature
  - For non-production

Testing solutions

Figure 6-25. Overview

Insight Inspector is a web-based debugging tool that helps you review and analyze events, agents, and entities that are associated with a solution test.

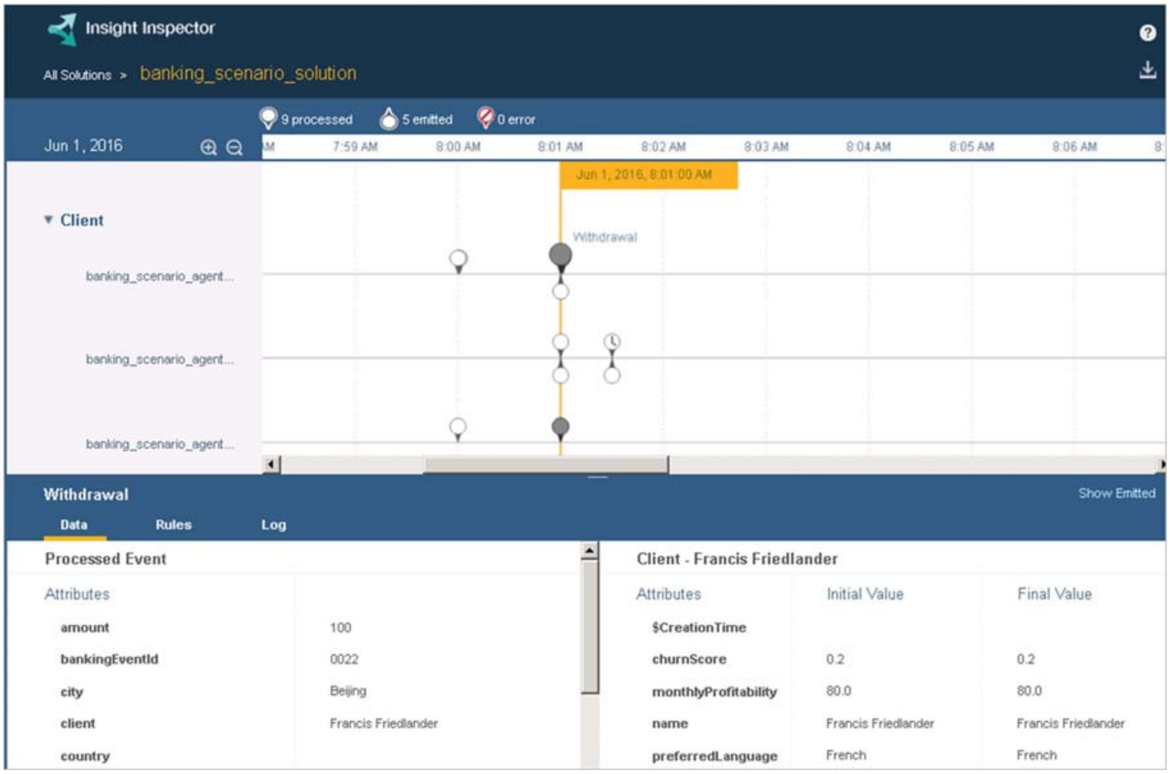
When you open Insight Inspector, you see a list of recorded tests that are identified by solution name and time.

When you open the test, you see the solution entities, agents, and event timelines that are organized into groups according to the entity type.

Insight Inspector is intended for testing during development; it is not a production tool.

IBM Training

Web interface



Testing solutions

Figure 6-26. Web interface

This slide shows an example of what you see when you open the solution to test in the inspector.

Record a test run to capture the event activity. After the test and after you stop the recording, you can open the recording in a browser.

Insight Inspector displays the event activity on a timeline. Events are represented as icons above and below the timeline.

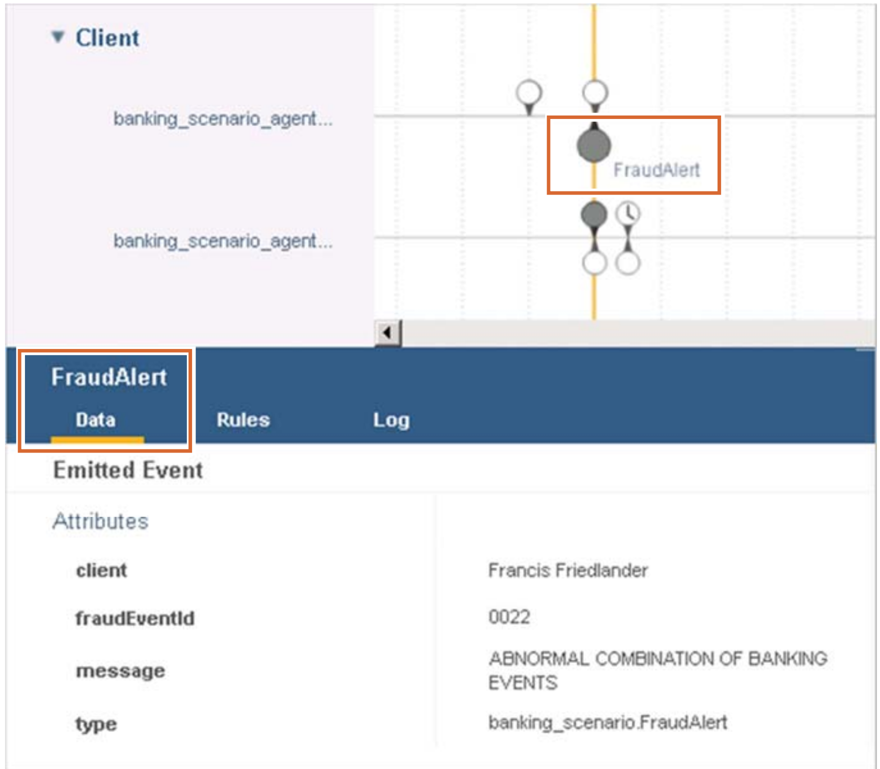
You can zoom and scroll to get a clearer view of the agents, entities, and events, and their details.

The lower pane includes three tabs: **Data**, **Rules**, and **Log**.

IBM Training

Checking event and entity data

- **Data** section
  - Lists entity and event attributes



Testing solutions

Figure 6-27. Checking event and entity data

The **Data** section displays information about the selected event and entities that are associated with the event. To review the event and entity data, click to select an event icon in the timeline.

When an event is processed, the agent might emit zero, one or multiple related events.

IBM Training

Verifying rules as fired

- **Rules** section
  - Displays which rules fired and did not fire

Testing solutions

Figure 6-28. Verifying rules as fired

The **Rules** section displays a list of rules that are fired as a result of processing the selected event. The rules are processed and evaluated during the recording. Use the **Fired** and **Not Fired** lists to verify that the correct rules are fired during event processing.

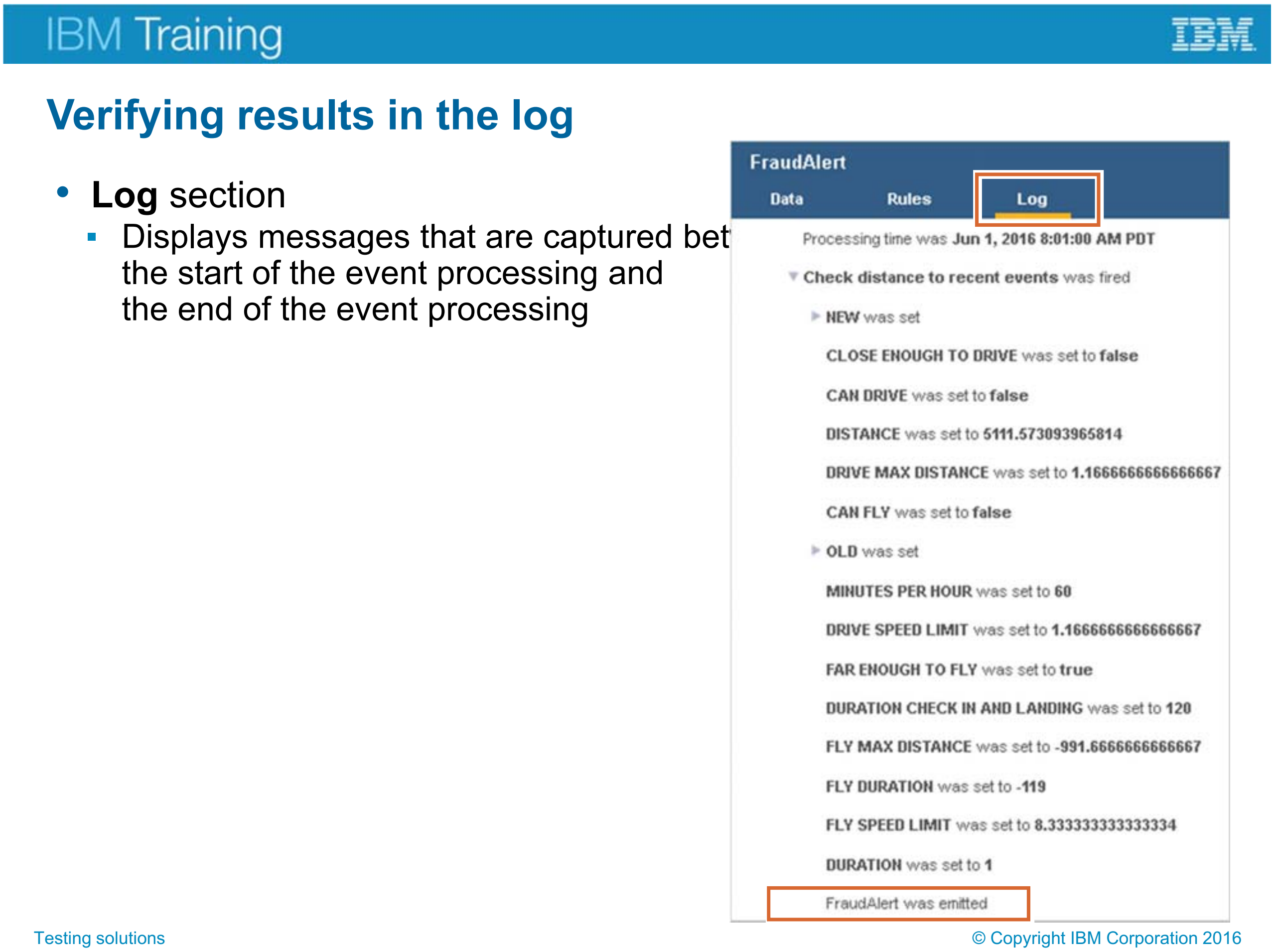


Figure 6-29. Verifying results in the log

Use the **Log** section to check the log information for a selected event.

The **Log** section displays messages that are captured between the start of the event processing and the end of the event processing.

The original server log files are stored in the `<InstallDir>/runtime/wlp/usr/servers/<server_name>/logs` directory.

IBM Training

Managing recording

- Start the recording by using `TestDriver.startRecording()` or the REST API:
  `http://[hostname]:[port]/ibm/insights/rest/recording/start/MySolution`
- Run your solution test client to load entities and submit events for processing
- Stop the recording by using `TestDriver.stopRecording()` or the REST API:
  `http://[hostname]:[port]/ibm/insights/rest/recording/stop/MySolution`
- View details at Insight Inspector URL:
  `http://[hostname]:[port]/ibm/insights`
  - For example: `http://localhost:9080/ibm/insights`

Testing solutions

Figure 6-30. Managing recording

To start and stop the recording, you can use the Test driver API or you can use the REST API.

After you start the recording, then you create the entities and submit events to the runtime. When the runtime processing is done, you stop the recording and open Insight Inspector in a browser. Your recording becomes available for you to select and review.

6.5. Insight Map Viewer

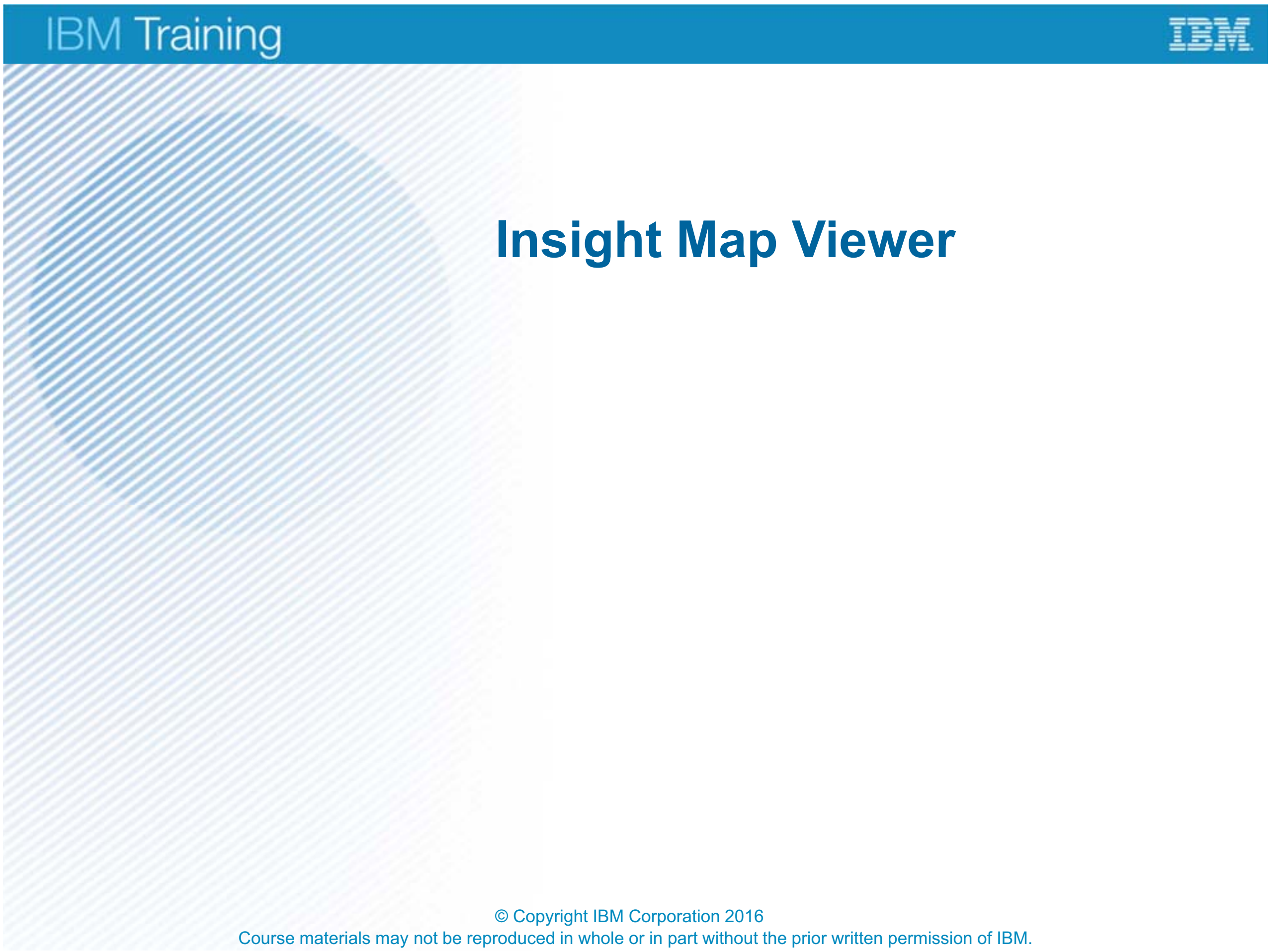


Figure 6-31. Insight Map Viewer

IBM Training

Insight Map viewer

- Visualize entities that have geospatial attributes
- View real-time changes to geographic locations of entities
- Highlight the entities on the map with different colors and symbols

Testing solutions

Figure 6-32. Insight Map viewer

You can use Insight Map Viewer to view real-time changes to entities that include attributes that are related to geographic locations.

If your solution involves entities that have geospatial attributes, you can use Insight Map Viewer to view the entities and how they change in response to events. The visualizations complement the detailed data that you obtain from Insight Inspector.

For example, if you have a solution for a car rental agency, you can create entities with moving geometry attributes to represent the vehicles that are rented. Entities with static geometry attributes can represent business partner locations. By using a map to view the moving vehicles and the partner locations, you can quickly check whether the solution is working as you expect it to.

You can also highlight the entities on the map with different colors and symbols to indicate their status based on attributes that are in your model. For example, you can indicate the type of vehicle or whether a location is open or closed.

IBM Training

Configuring the Map Viewer

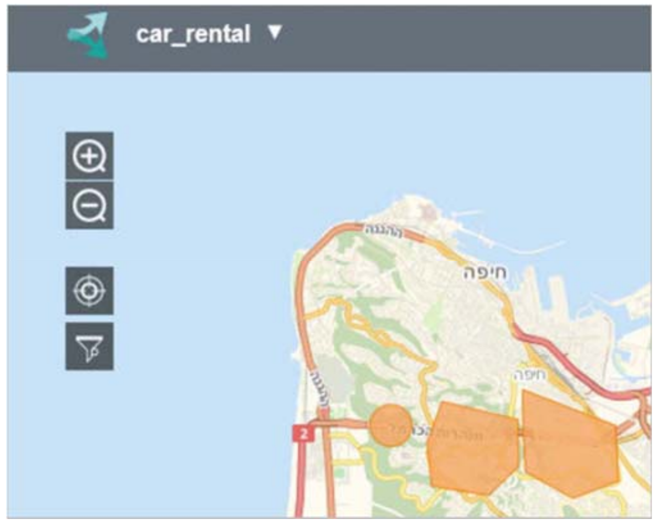
- Configure your development server by adding a <feature> element to the `server.xml` file

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

- Open Insight Map Viewer in a browser at this URL:

  http://*hostname*:*port*/ibm/maps/application.html

- Choose the solution that you want to view
- Upload the BOM from your solution workspace
  - In the **bom** folder, select the `.bom` file
  - Every entity in the `.bom` file is displayed as a separate layer on the map



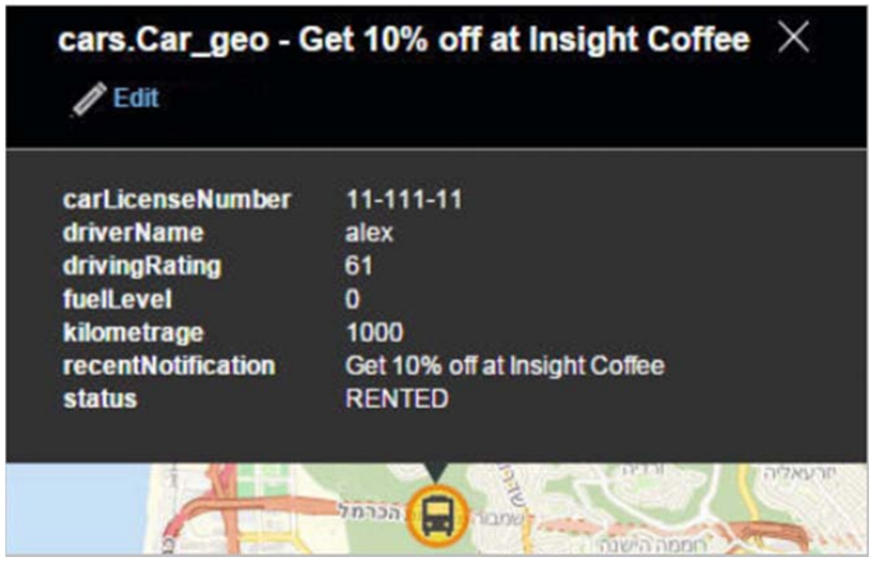
Testing solutions

Figure 6-33. Configuring the Map Viewer

IBM Training

Customizing the display

- Customize the color, size, and other visual characteristics of the entities
- When you click an entity, the entity's name and first-level attributes are displayed
- Edit the entity type by choosing:
  - Shape
  - Color
  - Size



Testing solutions

Figure 6-34. Customizing the display

IBM Training

Managing Insight Map Viewer settings

- If you modify the BOM in your solution, you can upload it again to Insight Map Viewer
  - After solution is redeployed, updates are displayed on the map
- Export your map customizations or import an existing configuration
  - Settings are saved as a JSON file
- Choose the underlying map layer
  - Default is Open Street Map
- Use **Clean up** menu to delete map configuration

Testing solutions

Figure 6-35. Managing Insight Map Viewer settings

You can upload a different BOM, import or export your map customizations, choose a different map style, show or hide entity overlays, and revert to the default settings.

IBM Training

Unit summary

- Test solutions with the TestDriver API
- Test solutions with the Test Client
- Work with the REST API
- Analyze event processing with Insight Inspector
- Work with the Map Viewer

Figure 6-36. Unit summary

IBM Training

Checkpoint questions

1. True or false: The TestDriver object is provided with **Decision Server Insights** and includes methods that are required to test a solution.

2. Which of the following options is *not* a troubleshooting tool for an Insights solution?
   a. Server trace logs
   b. Debug mode in Insight Designer
   c. Insight Inspector

3. True or false: To capture event activity in Insight Inspector, you must first record the activity by using REST API.

Testing solutions

Figure 6-37. Checkpoint questions

Write your answers here:

1.

2.

3.

IBM Training

Checkpoint answers

1. True or false: The TestDriver object is provided with **Decision Server Insights** and includes methods that are required to test a solution.
   Answer: True

2. Which of the following options is *not* a troubleshooting tool for an Insights solution?
   a. Server trace logs
   b. Debug mode in Insight Designer
   c. Insight Inspector
   Answer: b. Debug mode in Insight Designer

3. True or false: To capture event activity in Insight Inspector, you must first record the activity by using REST API.
   Answer: True

Testing solutions

Figure 6-38. Checkpoint answers

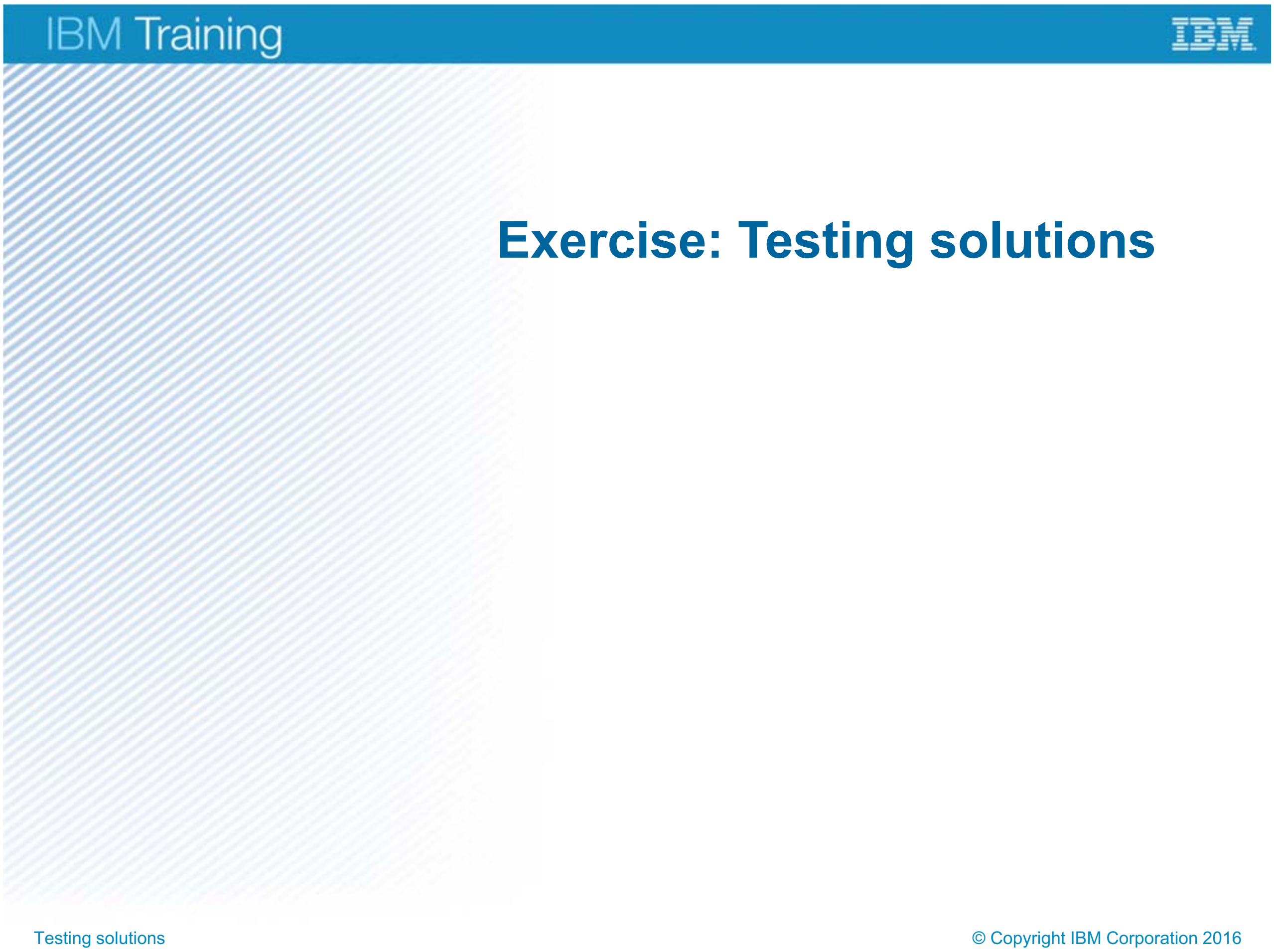


Figure 6-39. Exercise 10

IBM Training

Exercise objectives

- Create a test client project
- Run a test scenario

Testing solutions

Figure 6-40. Exercise objectives

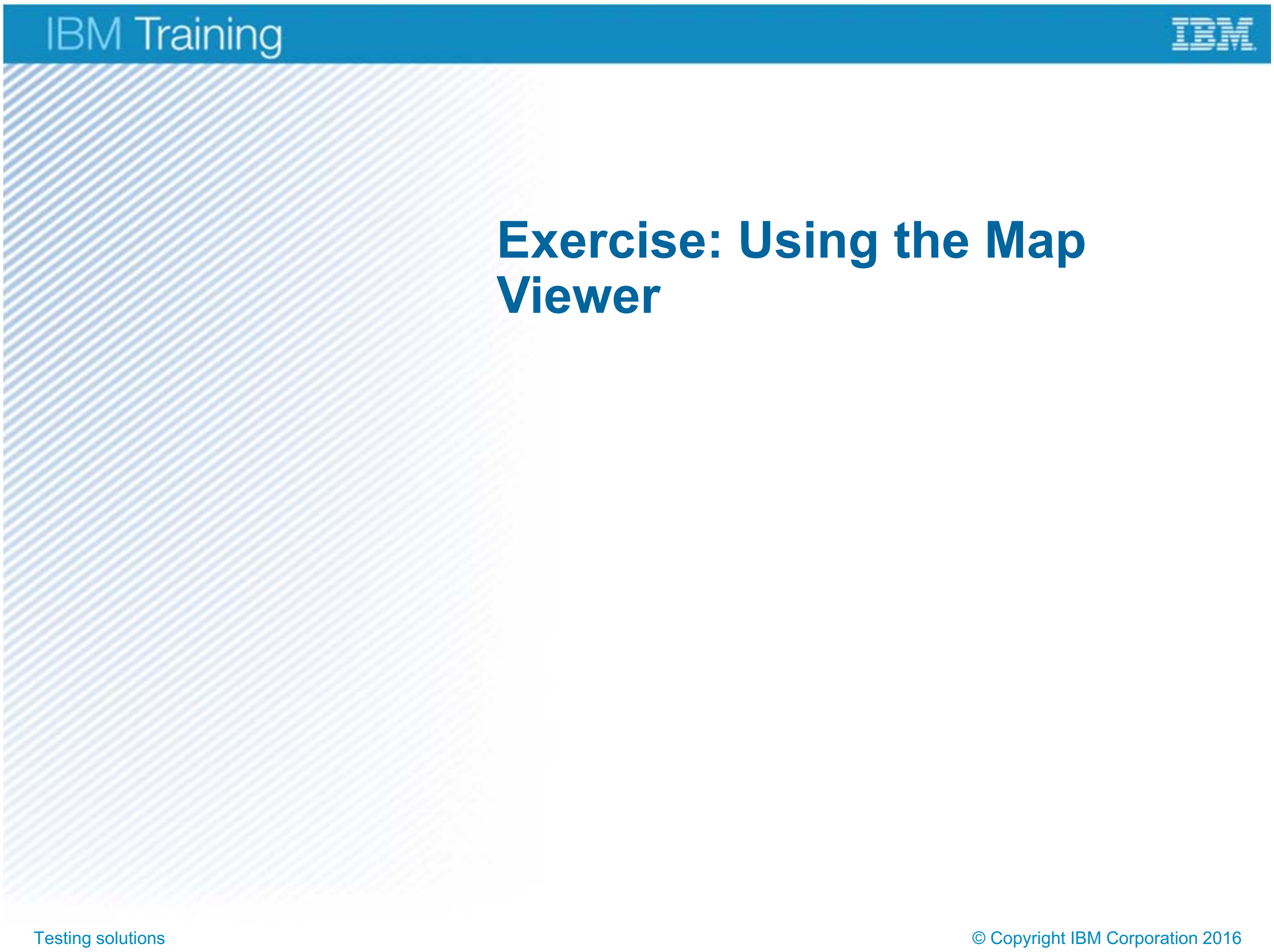


Figure 6-41. Exercise 11

IBM Training

Exercise objectives

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

Testing solutions

Figure 6-42. Exercise objectives

Unit 7. Modeling and defining connectivity

Estimated time

01:00

Overview

This unit describes how to define and manage connectivity for your solution.

How you will check your progress

- Checkpoint
- Exercise

IBM Training

Unit objectives

- Define inbound and outbound connectivity for a solution
- Configure and deploy connectivity

Modeling and defining connectivity

Figure 7-1. Unit objectives

IBM Training

Topics

- Event delivery
- Modeling connectivity
- Defining connectivity for your solution
- Deploying connectivity
- Troubleshooting connectivity issues

Modeling and defining connectivity

Figure 7-2. Topics

7.1. Event delivery

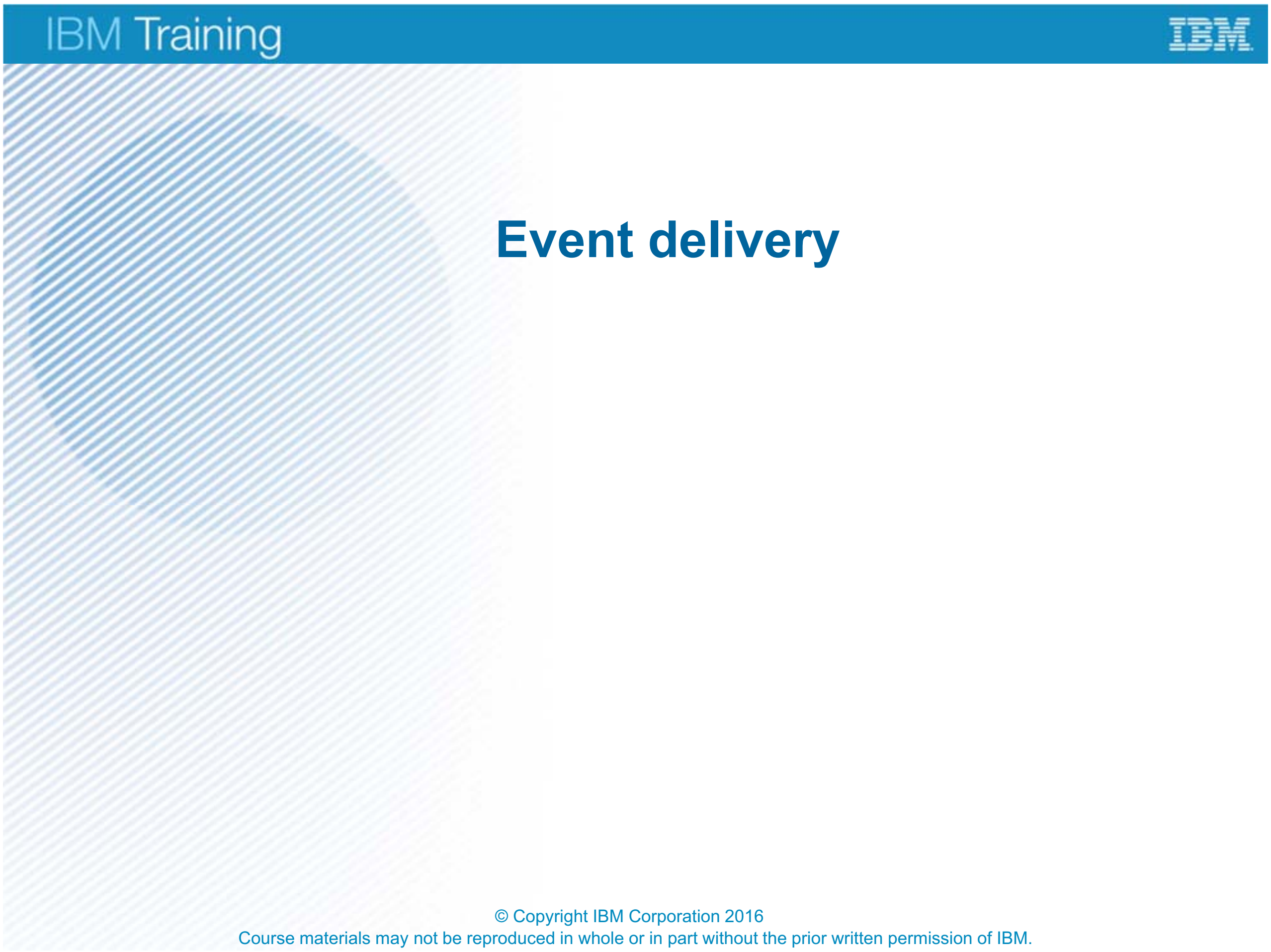


Figure 7-3. Event delivery

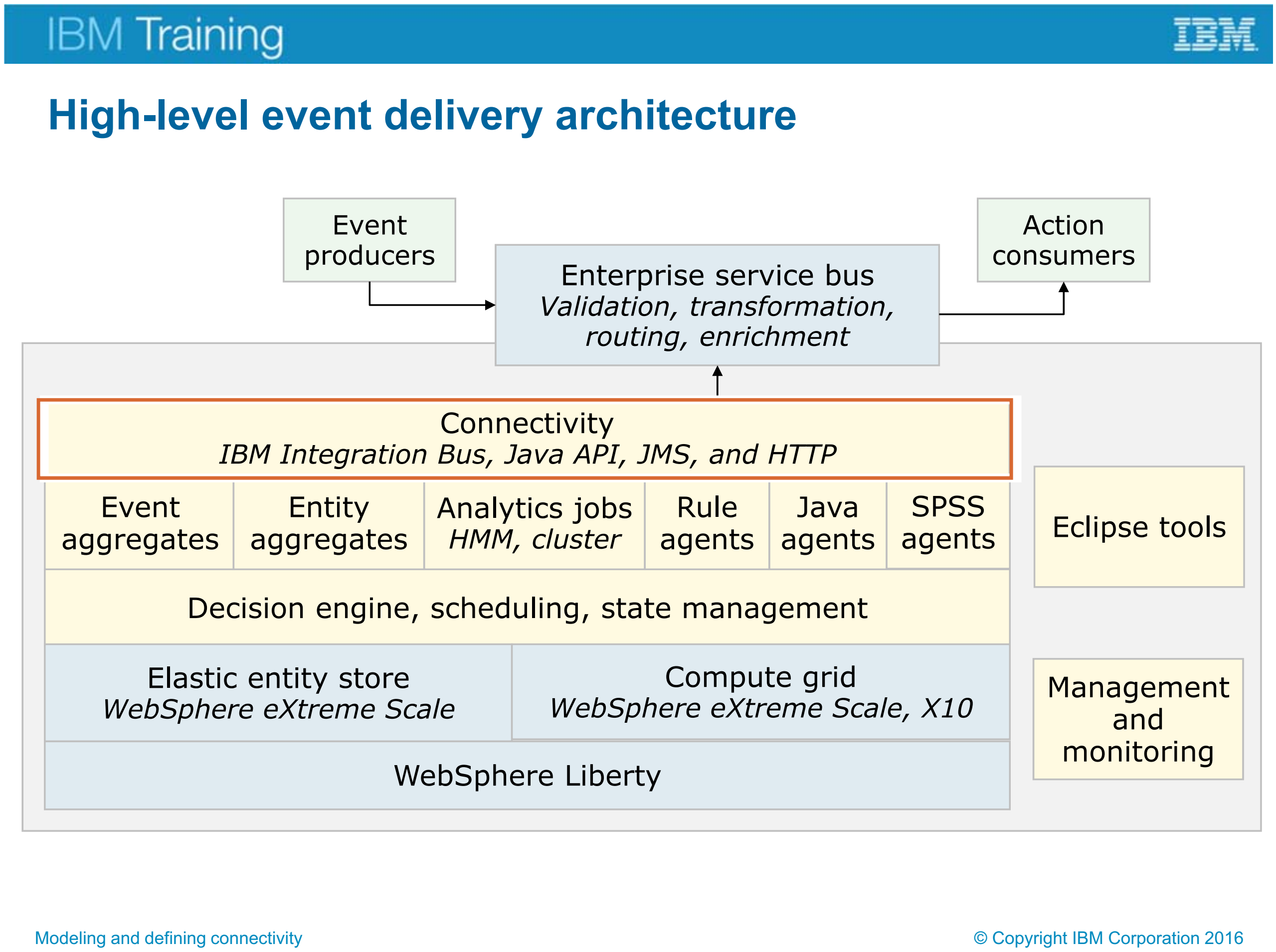


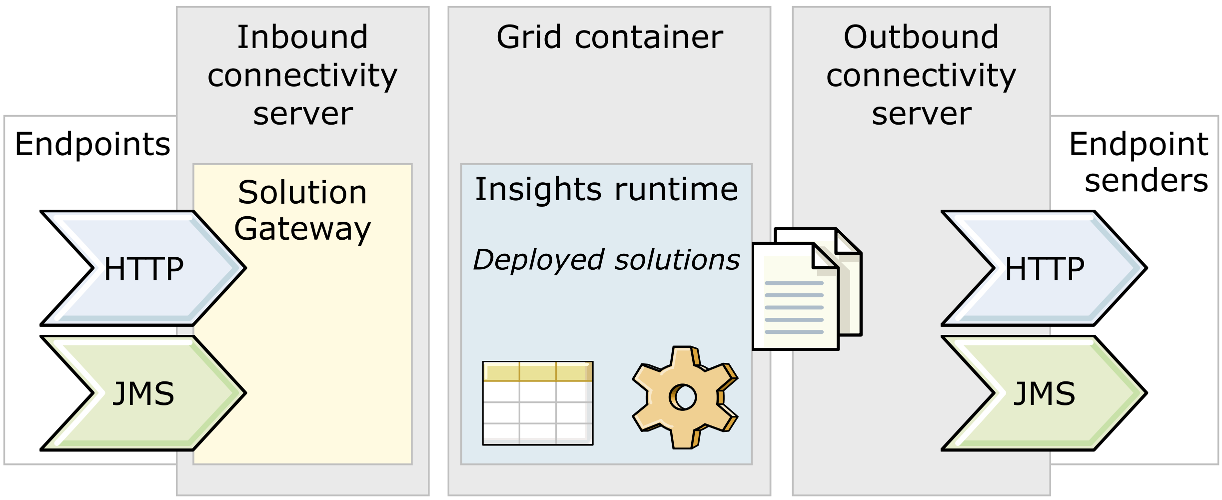
Figure 7-4. High-level event delivery architecture

Recall that the Decision Server Insights architecture includes a connectivity tier on top of the runtime tier. Decision Server Insights interacts with the outside world through the connectivity layer.

Inbound connectivity acts as a bridge from external event producers to the runtime. Outbound connectivity is the mechanism by which events are delivered from a solution to the event consumers or action consumers.

IBM Training

Connectivity architecture



Modeling and defining connectivity

Figure 7-5. Connectivity architecture

Here, you see the structure for connectivity components.

The diagram on this slide shows an inbound server and an outbound server. You can have separate servers or you can combine them, like the sample server in Decision Server Insights (cisDev). During the labs, you used a test Java client to submit events and get the results. With the test client, no connectivity was required.

To get the events in and out of the Insights runtime, two transport types are supported: HTTP and JMS. Event producers in the outside world pass HTTP or JMS messages to Decision Server Insights. Inbound messages are received by the Solution Gateway and routed to the runtime.

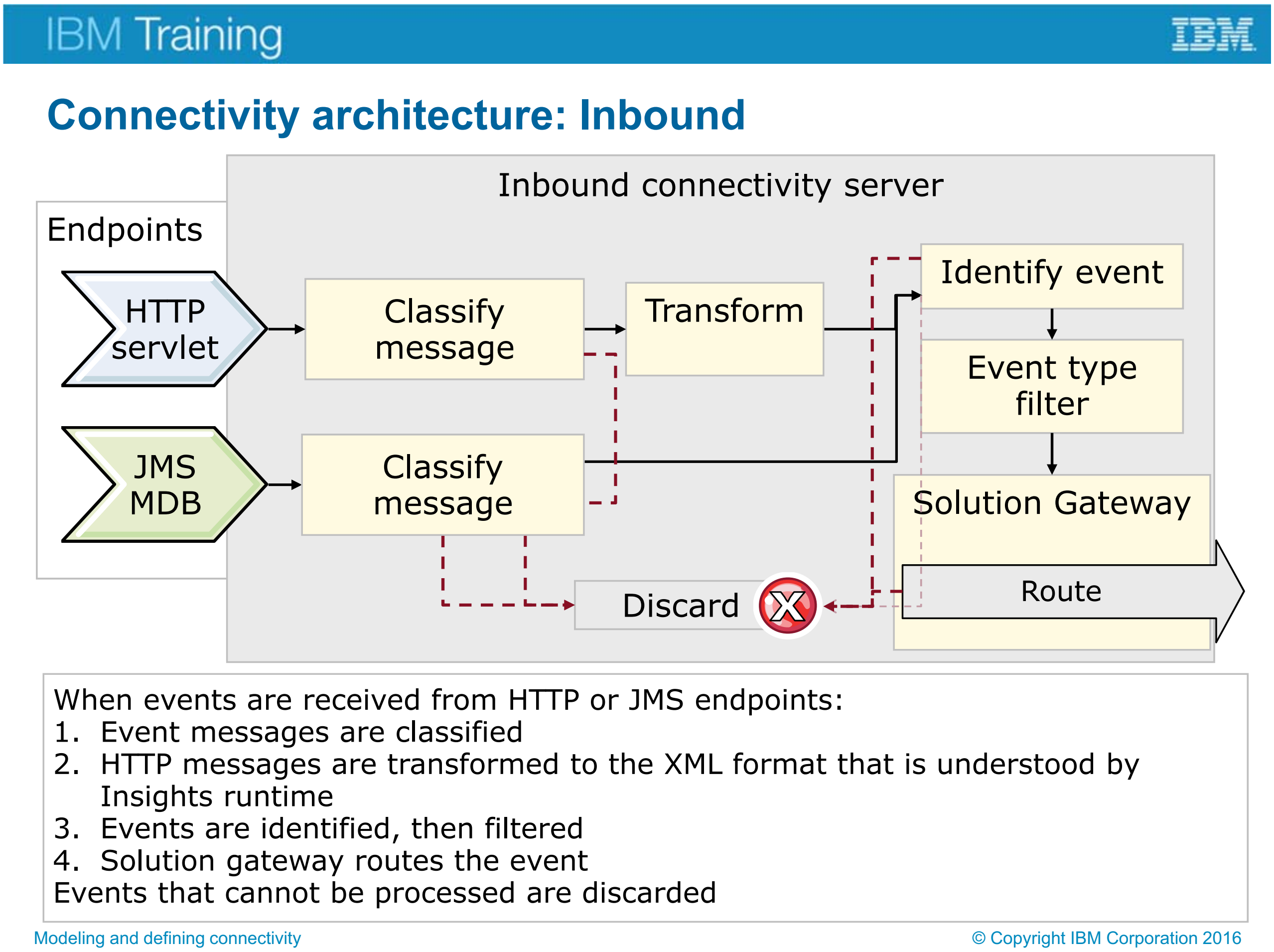


Figure 7-6. Connectivity architecture: Inbound

Inbound connectivity acts as a bridge from external messaging endpoints to the solution gateway.

Decision Server Insights uses inbound bindings for inbound JMS messages or XML messages over HTTP.

The inbound binding identifies the format and protocol of inbound messages, and must reference an inbound endpoint.

If an inbound message is not recognized by Decision Server Insights, the inbound binding can transform the message with an XSL transformation. Decision Server Insights classifies inbound messages by testing them against an XPath expression. If the message matches the XPath expression, the message is transformed into an event. If the message does not match the XPath expression, the message is matched against subsequent classifiers. If the message does not match any of the classifiers, a specified alternative action occurs, which can be either to discard the message, or to submit it to the runtime environment without transformation.

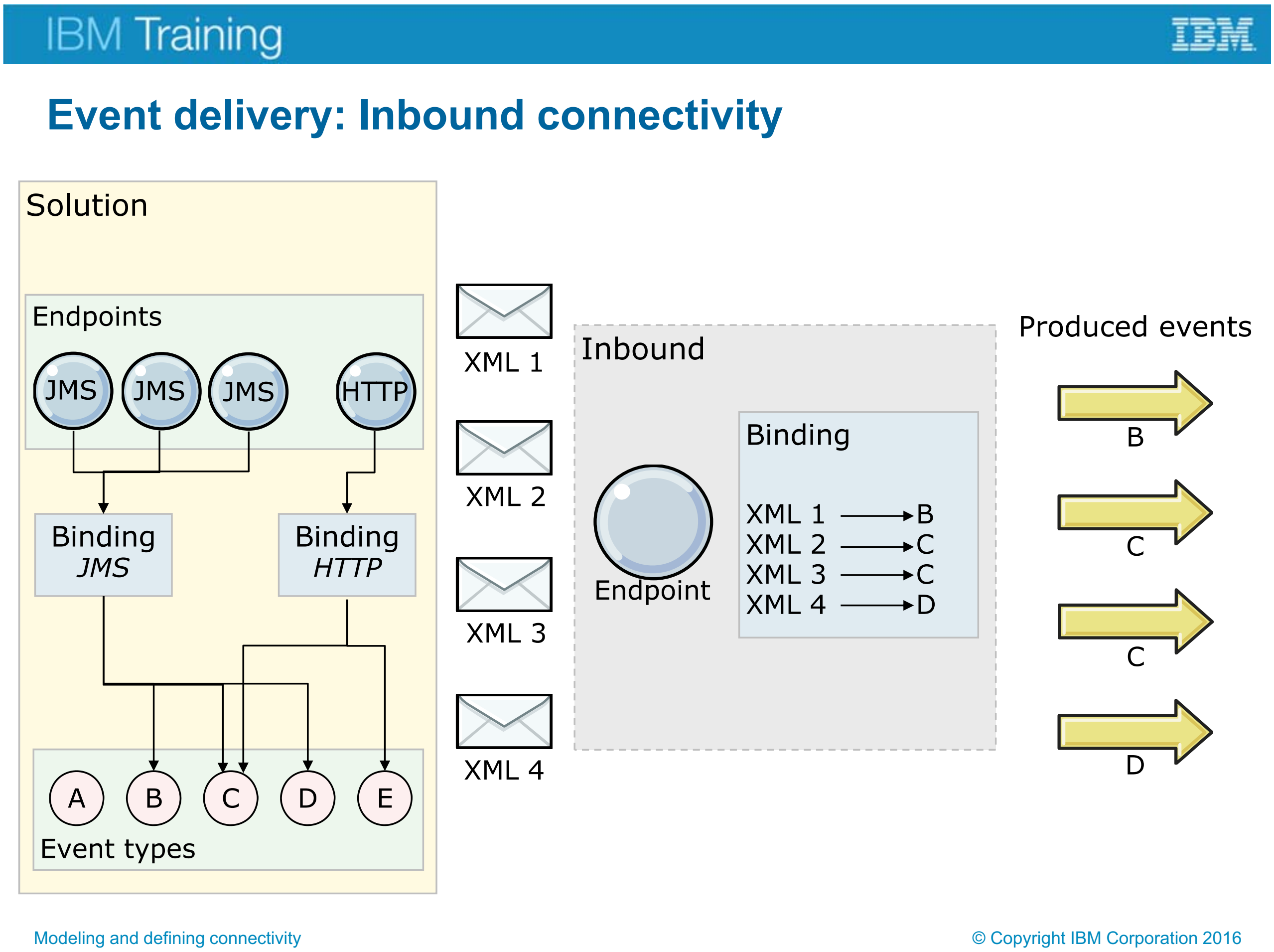


Figure 7-7. Event delivery: Inbound connectivity

Bindings and endpoints define how the solution receives and sends events. Binding describes the type of transport to be used, either JMS or HTTP, the events that are processed, and how the events are represented in the message.

You associate an event with a binding. You can have multiple bindings with different events, or you can have a single binding with all the events, or you can have a combination.

You also associate endpoints with a binding. The endpoint represents the origin of the messages, either the JMS destination or the HTTP URLs.

This diagram shows four inbound endpoints and the bindings that they reference. You also see the inbound XML messages, which arrive over HTTP or JMS, and the events that those messages produce.

IBM Training

Solution gateway for inbound connectivity

- The solution gateway is the entry point for events from external sources
- Access to the solution gateway
  - Directly through a Java API
  - Indirectly by modeling and deploying inbound connectivity as part of your solution
- The solution gateway establishes a connection between a solution and an object grid
  - Connections can be established between multiple solutions and a single object grid
- A gateway instance is used to submit events for processing, which involves placing them on an event queue

Modeling and defining connectivity

Figure 7-8. Solution gateway for inbound connectivity

The solution gateway API is the mechanism through which inbound events are submitted to the Decision Server Insights runtime environment. The solution gateway is not used for delivering outbound event messages.

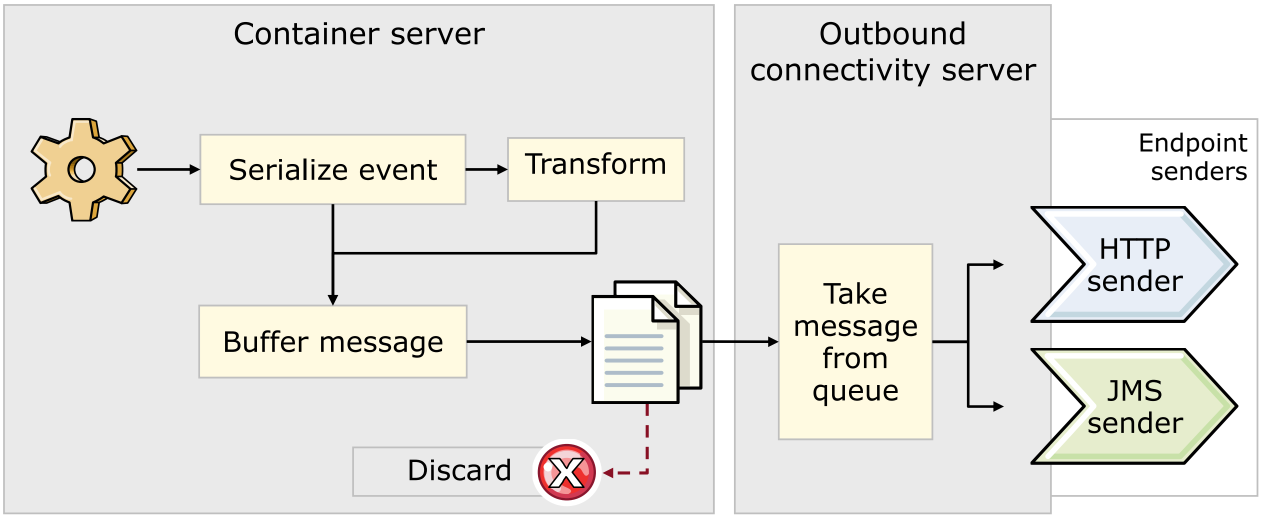
You can access the solution gateway either directly through a Java API, or indirectly by modeling and deploying inbound connectivity as part of your solution.

Each solution gateway instance represents a single, specified solution that is connected to the object grid. The solution gateway establishes a connection between a solution and an object grid. Connections can be established between multiple solutions and a single object grid.

A gateway instance is used to submit events for processing, which involves placing them on an event queue.

IBM Training

Connectivity architecture: Outbound



After the runtime processes events, and the result generates an outbound event:

1. Outbound events are serialized
2. Events with HTTP binding are transformed
3. Outbound event messages are buffered
4. Outbound connectivity server takes a buffered message from the queue and passes it to either the HTTP endpoint sender or the JMS endpoint sender
5. Events that cannot be delivered are discarded

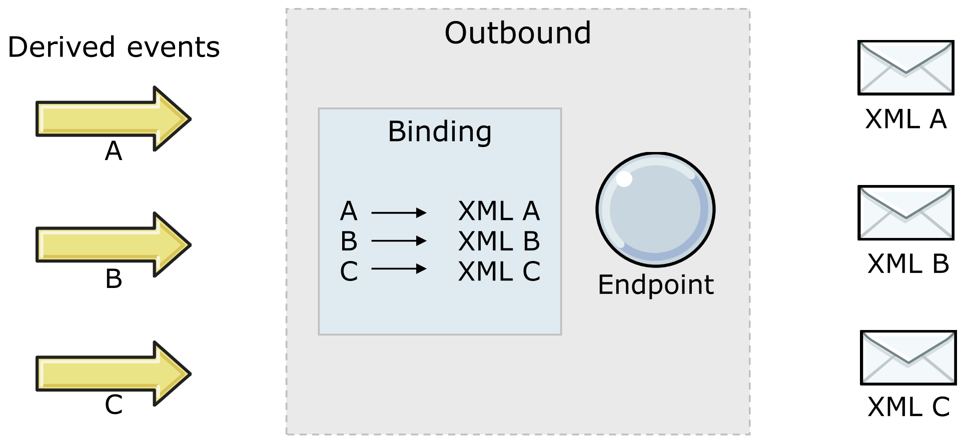
Modeling and defining connectivity

Figure 7-9. Connectivity architecture: Outbound

Outbound connectivity is the mechanism by which events can be delivered from a solution to the outside world. Decision Server Insights uses outbound bindings for sending outbound events in the form of serialized JMS or HTTP messages. The outbound binding determines which outbound events are sent, and determines the message format and protocol to be used. The outbound binding must reference an outbound endpoint that represents the destination for outbound JMS or HTTP messages. The destination is either a JMS connection factory and destination, or an HTTP URL.

IBM Training

Event delivery: Outbound connectivity



Modeling and defining connectivity

Figure 7-10. Event delivery: Outbound connectivity

Decision Server Insights does not have a programmatic equivalent of the solution gateway for outbound events.

Decision Server Insights uses outbound bindings for sending outbound events in the form of serialized JMS or HTTP messages. The outbound binding determines which outbound events are sent, and determines the message format and protocol to be used. The outbound binding must reference an outbound endpoint that represents the destination for outbound JMS or HTTP messages. The destination is either a JMS connection factory or a URL.

Here you see how Decision Server Insights serializes events into outbound messages that can be picked up and used by some external system.

7.2. Defining connectivity for your solution

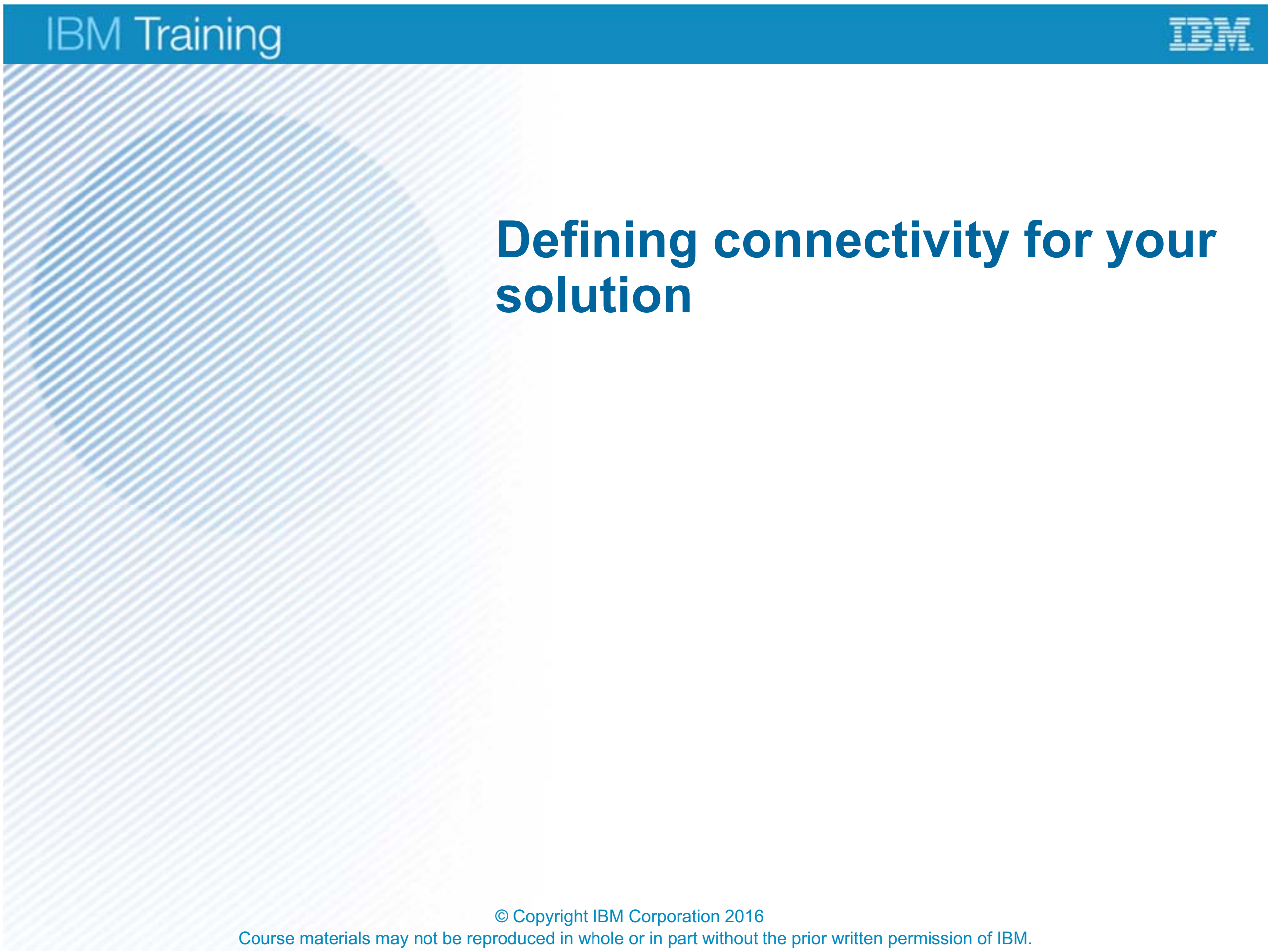


Figure 7-11. Defining connectivity for your solution

IBM Training

Defining connectivity in the solution project

- To define inbound or outbound connectivity, you create a `.cdef` file as part of the solution
  - The definition defines a binding with the type of transport to be used, either HTTP or JMS
  - The binding also defines events that will be processed
  - Endpoints are associated with a binding and define either the JMS destinations or HTTP URLs
- Steps:
  1. Define the connectivity for the solution
  2. Deploy the solution (which includes connectivity definitions)
  3. Deploy the connectivity configuration to the server

Modeling and defining connectivity

Figure 7-12. Defining connectivity in the solution project

When you create a solution in Insight Designer, you are not required to use connectivity to test your solution. The solution can be deployed to the Insight runtime and tested by using the test Java client that implements the solution gateway API. When you are ready to use inbound or outbound connectivity, you create a connectivity definition file (`.cdef`) as part of the solution.

This connectivity definition file defines the binding with the type of transport to use either HTTP or JMS, and it also defines which events to process. Endpoints are associated with a binding and define either the JMS destinations or HTTP URLs.

The connectivity definition file defines the solution connectivity as understood by the server, which is fixed; it cannot be changed. If you need to change it, it would need to be redeployed as a new version.

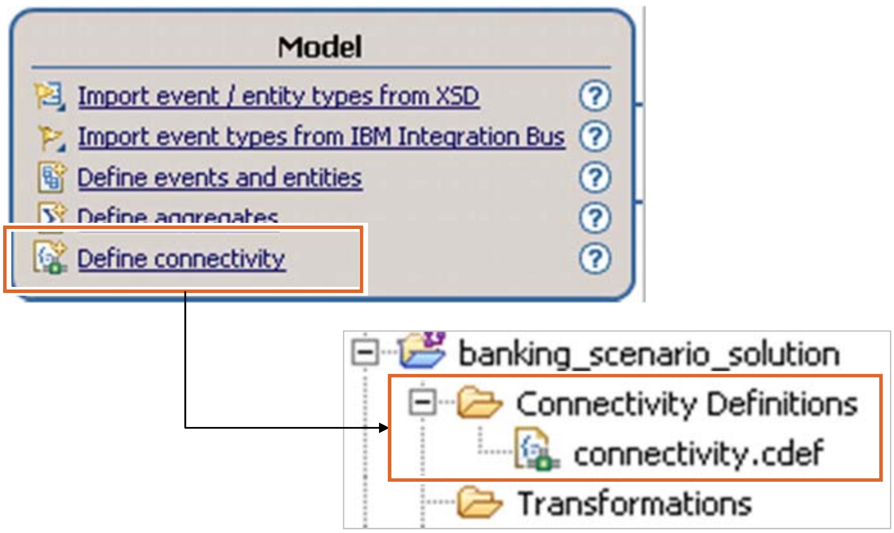
After you define connectivity for the solution, you deploy the solution to the runtime. You use the Export Solution Connectivity Server Configuration wizard to generate a connectivity configuration XML file.

To deploy the configuration, you use the `connectivityManager` command-line script. The script deploys the configuration file to the server. This configuration creates a Java Platform, Enterprise Edition application that runs on Liberty and contains the inbound endpoints. A configuration file is generated, and the `server.xml` file is updated to include this file. Client applications can then send and receive events to and from the Insights runtime.

IBM Training

Define connectivity in your solution

- To define connectivity for your solution, you can use the link in the Solution Map
- A wizard opens to create the `.cdef` file in the **Connectivity Definitions** folder of your solution



Modeling and defining connectivity

Figure 7-13. Define connectivity in your solution

To define connectivity, you can use the Solution Map link to open the wizard. The wizard automatically creates a `.cdef` file in the **Connectivity Definitions** folder of your solution.

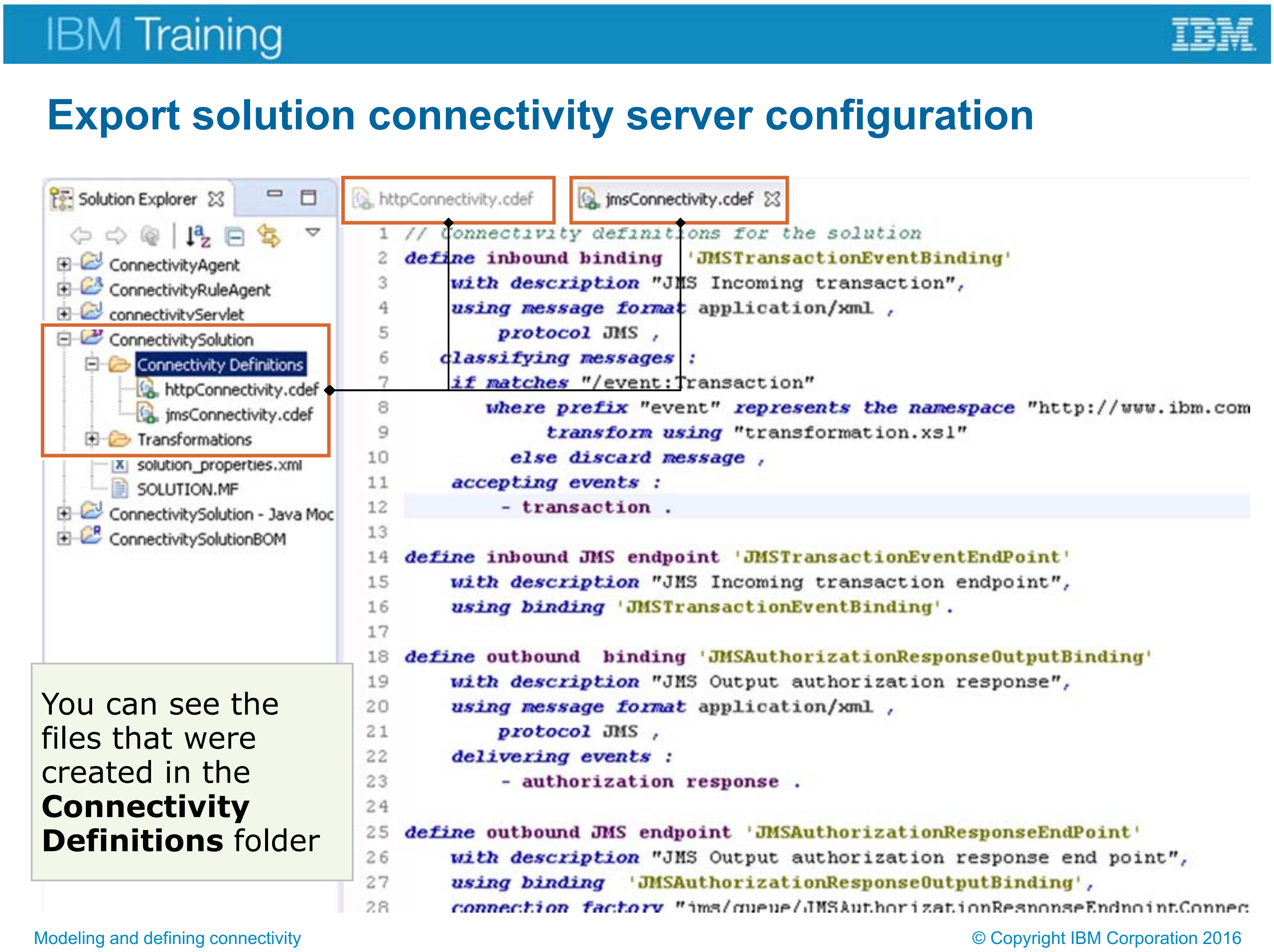


Figure 7-14. Export solution connectivity server configuration

The connectivity definitions are created as part of the solution. You can put all your connectivity definitions in a single file, or use separate files for inbound and outbound connectivity definitions. The definitions are stored in `.cdef` files.

In the connectivity definition file, you define:

- Binding: Interface for events and their representation
- Endpoints: Logical representation of the endpoint for the environment, either:
  - JMS endpoints represent a logical JMS queue or topic
  - HTTP endpoints represent an HTTP URL

Bindings and endpoints to define how a solution receives inbound events and sends outbound event messages. A binding describes the event types that are sent or received, and how they are represented in a message.

These definitions include the following information:

- Binding name
- Message format, either `application/xml` or `text/xml`
- Protocol, either JMS or HTTP

- Inbound messages to be received over this binding
- Endpoint name
- Endpoint binding name
- URL path (for HTTP only)

In the example that you see here, two files are created, one for HTTP and one for JMS binding.

In this example also, the inbound and outbound definitions are combined into one file. If you have separate inbound and outbound servers, you might want to have the different configuration files on the different servers.

IBM Training

Outbound binding and endpoint definitions

Inbound HTTP

```
define inbound binding 'HTTPTransactionEventBinding'
   with description "HTTP Incoming Transaction" ,
   using
      message format application/xml ,
   protocol HTTP ,
   classifying messages :
    if matches "/event:Transaction"
       where prefix "event" represents the namespace "http://www.ibm.com/
            transform using "transformation.xsl"
         else discard message ,
   accepting events :
      - transaction .

define inbound HTTP endpoint 'HTTPTransactionEventEndPoint'
    with description "HTTP Incoming transaction endpoint",
    using binding 'HTTPTransactionEventBinding',
   url path "/connectivity/Transaction" .
```

Modeling and defining connectivity

Figure 7-15. Outbound binding and endpoint definitions

To create the definition, the editor prompts you for each piece of information that is required. While you type, you can press Ctrl+Space to be prompted.

You need to specify:

- The binding name
- The message format, either `application/XML` or `text/XML`
- The protocol, either JMS or HTTP
- The inbound messages to be received over this binding
- The endpoint name
- The endpoint binding name
- The URL path for HTTP

If you define an HTTP inbound endpoint, you must provide the URL path on which this inbound endpoint receives messages. Make sure that the path contains at least two levels. The URL path is not required when you define a JMS inbound endpoint.

In this example, the inbound connectivity is bound to the “Transaction” event with the description: **`HTTP Incoming Transaction`**

The message format is `application/XML` and it uses the HTTP protocol. The URL for the inbound endpoint uses the path: `connectivity/Transaction`

IBM Training

Outbound binding and endpoint definitions

Outbound HTTP

```
define outbound binding 'HTTPTransactionOutput'
   with
      description "HTTP Transaction Output" ,
   using
      message format application/xml ,
   protocol HTTP ,
   delivering events :
      - authorization response .

define outbound HTTP endpoint 'HTTPTransactionOutputEndPoint'
   with
      description "HTTP Transaction Output endpoint" ,
   using
      binding 'HTTPTransactionOutput' ,
   url "http://localhost:9080/eventSender/response" .
```

Modeling and defining connectivity

Figure 7-16. Outbound binding and endpoint definitions

This slide shows an example of an outbound binding and endpoint from a connectivity definition file.

The outbound binding definition requires:

- the binding name
- the description
- the message format, in this case, “`application/xml`”
- the protocol
- the event being delivered

The endpoint definition includes:

- the endpoint name
- and description
- the referenced binding
- and the destination, in this case, the destination is the URL pointing to the local host on port 9080, with the path “`/eventSender/response`”

IBM Training

Providers for JMS

- The connectivity feature supports two JMS providers
  - WebSphere MQ
  - WebSphere Application Server
- Use the Connectivity Wizard to choose which type of provider to use
  - Wizard also prompts you for the required fields to complete the configuration
  - Note when using the WebSphere MQ Providers, the Connection Type can be only 'BINDINGS' if both Liberty and WebSphere MQ are running on the same machine
- When using WebSphere MQ, you need to set the path for the WebSphere MQ JMS Resource adapter
  - Example:

```
<variable name="wmqJmsClient.rar.location"
value="C:/wmq/wmq.jmsra.rar"/>
```

  - The `server.xml` file has commented-out sections that describe what needs to be enabled for the two JMS providers

Modeling and defining connectivity

Figure 7-17. Providers for JMS

Insight Designer has a Connectivity wizard to generate connectivity configurations for WebSphere Application Server or WebSphere MQ. Other configurations can be completed by using the XML editor or the WDT server configuration editor.

IBM Training

Transformations

- Use transformations to translate inbound and outbound XML messages
- Inbound XML translated into an XML message understood by Decision Server Insights
- Outbound event translated into XML file to be sent to the receiver

Modeling and defining connectivity

© Copyright IBM Corporation 2016

Figure 7-18. Transformations

Transformations allow you to send an arbitrary XML message to the channel that can be translated into an XML message that Decision Server Insights can understand. You use XSLT to do that. Similarly, you can transform an outbound event into a different XML file to be sent to the receiver.

You can create the transformation file from the **File > New Transformation** file menu in the Decision Insight perspective. You complete the details in the dialog box and select whether you want to transform inbound or outbound events (a different file is required for each), and it generates a template file for you.

IBM Training

Classifying inbound messages for transformation

- After the transformation is defined, the message must be classified so that a transformation can be selected
- The message is classified by using MessageContext, which you can use to make a choice of transformation based on XML or JMS headers
- Example:

```
define inbound binding 'binding1'
     using
         message format application/xml ,
         protocol JMS ,
     classifying messages :
         if matches "context:getJMSType() == 'PurchaseEvent'"
             transform using "transformPurchaseEvent.xsl"
         else discard message ,
     accepting events :
         - purchase event .
```

Modeling and defining connectivity

Figure 7-19. Classifying inbound messages for transformation

In order for transformations to take place, you need to classify the messages.

After the transformation is defined, the message must be classified as needing to be transformed. In the example here, you see that the binding includes the extra section: **`classifying messages`**

Transformations allow for more flexibility about how the XML gets into the system. The XML does not need to be in the correct format at the point it enters the system, which makes things easier from the client end.

7.3. Deploying connectivity

IBM Training

Deploying connectivity

Figure 7-20. Deploying connectivity

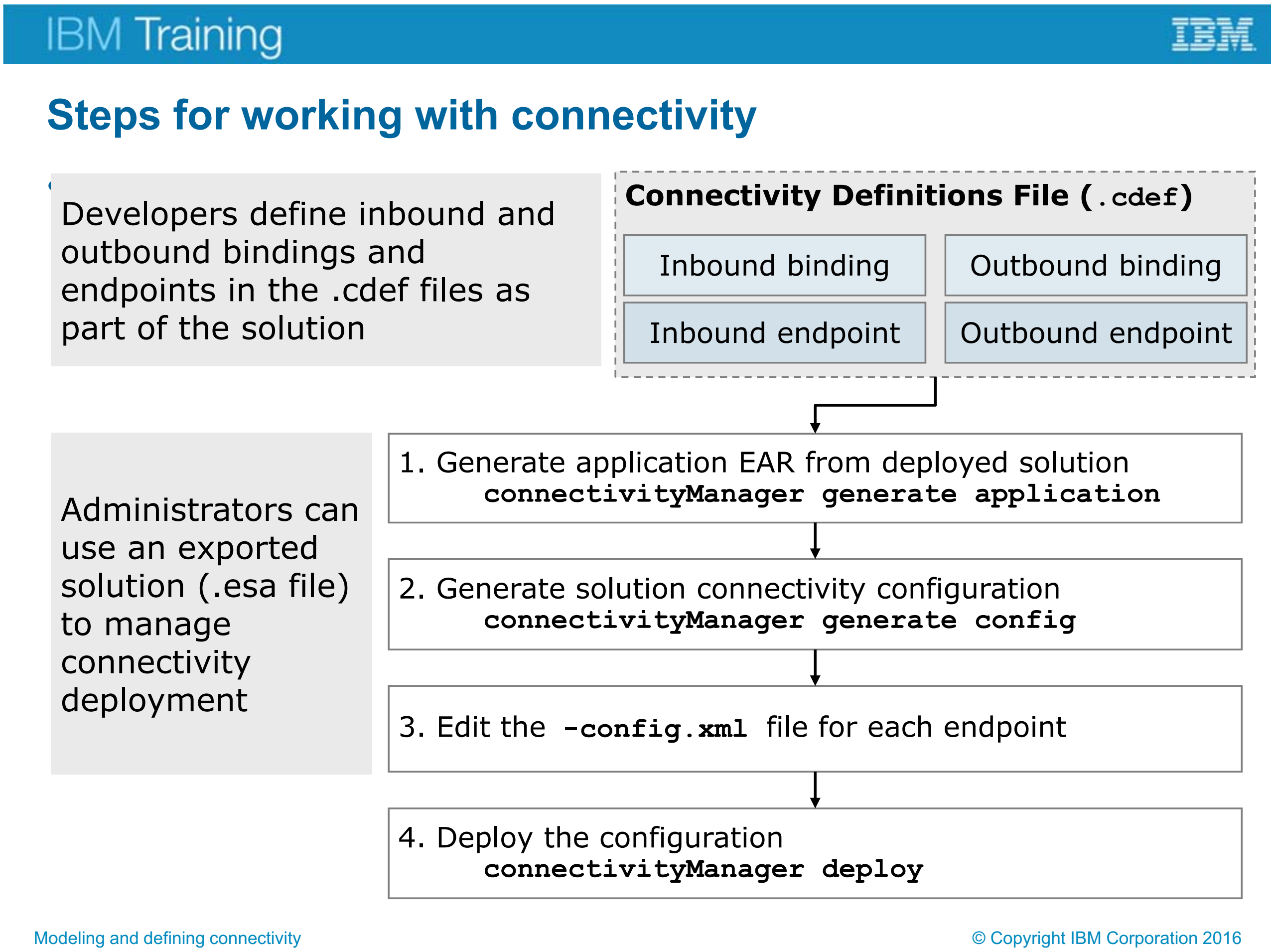


Figure 7-21. Steps for working with connectivity

Developers define inbound and outbound bindings and endpoints for the solution in `.cdef` files. Those inbound and outbound connectivity files must be deployed as XML files to the inbound and outbound servers.

Developers can export a solution that contains the connectivity definitions and pass the exported `.esa` file to administrators. Administrators can deploy the solution, generate the application EAR and connectivity configuration, edit the configuration endpoints and deploy the connectivity configuration to the servers.

IBM Training

Configuration and deployment

- Solution connectivity is configured as Liberty server configuration XML (`-config.xml`)
  - Identifies which endpoints the connectivity server should process
  - Maps the logical solution endpoints to the deployment environment
- Solution connectivity is deployed by using `connectivityManager`
  - Generates and deploys EAR for the requested inbound HTTP and JMS endpoints
  - Deploys solution connectivity configuration XML, which adds `include` to `server.xml`

Modeling and defining connectivity

Figure 7-22. Configuration and deployment

Solution connectivity is configured as a Liberty server configuration XML file. The connectivity configuration defines which endpoints the connectivity server should listen to. It also maps the logical solution endpoints to the deployment environment. For example, if you are using JMS inbound endpoints and HTTP for outbound endpoints, the configuration would include:

- JMS: Activation specifications for inbound JMS endpoints
- HTTP: Target URL, user ID, and password for outbound HTTP endpoints

When a solution has connectivity, inbound and outbound endpoints must be configured and deployed to the appropriate connectivity servers.

When you update a solution, you do not have to redeploy connectivity. You must redeploy connectivity only when the connectivity configuration for the solution changes.

IBM Training

Creating a solution connectivity configuration

- Use `connectivityManager generate` command to generate a skeleton solution connectivity configuration file
  - Generates placeholder configuration for the endpoints that are specified in the command prompt
  - Complete the configuration by using an XML editor or WDT server configuration editor
  - Example:
    ```
    connectivityManager generate config mysolution.esa
    mysolution-inbound-config.xml --inboundEndpoints="*"
    ```
- Use the `validate` command to validate your endpoint configuration
  - Example:
    ```
    connectivityManager validate mysolution.esa mysolution-inbound-
    config.xml
    ```

Modeling and defining connectivity

Figure 7-23. Creating a solution connectivity configuration

After you generate the connectivity configurations for your endpoints, you can use the `validate` command to validate the endpoint XML.

IBM Training

Editing the connectivity configuration

- Use a text editor to edit the configuration
  - Uncomment the sections that point to your application and define your endpoint
  - Map the security role

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

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

<!--Generated configuration for endpoint: connect1-->

    <ia_inboundHttpEndpoint endpoint="connectivity_solution/connect1" />
</server>
```

Map role to “admin”

Modeling and defining connectivity

Figure 7-24. Editing the connectivity configuration

The configuration points to the application EAR file in the `<application>` entry, which you uncomment. You also map the iaEventSubmitter role. In the example that is shown here, the iaEventSubmitter role is mapped to the “admin” user.

At the end of the file, the inbound endpoint is defined for an HTTP endpoint.

IBM Training

Deploying a connectivity configuration

- The connectivity configuration file must be deployed to the server or servers by using the `connectivityManager` script
  - Use this command to deploy the configuration XML file to the server
  - Example:
    `connectivityManager deploy local C:/Solutions/solutionFeature C:/Solutions/connectivity_config.xml`

- Results of running this action at the server level
  - A `<solution name>-inbound.ear` file is generated in the `apps` directory
  - A `<solution name>-config.xml` file is generated in the `root` directory of the server
  - The runtime server `server.xml` file the root directory is updated to include the `config` file

Modeling and defining connectivity

Figure 7-25. Deploying a connectivity configuration

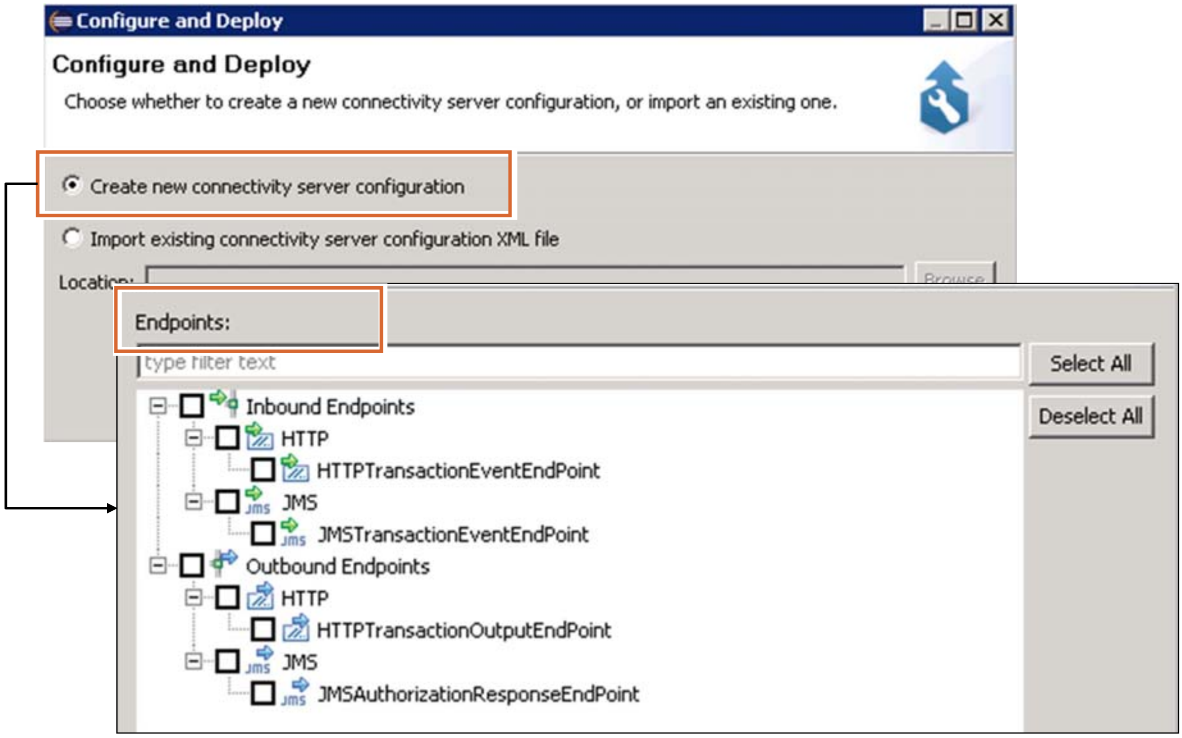
After you have your connectivity file, you use the **connectivityManager** command-line script to deploy it, which does the following tasks:

- It creates a Java Platform, Enterprise Edition application that runs on Liberty containing the inbound endpoints.
- A configuration file is generated with the connection details, and the **server.xml** file is modified to include this file.

After you restart your server, the connectivity is complete.

IBM Training

Deploying from Insight Designer



Modeling and defining connectivity

Figure 7-26. Deploying from Insight Designer

When you deploy the solution from Insight Designer by using the Solution Map, the deployment wizard generates the **config.xml** file and application EAR file in the correct directories of the runtime server.

The wizard first prompts you to create a connectivity server configuration. You can also choose to include all or some of the endpoints that you defined in the configuration. When you finish in the wizard, the wizard deploys the solution and connectivity to the runtime server and activates the connectivity for the solution.

7.4. Troubleshooting connectivity issues

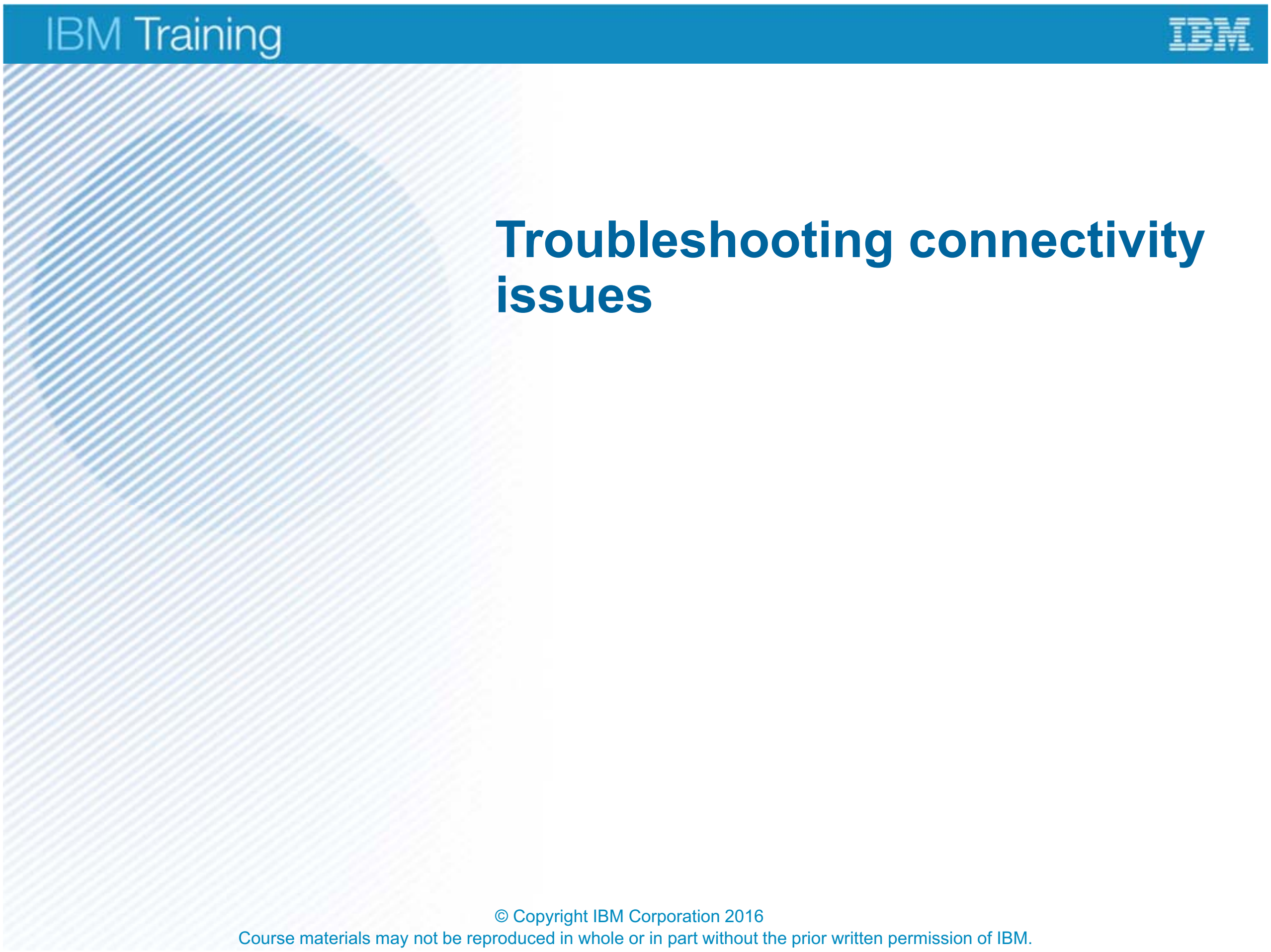


Figure 7-27. Troubleshooting connectivity issues

IBM Training

Troubleshooting (1 of 2)

- Trace
  - The following entry in the `server.xml` file turns on trace for connectivity

```
<logging maxFiles="10"
 traceSpecification="com.ibm.ia.connectivity*=fine:*=info"
 />
```

- Specific “must gather” items for support
  - `Messages.log`
  - `Trace.log`
  - Any FFDC
  - `server.xml` file, and any connectivity configuration files that `server.xml` includes
  - The `.cdef` files from the solution
  - The solution, if the customer is willing to share it

Modeling and defining connectivity

Figure 7-28. Troubleshooting (1 of 2)

Here, you see some troubleshooting tips, including how to start the trace and which log files are relevant.

IBM Training

Troubleshooting (2 of 2)

- Outbound Buffer Manager
  - Use the Outbound Buffer Manager to look at whether you have pending outbound events that cannot be sent for some reason
  - If required, you can also use the Outbound Buffer Manager to clear the events for an endpoint

Modeling and defining connectivity

Figure 7-29. Troubleshooting (2 of 2)

Here, you see some troubleshooting tips about the Outbound Buffer Manager.

IBM Training

Common issues and resources

If no messages are received at channel destinations:

- Check that the names in the configuration you are using are correct
- Check that the event that you submit to Insight Server on the inbound channel creates events that are to be sent through the outbound channels
- Use a test client to check that the events are being processed correctly by the server
  - Using a test client removes the channels from the path and determines whether events are being handled correctly

Modeling and defining connectivity

Figure 7-30. Common issues and resources

Some common issues that might cause events not to arrive might include incorrectly configuring the connection information, or the expected events are not generated by the Insights runtime because of the rule definitions.

To verify that the problem is in the connectivity, you can start by using a test client to send the event and get the results because the test client does not require connectivity. If the runtime handles the events properly, at least you know that your problem is with the connectivity.

IBM Training

Unit summary

- Define inbound and outbound connectivity for a solution
- Configure and deploy connectivity

Modeling and defining connectivity

Figure 7-31. Unit summary

IBM Training

Checkpoint questions

1. Which protocols are supported in connectivity definitions for the solution? Select all that apply:
   a. HTTP
   b. SOAP
   c. JMS
   d. JSON

2. True or false: A solution can have many inbound connectivity definitions.

3. True or False: You can use the `connectivityManager` utility to generate and deploy connectivity configuration files for your solution.

Modeling and defining connectivity

Figure 7-32. Checkpoint questions

Write your answers here:

1.

2.

3.

IBM Training

Checkpoint answers

1. Which protocols are supported in connectivity definitions for the solution? Select all that apply:
   a. HTTP
   b. SOAP
   c. JMS
   d. JSON

   Answer: a (HTTP) and c (JMS)

2. True or false: A solution can have many inbound connectivity definitions
   Answer: True

3. True or False: You can use the `connectivityManager` utility to generate and deploy connectivity configuration files for your solution.
   Answer: True.

Modeling and defining connectivity

Figure 7-33. Checkpoint answers

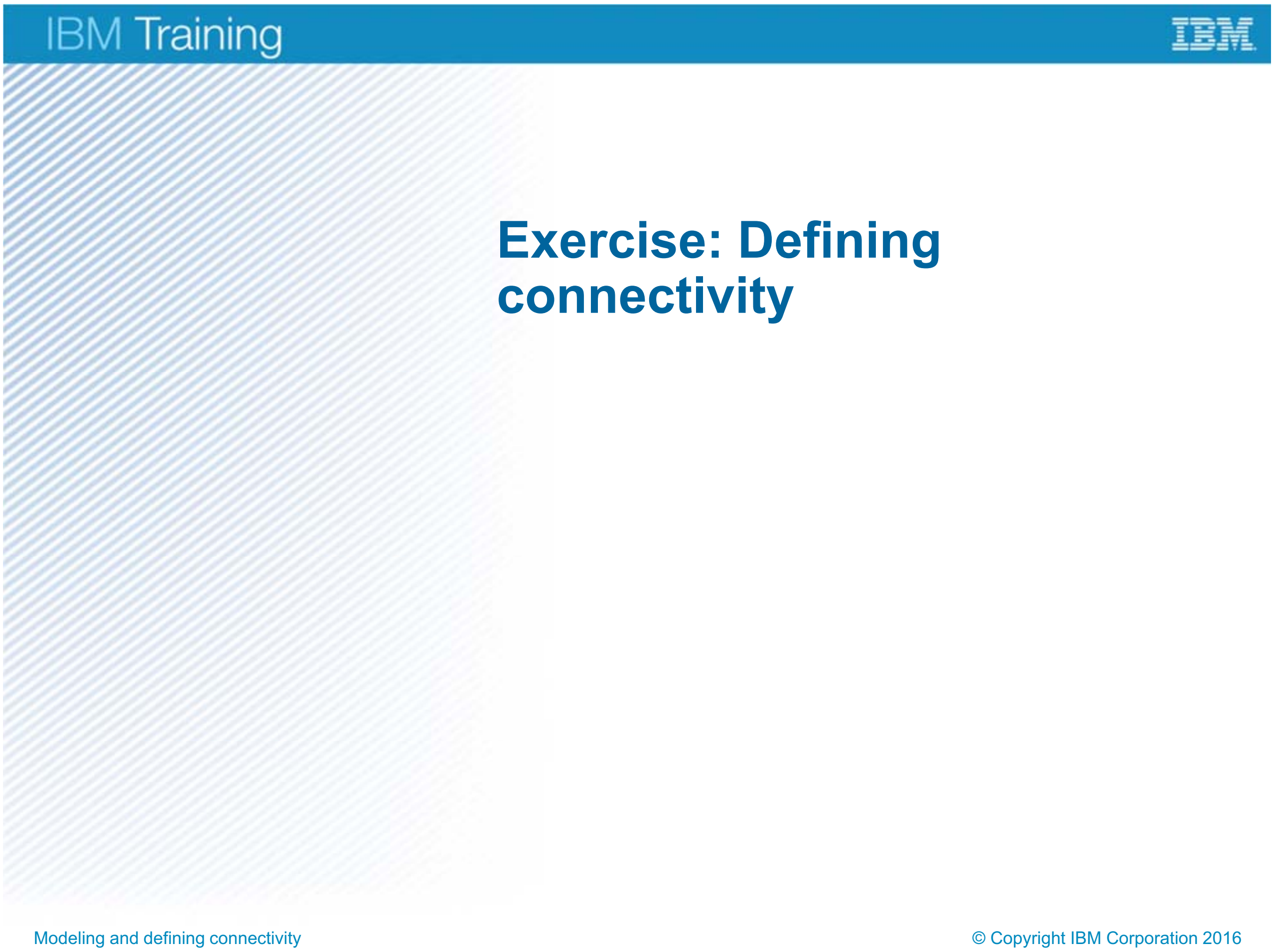


Figure 7-34. Exercise 11

IBM Training

Exercise objectives

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

Modeling and defining connectivity

Figure 7-35. Exercise objectives

Unit 8. Integrating Decision Server Insights

Estimated time

01:00

Overview

This unit explores the integration capabilities of Decision Server Insights.

How you will check your progress

- Checkpoint

IBM Training

Unit objectives

- Describe the integration capabilities of Decision Server Insights
- Explain the exchange event schemas between IBM Integration Bus and Decision Server Insights
- Consume IBM MQ or IBM Information Bus monitoring events
- Create a predictive scoring agent

Integrating Decision Server Insights

Figure 8-1. Unit objectives

IBM Training

Topics

- Overview of integration requirements
- Submitting events from an Integration Bus message flow
- Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights
- Integration with SPSS
- OSGi services

Integrating Decision Server Insights

Figure 8-2. Topics

8.1. Overview of integration requirements

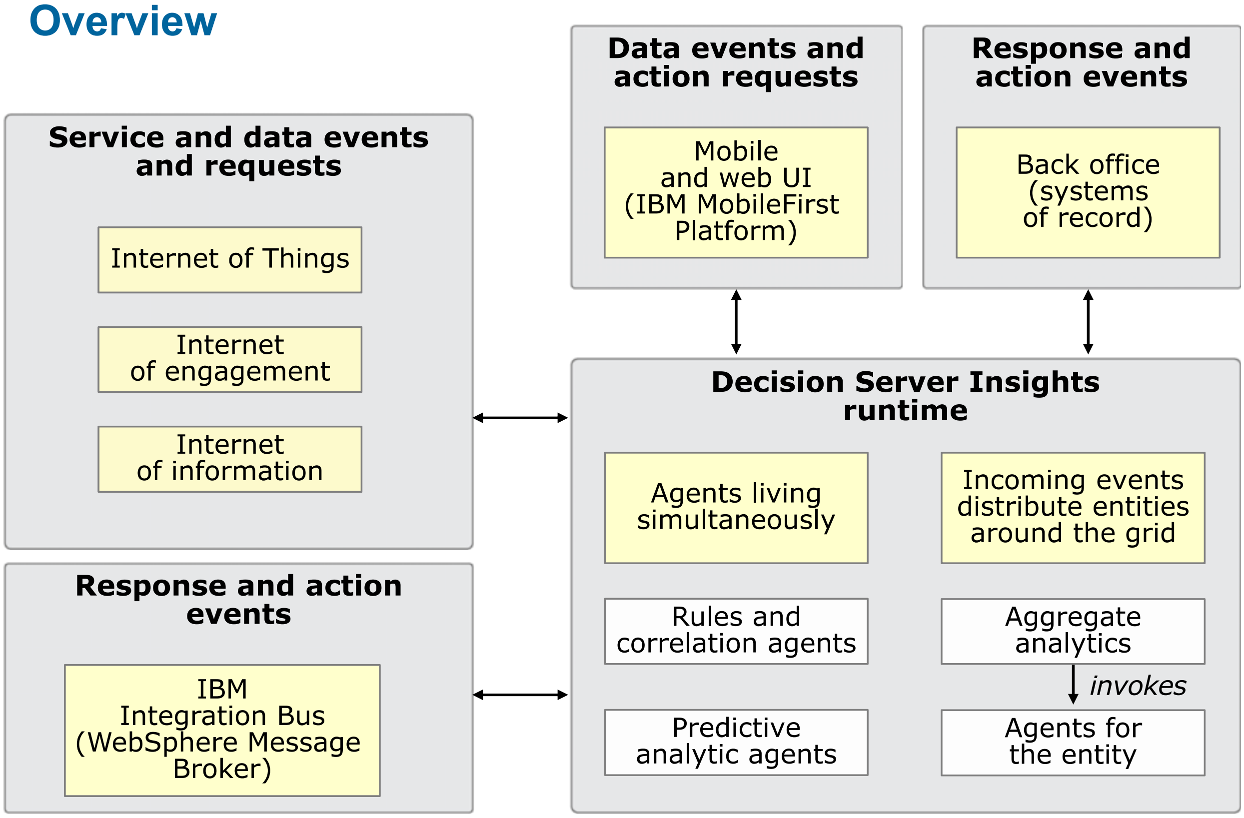
IBM Training

Overview of integration requirements

Figure 8-3. Overview of integration requirements

IBM Training

Overview



Integrating Decision Server Insights

Figure 8-4. Overview

Here you see an overview of the interaction between Decision Server Insights and the (outside) world.

Decision Server Insight must be able to listen to anything and everything.

Event and request sources or producers can include the internet, the Internet of Things, mobile devices, and others. Action consumers can be those same event sources, or they can include systems of record, or other applications, or BPM tools.

The events are brought to a Decision Server Insights runtime, where the data is. The runtime must maintain this wide-angle view of the primary entities, be aware of which events affect which entities, and be able to notify or alert systems about risks or opportunities that were detected. Data also must be replicated to permanent storage.

All this interaction must be handled at internet scale and with continuous availability in terms of hardware and software failure or updates and changes to solutions. As you can imagine, an insight solution that can meet the system requirements is a huge integration task.

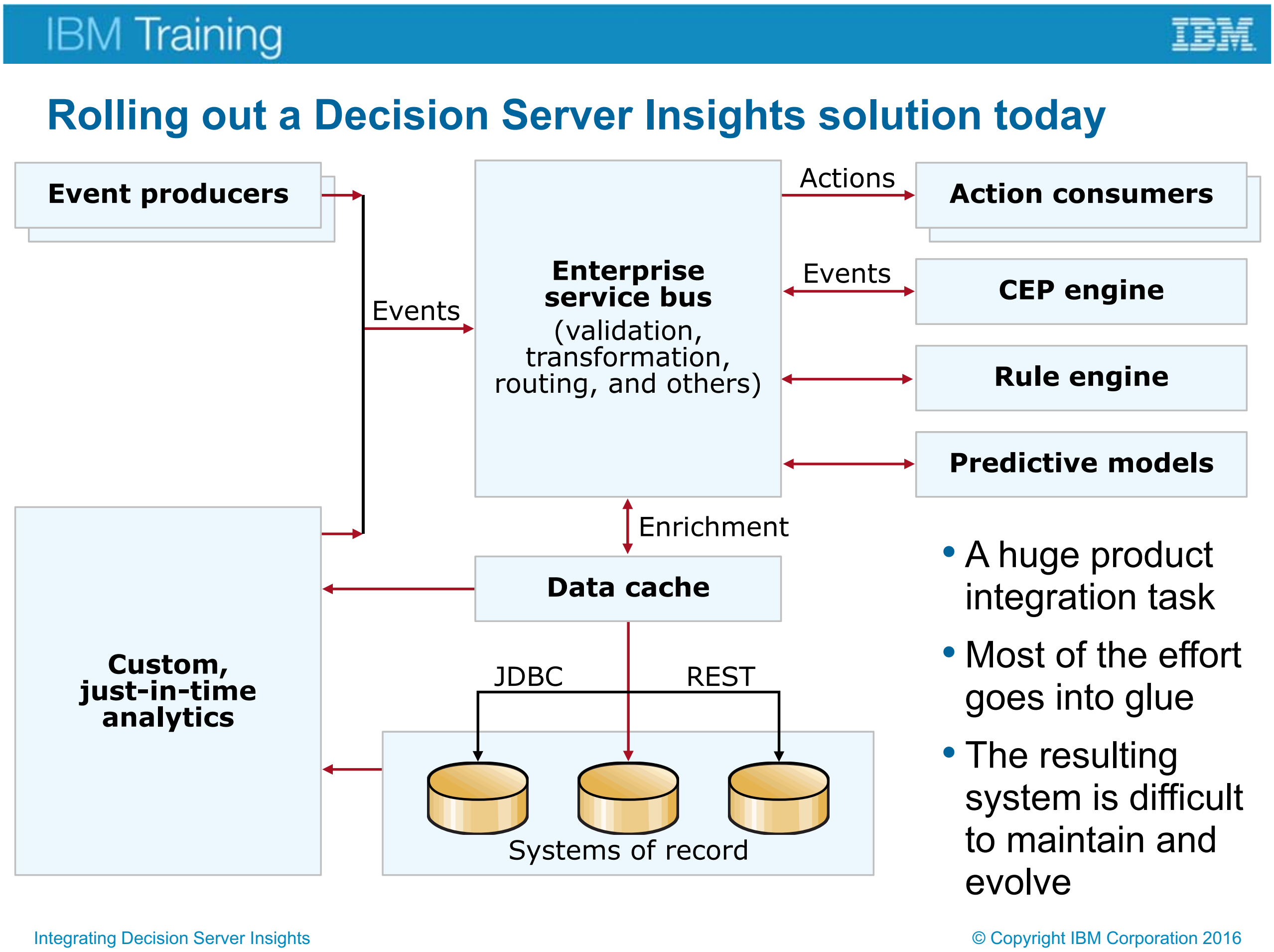


Figure 8-5. Rolling out a Decision Server Insights solution today

To better grasp the integration requirements, this graphic outlines what parts need to work together to roll out an Insights solution.

You need a complex event processing engine, a rule engine, access to predictive models, and scoring servers. You also need custom, just-in-time analytics; data stores and a data cache, plus an enterprise bus to handle incoming and outbound events.

Seeing all of these parts together can help to better appreciate how Decision Server Insights simplified the integration tasks through built-in integration support with other products such as IBM Integration Bus, WebSphere Message Broker, and SPSS.

8.2. Submitting events from an Integration Bus message flow

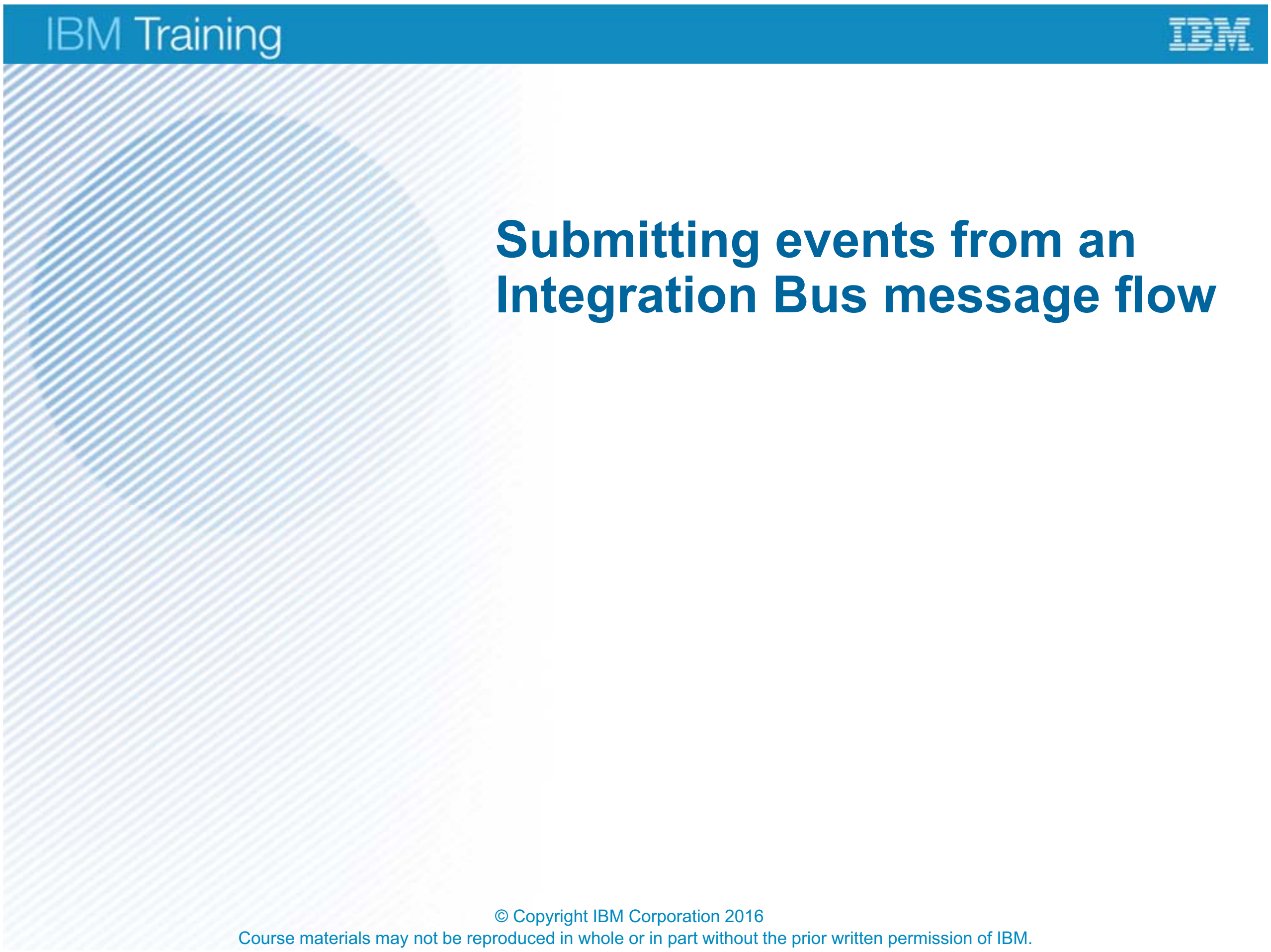


Figure 8-6. Submitting events from an Integration Bus message flow

IBM Training

Submitting events from an Integration Bus message flow

- You can emit events to your solution directly from a message flow by using the gateway API in a Java Compute Node
- You create a message flow that includes a Java Compute Node project, which has a class to submit the events
- You can use the same Java Compute Node class in multiple nodes where all nodes communicate with the same solution and grid

Integrating Decision Server Insights

Figure 8-7. Submitting events from an Integration Bus message flow

To handle the millions of potential incoming event messages, the events can all be published to a bus or a message flow as a single point of entry.

To improve performance, you can bypass the Decision Server Insights connectivity tier and emit events directly to your solution from the message flow. You can create a message flow that includes the Java compute node in the project, which has a class to submit the events. Users can use an IBM Integration Bus message flow as a direct source of events without needing to go through the Insights connectivity layer, thus providing a performance enhancement for the user.

Event schemas must be exchanged between IBM Integration Bus and Decision Server Insights. A Java Compute node is added to an IBM Integration Bus message flow. The Java Compute Node class can be defined to submit the events directly to the Insights solution by using the solution gateway API.

You can use that same Java Compute node in multiple nodes where all the nodes need to communicate with the same solution and compute grid.

IBM Training

Events model

- For events that are already defined in IBM Integration Bus, import the events into Decision Server Insights and use them in solutions
  - Remember: Annotations must be added to the events and their time stamps within the schemas in order for Insights to correctly import them
- For events that are already defined in Decision Server Insights:
  - Import the events by using IBM Integration Toolkit
  - Use them in message flows

Integrating Decision Server Insights

Figure 8-8. Events model

You can exchange event schemas between IBM Integration Bus and Decision Server Insights.

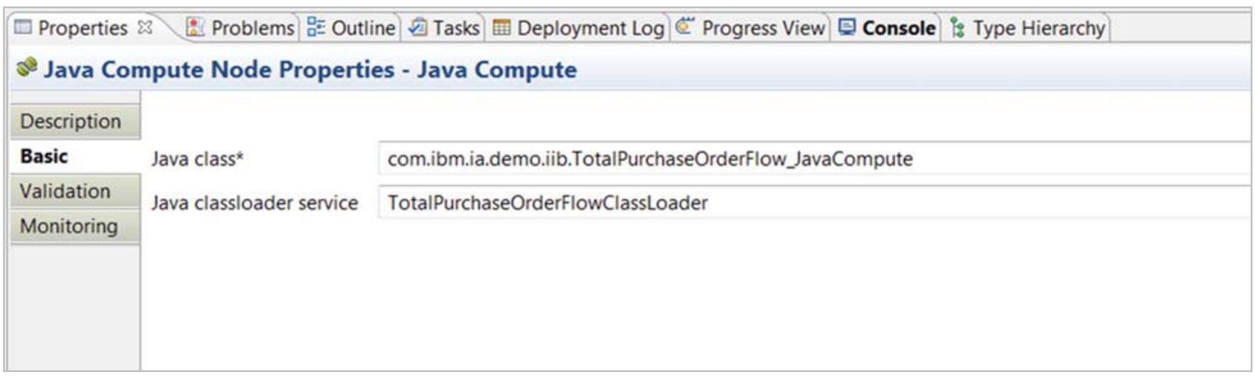
For events that are already defined in IBM Integration Bus, you import the events directly into Decision Server Insights and use in solutions. Remember that to import an XSD into Decision Server Insights, you need to add annotations to the events. By adding annotations, they can be recognized as events, and so the time stamps of those events can be imported correctly.

For events that are already defined in Decision Server Insights, you import the events by using IBM Integration Toolkit and use them in message flows.

IBM Training

Java Compute Node

- After the Java Compute Node project and class are created, add the Java Compute Node to the appropriate position in the message flow
- In the Properties view, the previously defined Java class can be selected
- A name must be given for a class loader service, which is configured later during the process



Integrating Decision Server Insights

Figure 8-9. Java Compute Node

The information that is provided here is aimed at IBM Integration Bus developers. First, create the Java Compute Node Project and Class. Then, write the Java code to submit events to the Insights solution.

After the Java Compute Node Project and class are created, you add the Java Compute Node to the appropriate position in the message flow. In the Properties view, the previously defined Java class can be selected. A name must be given for a class loader service, which you also configure.

IBM Training

Deploying IBM Integration Bus message flows

- Create and configure the Java class loader
- Deploying the application is then a case of dragging and dropping onto the target integration server

Integrating Decision Server Insights

Figure 8-10. Deploying IBM Integration Bus message flows

After the Java class loader is created and configured, you can deploy the application.

8.3. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

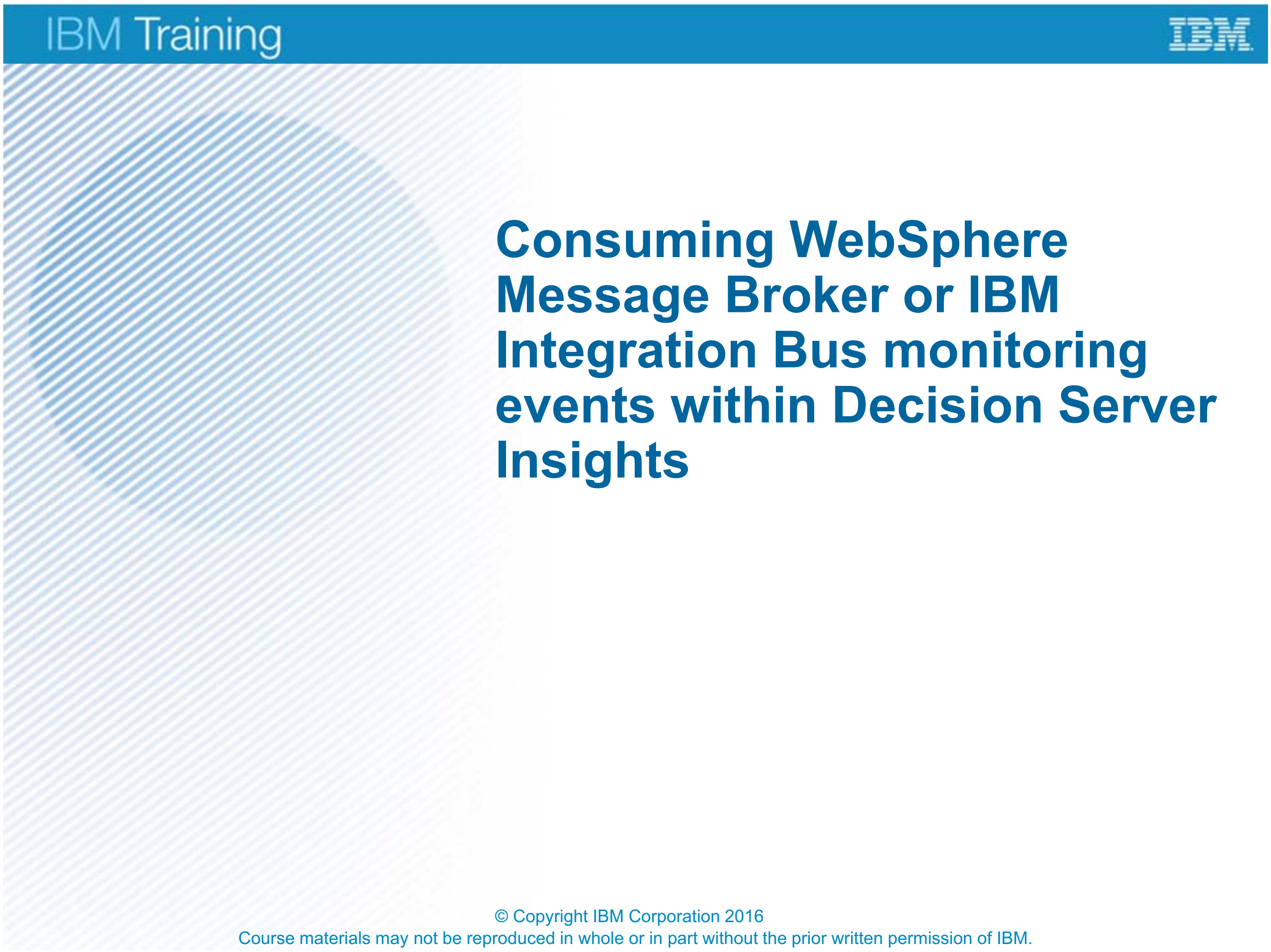


Figure 8-11. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

IBM Training

Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

- Insight Designer includes a wizard that allows the user to import IBM Integration Bus or WebSphere Message Broker monitoring events

- The wizard:
  - Directly imports each of the selected event types in the solution BOM project
  - Creates a transformation for each of the event types so that at runtime they can be transformed into a form that the Insights runtime can understand
  - Configures a JMS binding connectivity definition, containing all the necessary classifiers to process incoming events into the solution

Integrating Decision Server Insights

Figure 8-12. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

IBM Integration Bus and WebSphere Message Broker can be configured to emit ***monitoring*** events that would be useful to Decision Server Insights users. However, the IBM Integration Bus events are structured in a complex manner and by default, cannot be understood in Insights, nor did they have schemas that can be imported into Decision Server Insights.

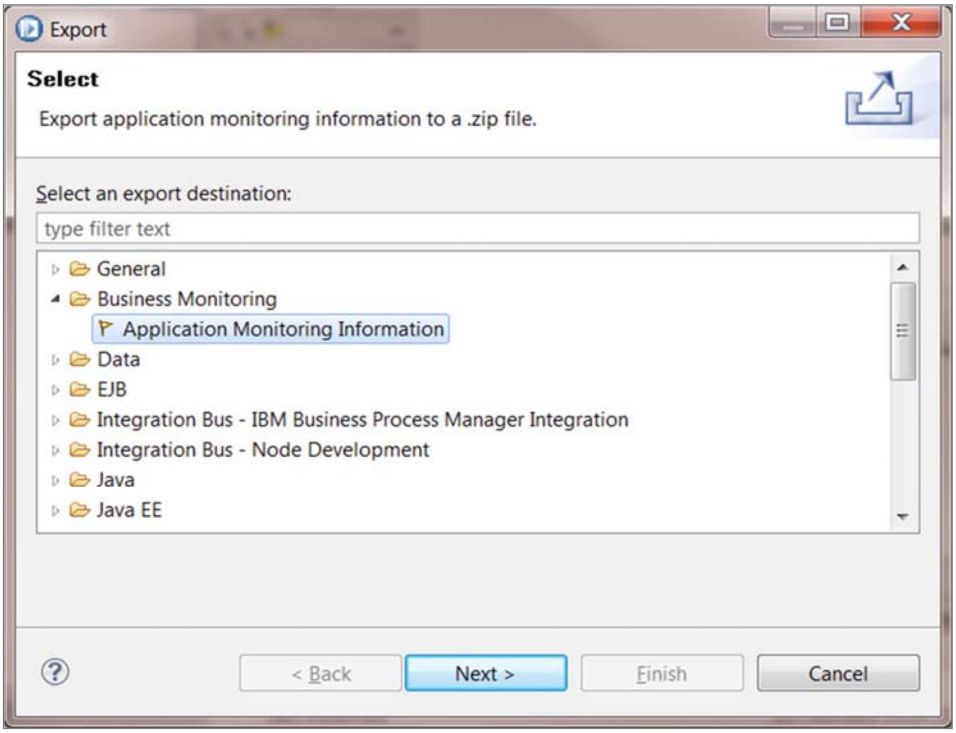
Decision Server Insights provides a wizard that you can use to directly import each of the selected event types into the solution BOM. You can create a transformation for each of the event types, so that at runtime, the events can be understood and transformed into the correct format.

The wizard also configures JMS binding connectivity definitions that contain all the necessary classifiers to process events that are incoming to the solution.

Trying to do these types of transformations manually can be time-consuming and tricky to get right, so this integration feature is a significant time-to-value add.

IBM Training

Step 1



- Configure the Monitoring events on the IBM Integration Bus or WebSphere Message Broker message flows
- Export the **Application Monitoring Information** from the IBM Integration Bus or WebSphere Message Broker tools

Integrating Decision Server Insights

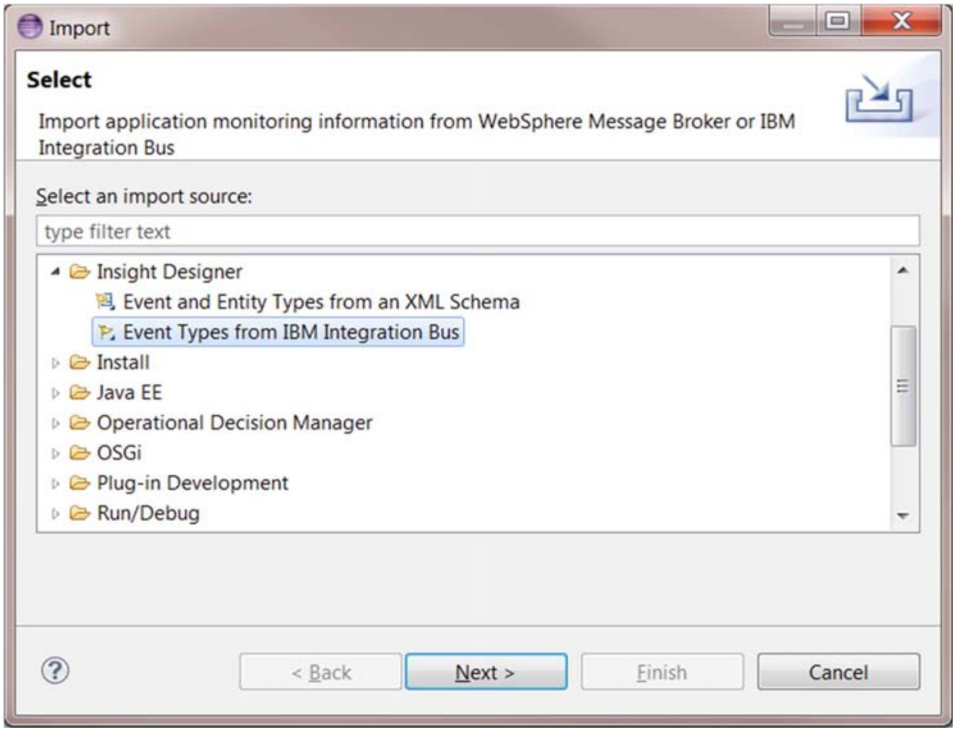
Figure 8-13. Step 1

So, how would this work? The first step is to export the application monitoring information from the IBM Integration Bus or WebSphere Message Broker tools by using the provided wizard.

Through the wizard, you select the flows that you want to export monitoring information from, and it produces a `.zip` file.

IBM Training

Step 2



- To import the event types into a Decision Server Insights solution, use the “Import Event Types from IBM Integration Bus” wizard
- This wizard can be found on the **File > Import** menu, or by using the Solution Map

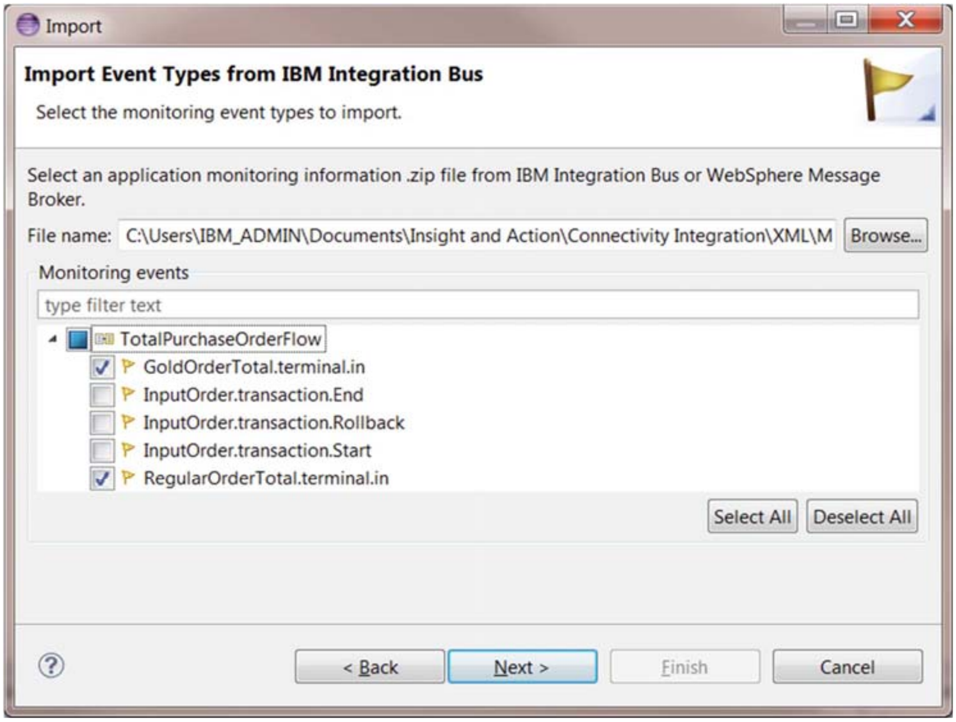
Integrating Decision Server Insights

Figure 8-14. Step 2

The `.zip` file can then be imported into Decision Server Insights by using the wizard that is highlighted, called “Event Types from IBM Integration Bus”.

IBM Training

Step 3



- Select the **Application Monitoring Information** file, which was exported from the IBM Integration Bus or WebSphere Message Broker tools
- The wizard then lists the events that it finds within the file so that the ones that should be imported can be selected

Integrating Decision Server Insights

Figure 8-15. Step 3

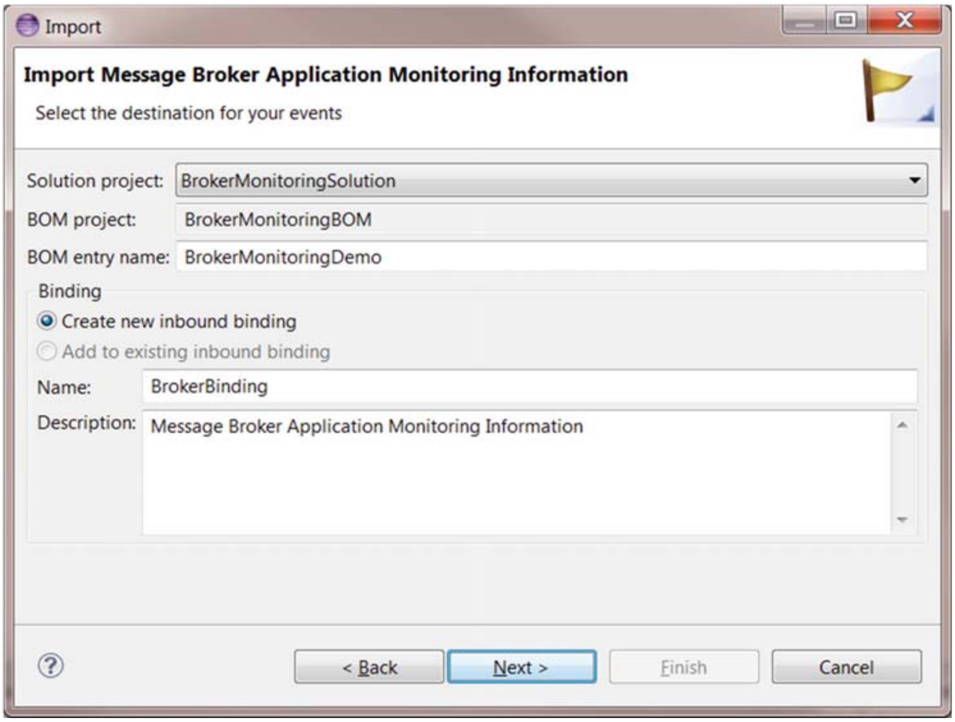
The first step is to select the `.zip` file that you exported from IBM Integration Bus or WebSphere Message Broker. You can then import that compressed file into your Decision Server Insights solution.

Importing the compressed file produces a list of monitoring events that are contained in that monitoring information file. You then select which events you want to import.

In the screen capture, you see that five monitoring events are listed, but only two are of interest and selected.

IBM Training

Step 4



- A number of elements can be specified:
  - The solution, in which to import the event types
  - The BOM entry name, where the event types should live
  - A binding name for the JMS connectivity definition

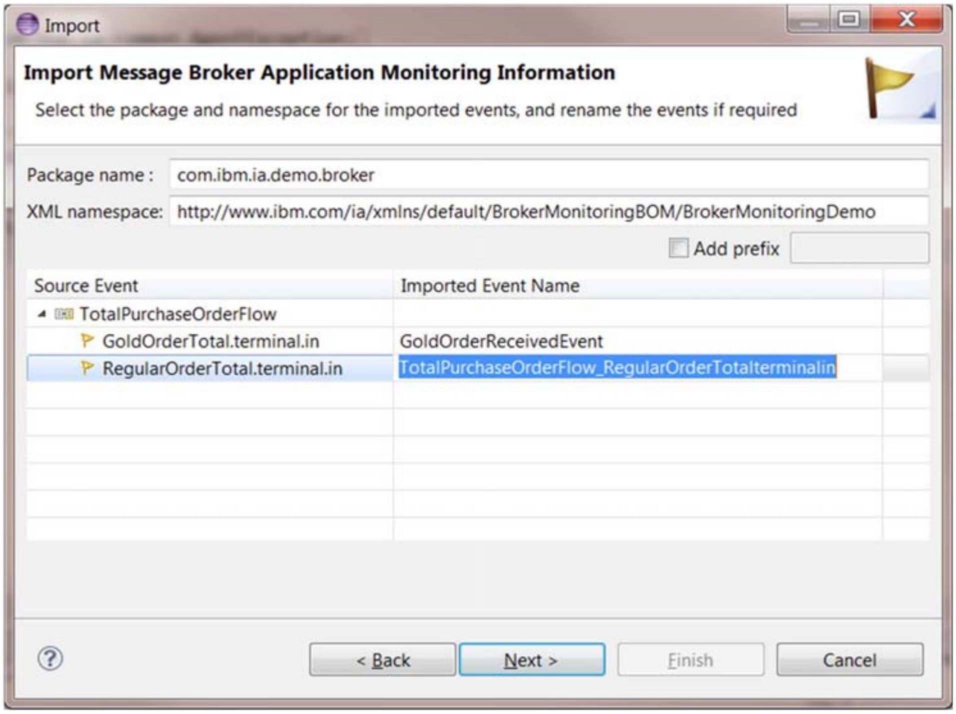
Integrating Decision Server Insights

Figure 8-16. Step 4

On the next page of the wizard, you see a number of elements that you specify. These elements include the solution into which you import the event types, the BOM entry name where the event types should live, and a binding name for the JMS connectivity definition. You can also select whether you want to create an inbound binding or add to an existing binding.

IBM Training

Step 5



- On the third page, you can specify the package for the events and the namespace for the transformed events

Integrating Decision Server Insights

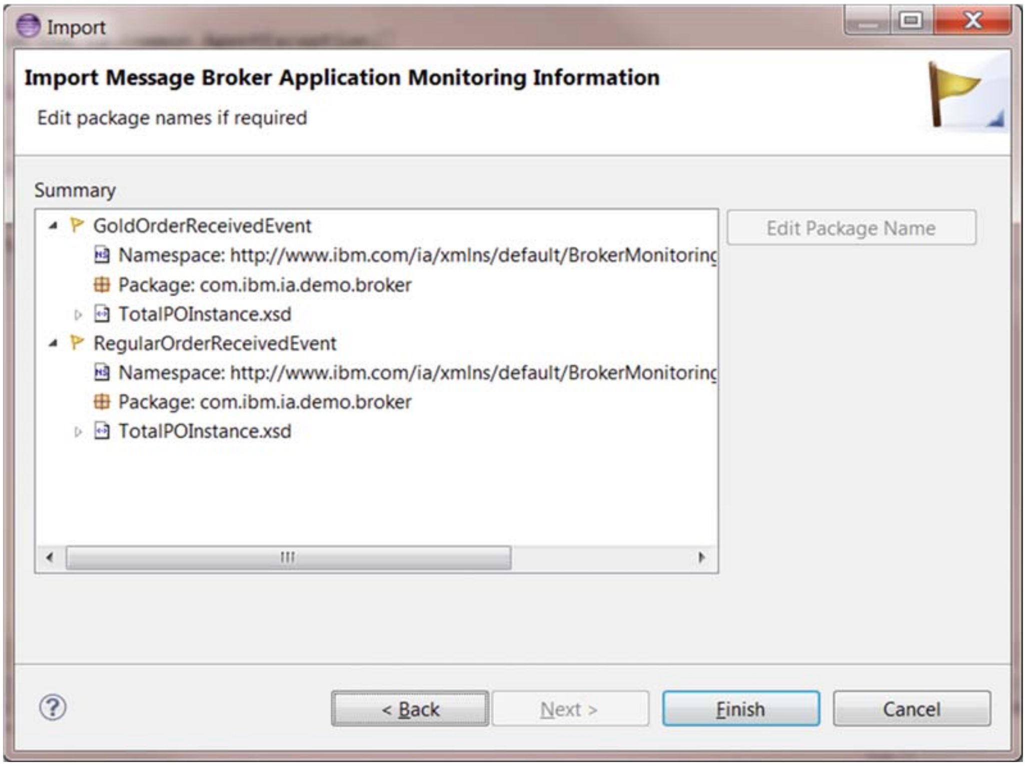
Figure 8-17. Step 5

Define the package that these events should belong to.

You have an opportunity here to rename the monitoring events. You can use a name that has a more meaningful verbalization than the default name that comes from the IBM Integration Bus and WebSphere Message Broker tools.

IBM Training

Summary of the events to be imported



Integrating Decision Server Insights

Figure 8-18. Summary of the events to be imported

The final page of the wizard provides a summary of the events to be imported, the namespaces and packages, and it notifies you about any additional schemas that are being imported.

IBM Training

The result

- On completion, a new BOM entry is added to the BOM project
- This new BOM entry contains the imported events
  - In this case, it also imported an additional schema that was required for the events
- A new connectivity definition (`.cdef`) file was added, defining a binding to classify the new events
- Two transformations were added to the Transformations folder of the solution
- The events are now ready for the user to use within agents, aggregates, and others
- All that remains for the user to do is to add a second connectivity definition for an inbound JMS endpoint that uses the generated binding
  - This connectivity definition can be in a separate connectivity definition file

Integrating Decision Server Insights © Copyright IBM Corporation 2016

Figure 8-19. The result

On completion, you have a new BOM entry that is added to the BOM project. The new BOM entry contains the imported events and any additional schemas that were required for those events.

You also have the connectivity definition, which defines the binding to classify the new events. The required transformations are added to the transformations folder of the solution, and events are ready for use within your agents or aggregates or other artifacts.

All that is left for you to do is to add another connectivity definition for an inbound JMS endpoint that uses the generated binding.

The endpoint needs to be configured to listen to the appropriate monitoring topic of the broker and that can be done by using the Export Connectivity Configuration wizard.

If monitoring event types change, you might need to update your Insights solution, and you can use the wizard again to reimport the monitoring definition. If the BOM name is the same as the first time you ran the wizard, all event types can be updated automatically.

8.4. Integration with SPSS

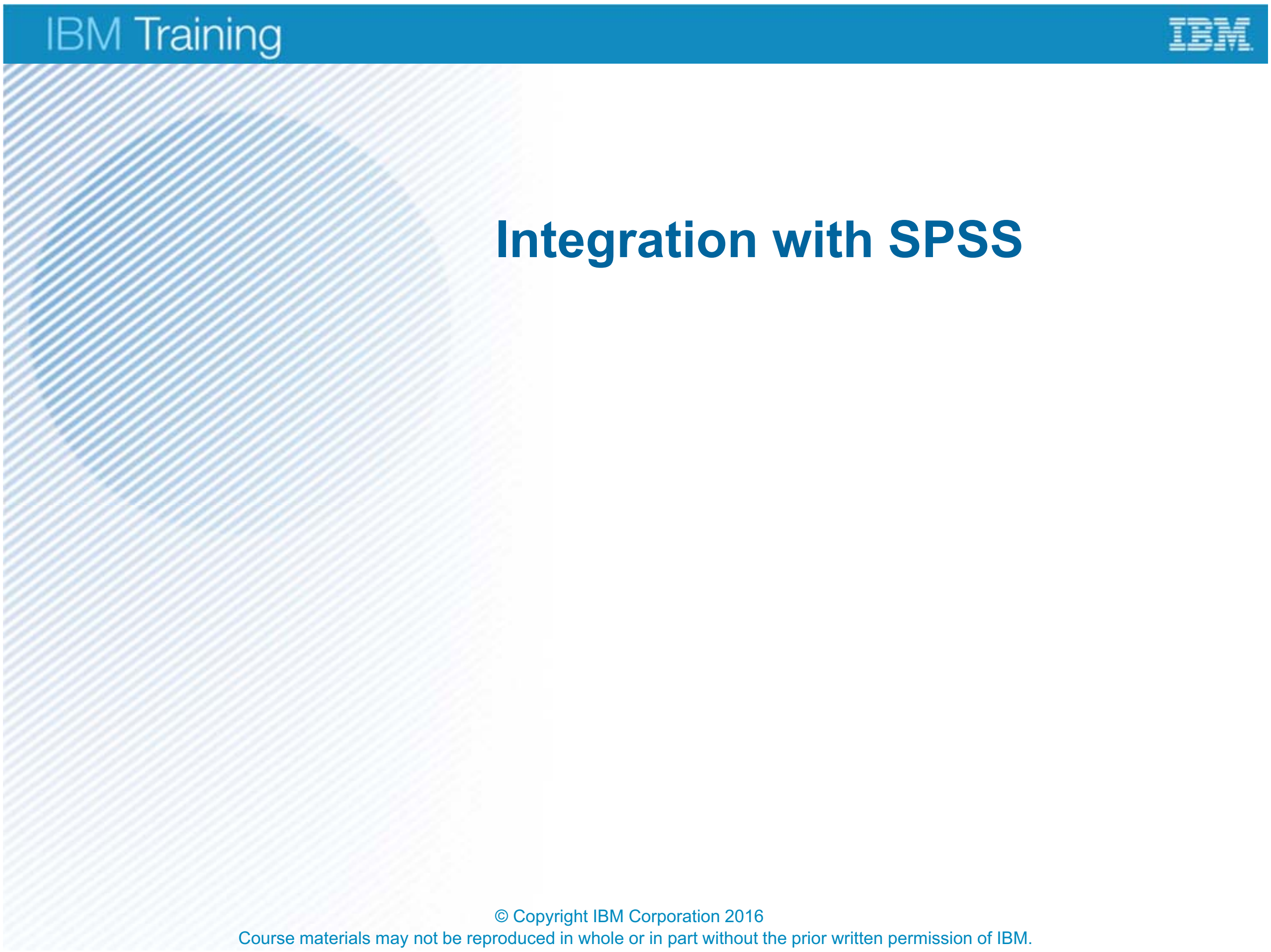


Figure 8-20. Integration with SPSS

IBM Training

SPSS

- SPSS Modeler
  - Set of data mining tools
  - Used to develop predictive models
- SPSS Collaboration and Deployment Services
  - Centralized, secure storage of analytical assets for widespread use and deployment
- SPSS scoring model
  - The set of rules, formulas, or equations that are extracted from your source data and that you can use to generate predictions

Integrating Decision Server Insights

Figure 8-21. SPSS

What is SPSS?

SPSS stands for a Statistical Package for Social Sciences, and it refers to a whole family of products. Decision Server Insights focuses on the SPSS Modeler and SPSS Collaboration and Deployment Services tools.

SPSS Modeler is a set of data mining tools that you use to develop your predictive models. The SPSS Collaboration and Deployment Services tool has a centralized repository where you can store the predictive models that were created in SPSS Modeler to make them available for widespread use and deployment.

If you look at the SPSS Modeler documentation, the definition of the scoring model is:

> *A set of rules, formulas, or equations that are extracted from your source data and that you can use to generate predictions.*

For example, you can create a model that takes a customer’s purchasing history as input and returns the likelihood that the customer will make a purchase in the next month.

IBM Training

Predictive analytics

- Integration is supported on SPSS Collaboration and Deployment Services Server
  - SPSS installation and licensing are separate from Operational Decision Manager
  - Supported on versions 5 and 6 of SPSS Collaboration and Deployment Services Server
- Integration through predictive scoring agents
  - Specialized Java agent that retrieves a scoring configuration from a predictive model
  - Agents are created in Insight Designer by using a wizard that connects to a running SPSS scoring service to obtain metadata about the scoring configuration
  - The predictive service must be deployed on an SPSS Collaboration and Deployment Services Server

Integrating Decision Server Insights

Figure 8-22. Predictive analytics

SPSS integration is supported through Decision Server Insights predictive scoring agents.

All the processing and decision logic is designed within agents, and the predictive scoring agent is a Java agent with the ability to retrieve scores from predictive models on an external SPSS scoring server.

When you create the predictive scoring agent, you use information about the predictive model that you want to invoke to generate Java code. This way, you must complete only a few to-do tasks to get the agent up and running.

Decision Server Insights has built-in features to allow predictive scoring agents to easily connect to an SPSS scoring server, without need for additional boot-strapping. You just set three solution properties, and all the rest of the integration work is taken care of.

Why use predictive analytics? The results from these predictive models can help make decisions at the time of interaction. For example, when you consider the likelihood of a customer to make a purchase, if the likelihood is low, you might consider offering discount coupons to encourage activity.

IBM Training

How it works

- Step 1: Constructing the input to the scoring model
  - Data from the event, bound entity, related entities, global aggregates
  - Decide which data must be sent to the SPSS model
  - SPSS model expects certain inputs
  - Decide where to get the data for each of those inputs
  - Insights data is mapped to a form that is required by the SPSS predictive model
- Step 2: Invoke scoring
  - Inputs are sent
- Step 3: Decide what to do with the output
  - Send returned values as an event?
  - Store the score for later reference, for example with the bound entity

Integrating Decision Server Insights

Figure 8-23. How it works

So, how does it all work? You go through three main steps. The first step is to construct the input to the scoring model.

You have access to a large quantity of data from the event, the bound entity, related entities, aggregates, and others. You need to decide which information gets sent to the SPSS model. The SPSS model has certain inputs that it expects to receive, so you decide where to get that data from for each of those inputs.

The second step is to invoke the scoring server by sending the input, and then you receive the scoring output.

Finally, the third step is to decide what to do with the output. The scoring values that are returned might be useful to other agents. So you might want to store the score in the bound entity so that it is available to other agents.

IBM Training

Creating a predictive scoring agent

- When you create a predictive agent, you create it with the wizard, but also include connection details to a running scoring server
- Create the agent descriptor
- The wizard prefills Java code
- Define the solution properties

Integrating Decision Server Insights

Figure 8-24. Creating a predictive scoring agent

To create the agent, you use the predictive scoring agent wizard. The first page of the wizard is similar to the Java agent wizard. However, this wizard also needs connection details of a running Collaboration and Deployment Services server that contains the scoring configuration that you want to invoke. The wizard connects to the server to retrieve available configurations for you to select.

After the agent is created, you define the descriptor, as you do for other agents. For the code itself, the wizard generates much of the Java code for you, according to the scoring configuration that you selected. If the configuration requires several inputs, the code for each of those inputs is already in place, and you enter where to grab that input from.

IBM Training

Troubleshooting

- Connection problems
  - Check that the URL is in the correct form
  - Check to see whether the scoring server is running
  - Ensure that credentials are correct
  - Contact SPSS administrator

Integrating Decision Server Insights

Figure 8-25. Troubleshooting

Limitations: SPSS is included separately.

If you run into problems while trying to create the agent with the wizard, check your connection details, credentials, and that the URL is in the correct form. You should also verify that you have the right permissions.

You might also need to confirm that the SPSS server is running and not locked, you might need to contact the SPSS administrator.

IBM Training

Suggested practices

- Values for the SPSS connection solution properties should be defined at deploy time in the `server.xml` file, rather than within the solution
  - Uses the WebSphere Developer Tools server configuration editor for encoding the password so that it does not need to be stored in plain text
  - You must still define the property names in the `solution_properties.xml` file in the solution project, even if the values are being assigned on the server
  - If the password is encoded, the `encoded="true"` flag must also be specified in the `solution_properties.xml` file
- Keep decision logic separate from SPSS invocation logic for a clearer model
  - If you want the scores to be immediately used within agent logic and decision making, then create a rule agent or a Java agent to undertake that logic
- Scores can be sent to agents by either:
  - Emitting an event, which triggers the agent, *or*
  - Storing the scores in the bound entity and making sure that the predictive scoring agent has a higher priority than the subsequent agents

Integrating Decision Server Insights

Figure 8-26. Suggested practices

Integration with ODM Decision Server Insights supports SPSS Collaboration and Deployment Services versions 5 and 6.

This slide lists some suggested practices, such as specifying the values for the SPSS connection at deployment time, not design time. A key suggestion is to maintain your decision logic separate from the SPSS invocation logic. It clutters the model when your SPSS agents also make decisions in addition to retrieving information from SPSS servers. The predictive scoring agent should be used to retrieve only scores. The scores can then be used in your rule agents to apply business logic.

8.5. OSGi services

Figure 8-27. OSGi services

IBM Training

OSGi Services

- An OSGi service is a service that is defined by a Java interface and packaged within an OSGi bundle
- A service can be called from a Java agent or a rule agent to provide access to external systems or advanced computation capabilities

Integrating Decision Server Insights

Figure 8-28. OSGi Services

An OSGi service is a service that is defined by a Java interface and packaged within an OSGi bundle.

A service can be called from a Java agent or a rule agent to provide access to external systems or advanced computation capabilities.

If you want to call an OSGi service from a Java agent or a rule agent, you must make the service available to Insight Server. You can either deploy the OSGi bundle that contains your service independently of the solution, or package the OSGi bundle inside the solution.

- To package the bundle in the solution archive, you add a reference in the solution project.
- To package an OSGi service inside a solution, you add a reference to the OSGi plug-in project in the solution project.

To access an OSGi service from the rules, you must create a business object model (BOM) that maps to the Java code of the service. First, you create a project to contain the BOM, and then you reference the project in the rule agent.

The OSGi service is represented by a verbalized BOM class that corresponds to the Java interface of the OSGi service. This BOM class must contain a custom property named OSGi.service. The value of this property is the fully qualified name of the Java interface.

The BOM class for the OSGi service must be created in a separate rule project because it cannot be hosted by the solution BOM project. This rule project must have a reference to the plug-in project that contains the Java interface. This plug-in project serves as the Java Execution Model (XOM).

To be accessible in the rules, a method of the service must have a corresponding method in the BOM. This corresponding method must be static, and have the same name and the same number of parameters. Each parameter must have a type that corresponds to the Java type of the parameter in the Java interface. The corresponding BOM type usually has the same name as the Java type.

IBM Training

Unit summary

- Describe the integration capabilities of Decision Server Insights
- Explain the exchange event schemas between IBM Integration Bus and Decision Server Insights
- Consume IBM MQ or IBM Information Bus monitoring events
- Create a predictive scoring agent

Integrating Decision Server Insights

Figure 8-29. Unit summary

IBM Training

Review questions

1. True or false: Decision Server Insights can receive events that are submitted from IBM Integration Bus message flow.

2. What is the simplest way to integrate SPSS scoring model with Decision Server Insights?
   a. Create a rule agent
   b. Create global entity aggregate
   c. Create a predictive scoring agent

3. True or false: An IBM Integration Bus event type can be imported into a Decision Server Insights solution.

Integrating Decision Server Insights

Figure 8-30. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. True or false: Decision Server Insights can receive events that are submitted from IBM Integration Bus message flow.
   Answer: True

2. What is the simplest way to integrate SPSS scoring model with Decision Server Insights?
   a. Create a rule agent
   b. Create global entity aggregate
   c. Create a predictive scoring agent

   Answer: c. Create a predictive scoring agent

3. True or false: An IBM Integration Bus event type can be imported into a Decision Server Insights solution.
   Answer: True

Integrating Decision Server Insights

Figure 8-31. Review answers

Unit 9. Configuring Insight Server

Estimated time

01:00

Overview

This unit explains how to configure Decision Server Insights.

How you will check your progress

- Checkpoint
- Exercises

IBM Training

Unit objectives

- Describe WebSphere eXtreme Scale basics
- Describe the Decision Server Insights reference topology
- Design and configure a production topology

Configuring Insight Server

Figure 9-1. Unit objectives

IBM Training

Topics

- eXtreme Scale basics
- Topologies
- Sizing
- Configuring production topology

Configuring Insight Server

Figure 9-2. Topics

9.1. WebSphere eXtreme Scale basics

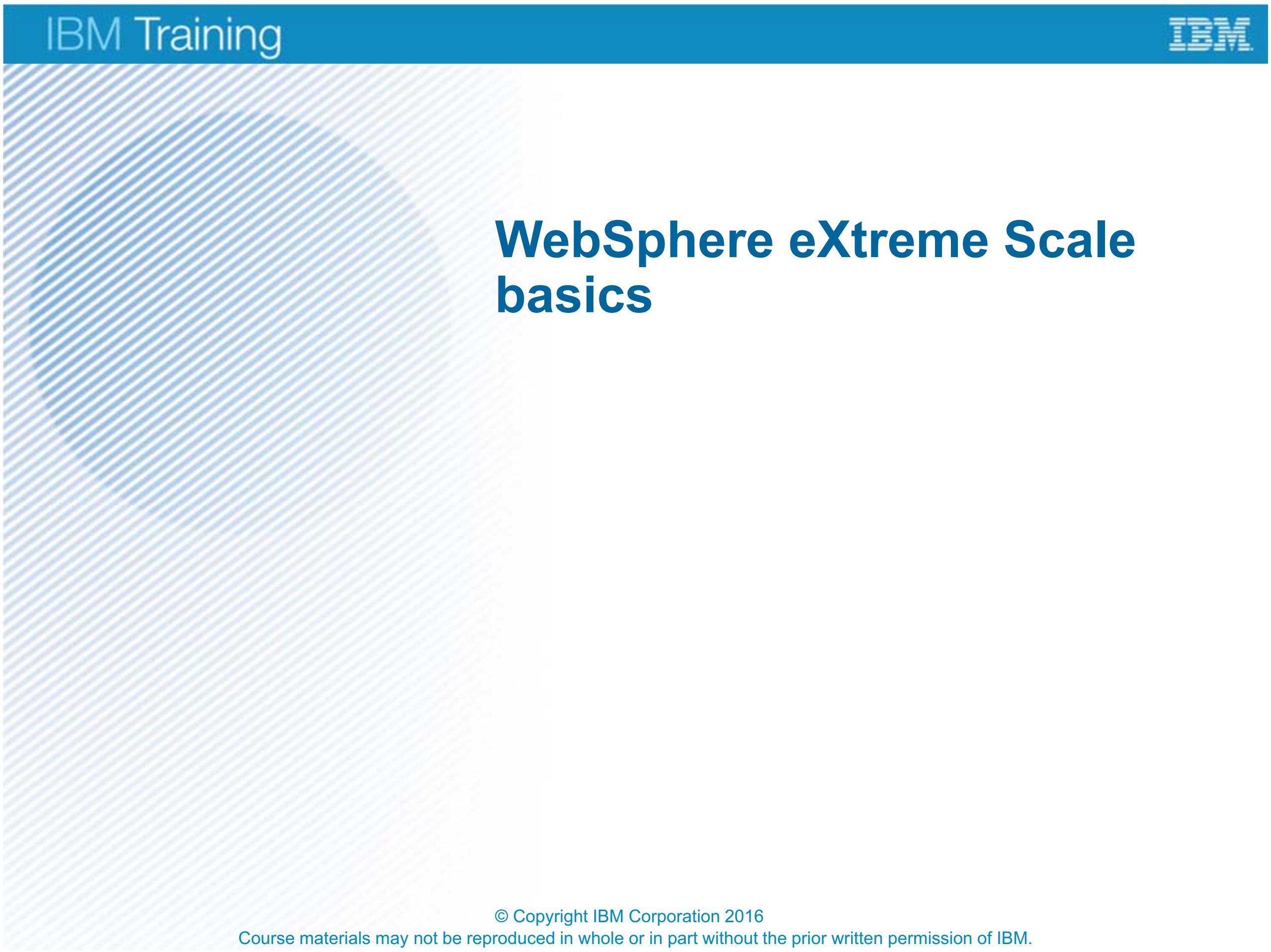


Figure 9-3. WebSphere eXtreme Scale basics

IBM Training

What is IBM WebSphere eXtreme Scale

- eXtreme Scale is an elastic, scalable, in-memory data grid
  - Dynamically processes, partitions, replicates, and manages application data across hundreds of servers
  - Provides transactional integrity and transparent failover
- Principles of extreme scalability
  - Put data in memory
  - Partition the data to enable linear horizontal scale-out
  - Caching

Configuring Insight Server

Figure 9-4. What is IBM WebSphere eXtreme Scale

IBM WebSphere eXtreme Scale is an elastic, scalable, in-memory data grid. It dynamically processes, partitions, replicates, and manages application data across hundreds of servers. eXtreme Scale can perform massive volumes of transaction processing with high efficiency and linear scalability. It provides transactional integrity and transparent failover to ensure high availability, high reliability, and consistent response times.

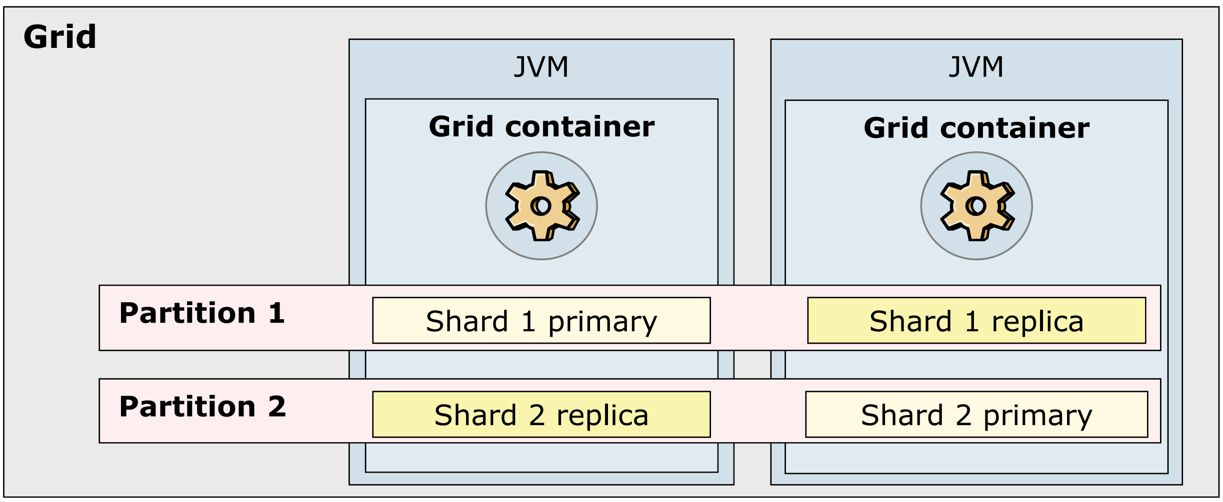
The term elastic means the grid monitor and manages itself, and is self-healing by automatically recovering from failures. It allows *scale-out*, which means memory capacity can be added while the grid is running without a restart. And it allows *scale-in*, which is dynamic removal of memory capacity. This elastic capability means that the grid can automatically recover from failures.

The principles of extreme scalability include putting data in memory, partitioning the data to allow horizontal scale-out, and caching.

IBM Training

Understanding a grid

- *Grids* divide the data into partitions
- Each *partition* holds an exclusive subset of the data
- Within the partition, the data is stored in *shards*
  - Primary shard contains the primary copy of the subset of data
  - Replica shards contain copies of the primary shard



Configuring Insight Server

Figure 9-5. Understanding a grid

Grids, partitions, and shards are the major building blocks when you do capacity planning for WebSphere eXtreme Scale. Maps are important as a way for you to determine the size of the data that you store in the grid. You need to calculate the data that is stored in each map in your grid to arrive at a total size of data that you want to store.

- A *grid* divides the data set into partitions
- Each partition holds an exclusive subset of the data. The data can be partitioned based on the key and the data for a partition is stored at run time in a set of shards.
- A shard represents a partition that is placed on a *container*. Each partition has an instance that is a *primary shard* for the primary copy of the data. You can also configure a number of *replica* shards. The replica shards are either synchronous or asynchronous.

The relationship between the grid, partitions, and shards is illustrated on the slide.

A *container server* physically holds the application data for the grid.

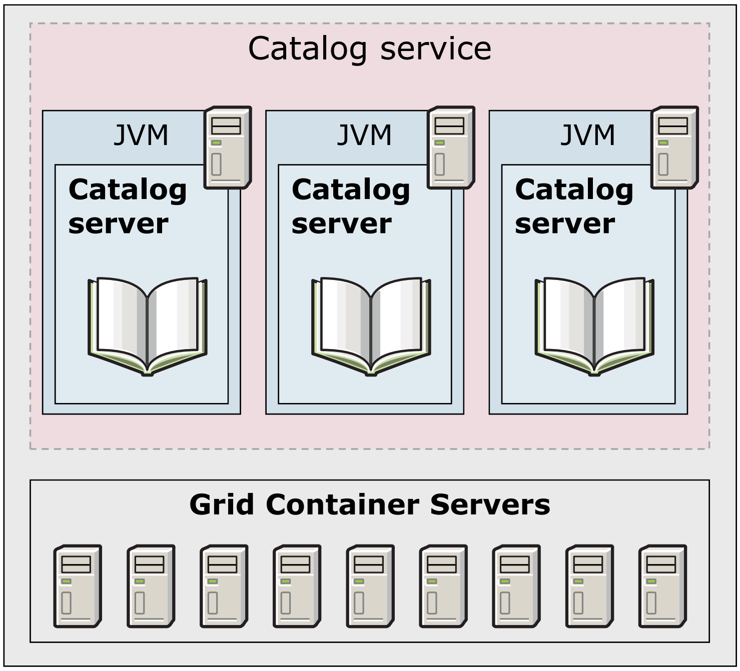
A grid has one or more partitions. Each partition is represented by a primary shard that is hosted on a container, and optionally one or more replica shards that are hosted on other containers.

Any number of containers can be run on a host (vertical scaling) and any number of hosts can run extra containers (horizontal scaling). A key feature of an eXtreme Scale grid is that more containers can be added dynamically on any server in the network.

IBM Training

Catalog service

- *Catalogs* become central nervous system of the grid



Catalog service:

- Keeps track of partitions and shards
- Redistributes shards when a container joins or leaves the grid
- Monitors grid health
- Provides data location services to grid clients
- Enforces policies and rules
- Ensures high availability and group service

Configuring Insight Server

Figure 9-6. Catalog service

The *catalog service* parses the deployment policy and grid configuration files. It uses their definitions to control placement of shards across the available container servers in the grid. It also discovers and monitors the health of the containers, automatically balancing shard placement as necessary when a new container is added to the configuration, or if a container is stopped.

One or several catalog servers can run in your grid environment. However, only one is chosen automatically as the *master catalog*. The catalog service is designed to service hundreds or thousands of container servers.

WebSphere eXtreme Scale uses logical placement rules (implemented by the catalog server) to ensure that replica shards are held in containers that are running on different host machines than the primary shard. WebSphere eXtreme Scale determines the types and placement of replica shards by using a deployment policy, which specifies the minimum and maximum number of synchronous and asynchronous replicas.

IBM Training

WebSphere eXtreme Scale terminology (1 of 3)

Term	Definition
Map	A cache that stores Java objects based on key-value pairs.
Mapset	A collection of logically related maps that can be partitioned and replicated over a number of servers.
Grid	A collection of mapsets that might span multiple Java virtual machines and that you can connect to and access data.
Partition	Logical representation of a subset of the data in the mapset, plus any replicas that each subset might have. • The number of partitions (*n*) is a configurable attribute of a mapset • Partition numbering starts at 0 • The mapset data is distributed across the *n* partitions
Replica	A copy of the primary data that is stored remotely about the primary and other replicas. • *Synchronous replica*: Updated transactionally when the primary is updated to ensure no data loss when the primary data is lost. • *Asynchronous replica*: Updated after the transaction is complete for faster transaction performance, but with increased risk of data loss in the face of failures.

Configuring Insight Server

Figure 9-7. WebSphere eXtreme Scale terminology (1 of 3)

On these slides, you see a list of eXtreme Scale terms that are relevant to Decision Server Insights.

IBM Training

WebSphere eXtreme Scale terminology (2 of 3)

Term	Definition
Shard	Provides the physical memory storage for the contents of a partition. • *Primary shard*: Contains the primary partition. • *Replica shards*: Backup of all the data in the primary shard.
Grid container	A container for the shards (all the cached data).
Container server	A Java virtual machine that runs WebSphere eXtreme Scale and hosts one or more grid containers. Decision Server Insights uses a WebSphere Application Server Liberty profile as its grid container servers.
Catalog server	Provides management of the entire grid. When there is more than one catalog server, one of them is the *master* or *primary* catalog server, and coordinates work among the servers to provide the catalog services.
Catalog service	• Keeps track of partitions and shards. • Redistributes shards when a container joins or leaves the grid. • Monitors the health of the grid. • Provides data location services to grid clients.

Configuring Insight Server

Figure 9-8. WebSphere eXtreme Scale terminology (2 of 3)

For Decision Server Insights, the preferred grid topology has only one grid container per grid container server. In addition, only one grid container server runs on each host machine. For this reason, this course uses the term *container* to refer to both the grid container (runtime) server and grid container, except in cases where appropriate differentiation is required.

IBM Training

WebSphere eXtreme Scale terminology (3 of 3)

Term	Definition
Catalog service domain	The group of catalog servers, together with the group of container servers that they oversee.
Quorum	An agreement between members of the catalog server group on what needs to be done for grid lifecycle operations. • For example, when a network brownout occurs, communication between catalog servers might be lost, and more than one catalog server becomes the primary server; known as *split-brain syndrome.* • If a split-brain syndrome occurs when quorum is enabled, grid work is suspended. • Recovery from the loss of quorum typically requires manual intervention.
Majority quorum	Ensures that quorum is achieved and grid work can be performed while more than half of the catalog service members are active and aware of each other.
Catalog server cluster endpoints	Configured for container servers to establish a communications link with the catalog servers. These endpoints become part of the catalog service domain (which means they are part of the grid).

Configuring Insight Server

Figure 9-9. WebSphere eXtreme Scale terminology (3 of 3)

9.2. Topologies

IBM Training

Topologies

Figure 9-10. Topologies

IBM Training

Designing a topology

- Consider:
  - How many servers of each type you require, and allocate unique names to them
  - Whether to configure persistence for your grid data
- Inbound and outbound connectivity servers
  - Use one of each?
  - HTTP or JMS?
- Catalog service
  - High availability requires 3 or more catalog servers
  - Each catalog server should be on a separate machine
- Container servers
  - High availability requires a minimum of 3 servers
  - How many solutions must run in the grid?
  - What is the volume of events that must be processed?
  - How many partitions must be allotted for the primary data and replicas?

Configuring Insight Server

Figure 9-11. Designing a topology

When designing your topology, consider the questions that are listed here.

IBM Training

Reference topology goals

- The grid must be highly and continuously available in a normal operation mode
  - Must withstand the loss of one container server without any loss of data and without loss of access to data
  - Must tolerate the controlled shutdown of one container at a time for the purpose of applying "rolling updates" or for any other maintenance activity
  - Should accept and use new containers (within a limit that is determined by the configured number of partitions and replicas)
  - Should tolerate the simultaneous loss of 2 containers with the accepted risk of some data loss
- Catalog service should be highly available and rely on majority quorum
- Grid data must be recoverable in case of disaster or in case of a controlled shutdown of the grid
- The system should tolerate the failure of at least one inbound and one outbound connectivity server
- Event throughput should not suffer significantly as a result of the previous requirements

Configuring Insight Server

Figure 9-12. Reference topology goals

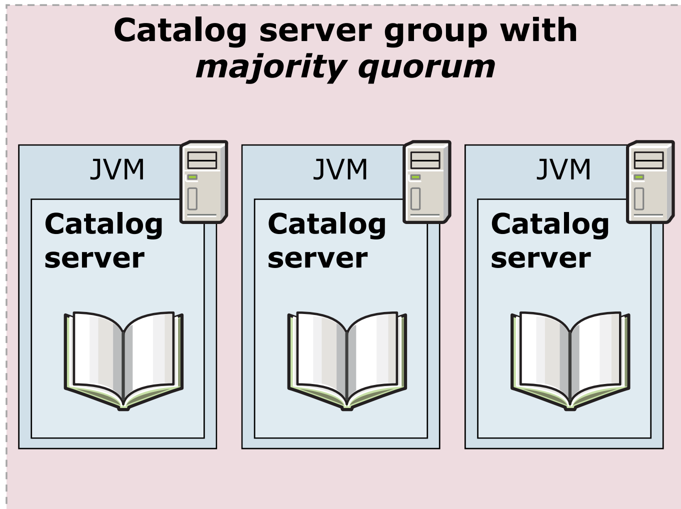
To meet the goals of high and continuous availability, the grid must be able to support loss of a container without losing data or access to data.

Grid data must be recoverable if you have a disaster where more than two containers are lost, such as during a power outage. If you need a controlled shutdown of the grid, such as when you increase the number of partitions, the data must be recoverable.

IBM Training

Catalog service

- For high availability: 3 catalog servers on 3 separate hosts



Configuring Insight Server

© Copyright IBM Corporation 2016

Figure 9-13. Catalog service

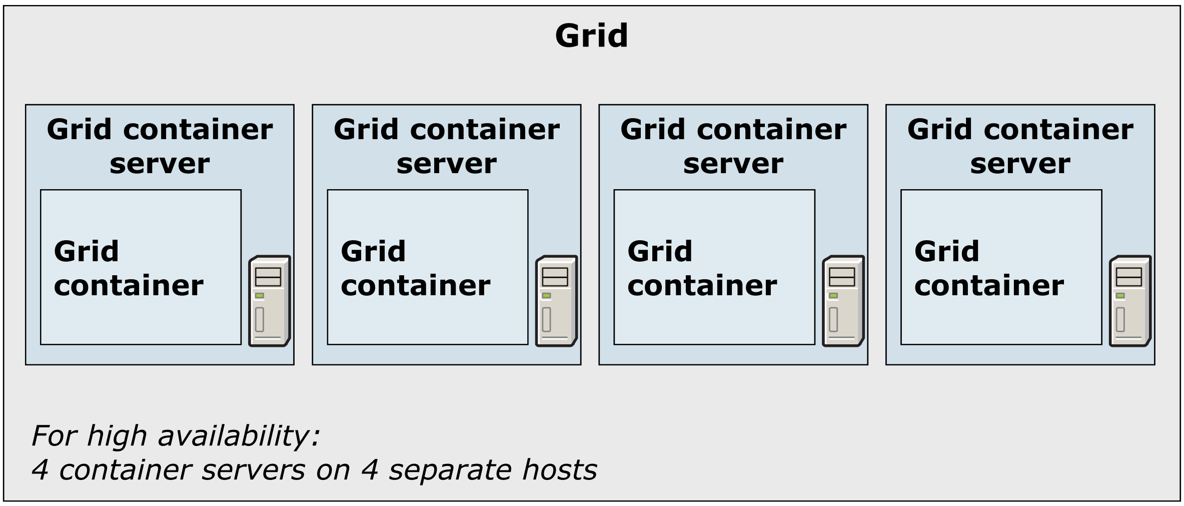
The reference topology suggests 3 catalog servers on 3 separate hosts.

In a normally functioning grid, catalog server usually has little work to do, and so it has little influence on scalability.

IBM Training

Containers and partitions (1 of 2)

- For high availability, use a minimum of 3 container servers
  - Use 4 container servers to allow for one server to be shut down for upgrade or maintenance
- Each container should run on a single host
- Each host should have a minimum of two cores



Configuring Insight Server

Figure 9-14. Containers and partitions (1 of 2)

The number of required container servers depends on the number and type of solutions that are running on the grid and the volume of events that they process. This number can grow or shrink.

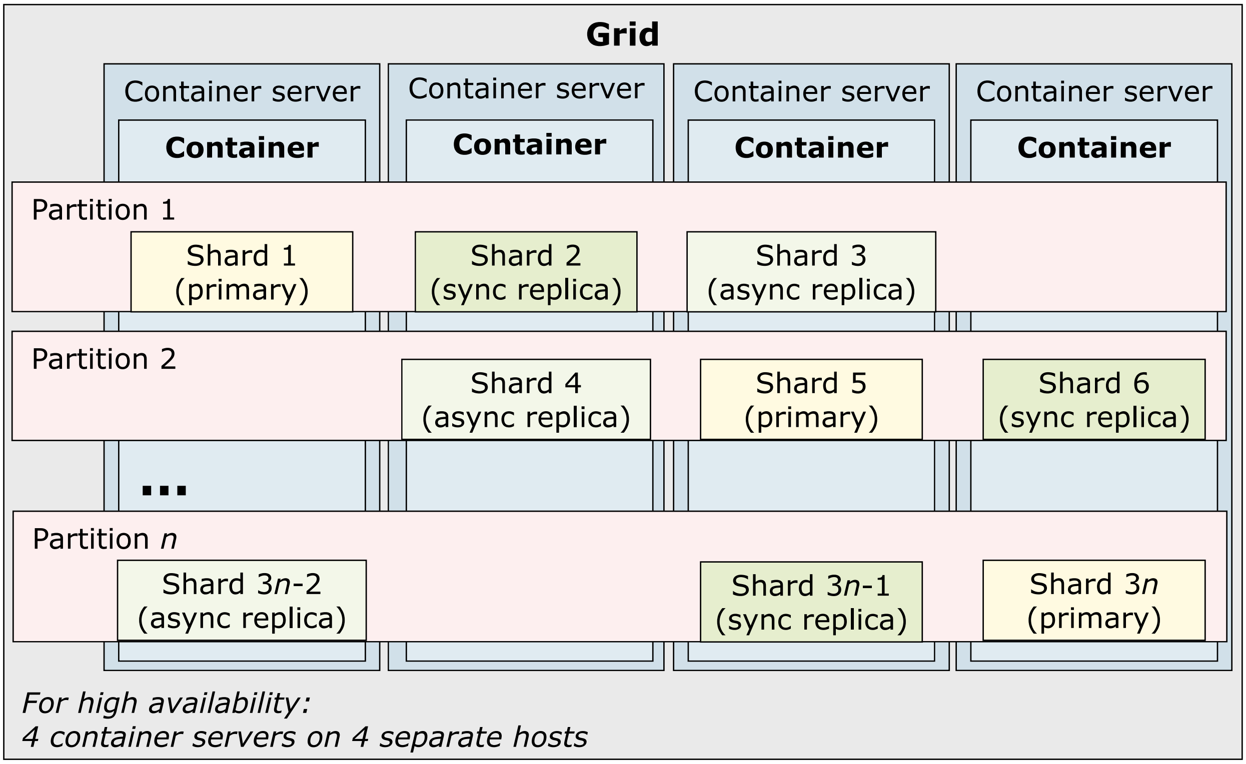
From the topology goals, you can infer that, in addition to the primary data, you should have a synchronous replica and an asynchronous replica. If the primary data is lost, the synchronous replica immediately becomes the primary, and the asynchronous replica becomes synchronous. This configuration meets high availability goals and ensures that maintenance can be performed on one container at a time. A suggested practice is to use four containers that run on four separate hosts, regardless of throughput.

The maximum number of containers that you can dynamically add to the grid is constrained by the number of partitions and the number of replicas that you configure for each partition. The number of partitions is configured when you set up the grid and can be changed only when the grid is down. You do not need more container servers than the maximum number that can be used to allocate partitions.

Having many partitions can facilitate growth and better balancing of memory resources, but also means more threads, increased use of grid communication resources, and greater level of balancing work. When you have many partitions but few containers, adding or removing containers involves a significant amount of grid reconfiguration work.

IBM Training

Containers and partitions (2 of 2)



Configuring Insight Server © Copyright IBM Corporation 2016

Figure 9-15. Containers and partitions (2 of 2)

The minimum suggested number of cores is two per server to enable faster start, stop, and management of external operating system (OS) requests without interrupting the workload.

Because the suggested practice is to use one container server per machine, you can determine the number of partitions to use with this equation: C x M x 2

Where:

- C is the maximum number of containers that you expect to have
- M is the number of cores of the container host

This equation assumes that you have an equal number of cores in each container host machine.

Certain eXtreme Scale operations are more efficient if the number of partitions is a prime number, so a good practice is to round the number of partitions up to the next prime number.

Example:

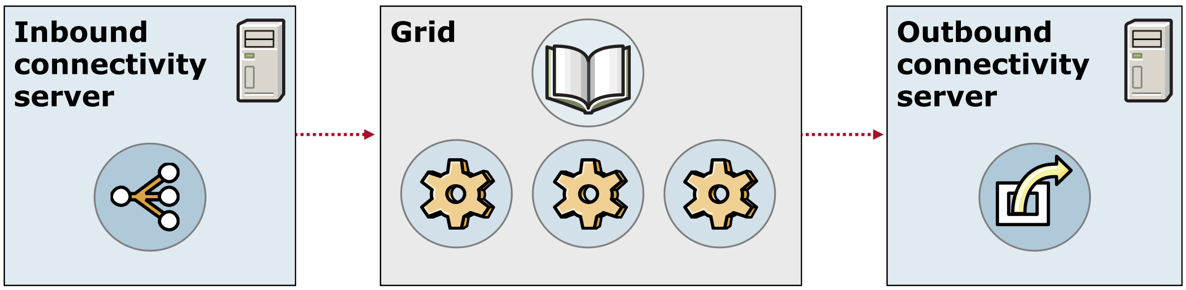
- Assume that you have a maximum of 6 container server hosts (C = 6), each with 4 cores (M = 4).
- Number of partitions = C x M x 2 = 6 x 4 x 2 = 48
- Round the result up to the next prime number: 53

The default product configuration for the number of partitions is 127.

IBM Training

Inbound and outbound connectivity

- Minimum configuration:
  - One inbound server on one physical host
  - One outbound server on a separate physical host



- For high availability, use two nodes each for inbound and outbound connectivity
  - You can reduce to two nodes, each with an inbound and outbound connectivity server
  - To scale vertically, add more servers to the same host
  - To scale horizontally, add more hosts

Configuring Insight Server

Figure 9-16. Inbound and outbound connectivity

Connectivity servers manage event flow to and from the system. Redundancy is required to avoid a single point of failure.

Inbound and outbound connectivity servers are responsible for the event flow in and out of the system, so there should be some redundancy for these servers. A good starting point is to use one of each on two different machines, which is the minimum requirement to avoid a single point of failure.

No rule exists for determining the actual number of inbound or outbound servers that are required for a particular event throughput. The number of inbound and outbound servers varies according to several parameters, including:

- Size of the event data
- Type of event transformations that might be performed
- Whether event persistence is used for Messaging Service (JMS)
- Type of protocol used (for example, HTTP or JMS)

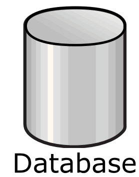
For best results, start with the two servers and monitor resource usage (for example, CPU and memory).

For high-event rates, you can add inbound and outbound connectivity servers to the same machine until the machine's ideal resource consumption threshold is reached. After the threshold is reached, you can add new machines if required.

IBM Training

Data persistence

- In a single site topology, persist grid data to a database
  - Enable recovery from a disaster
  - Enable any rare maintenance activities that require the grid to be shut down



- Synchronous mode (write-through)
  - High performance penalty
  - Ensures no data loss
- Asynchronous mode (write-behind)
  - Performs database updates in batches
  - Risk of some data loss
- Reference topology uses asynchronous persistence

Configuring Insight Server

Figure 9-17. Data persistence

Decision Server Insights operates on an in-memory data grid. However, persistence of grid data is required to enable disaster recovery or maintenance shutdown of the grid.

You can choose between synchronous (write-through) or asynchronous (write-behind) database persistence modes. Synchronous persistence typically implies a high performance penalty. Asynchronous persistence performs the database updates in batches and carries a small performance penalty. The frequency of database updates might vary depending on the level of accepted risk. It is set in terms of a time interval and the number of batched updates that use a parameter that is called `writeBehind`. For example, if you set the `writeBehind` value to `T20;C200` then a write to the database happens every 20 seconds or every time the number of pending updates reaches 200, whichever condition happens first.

The reference topology uses write-behind persistence mode to avoid the performance cost and throughput penalty of the synchronous persistence option.

WebSphere eXtreme Scale can queue the database updates, allowing for some database downtime, but ideally the database should be highly available (for example, by using DB2 high availability disaster recovery). For high event loads, consider using a highly scalable database to ensure that the database does not become a bottleneck.

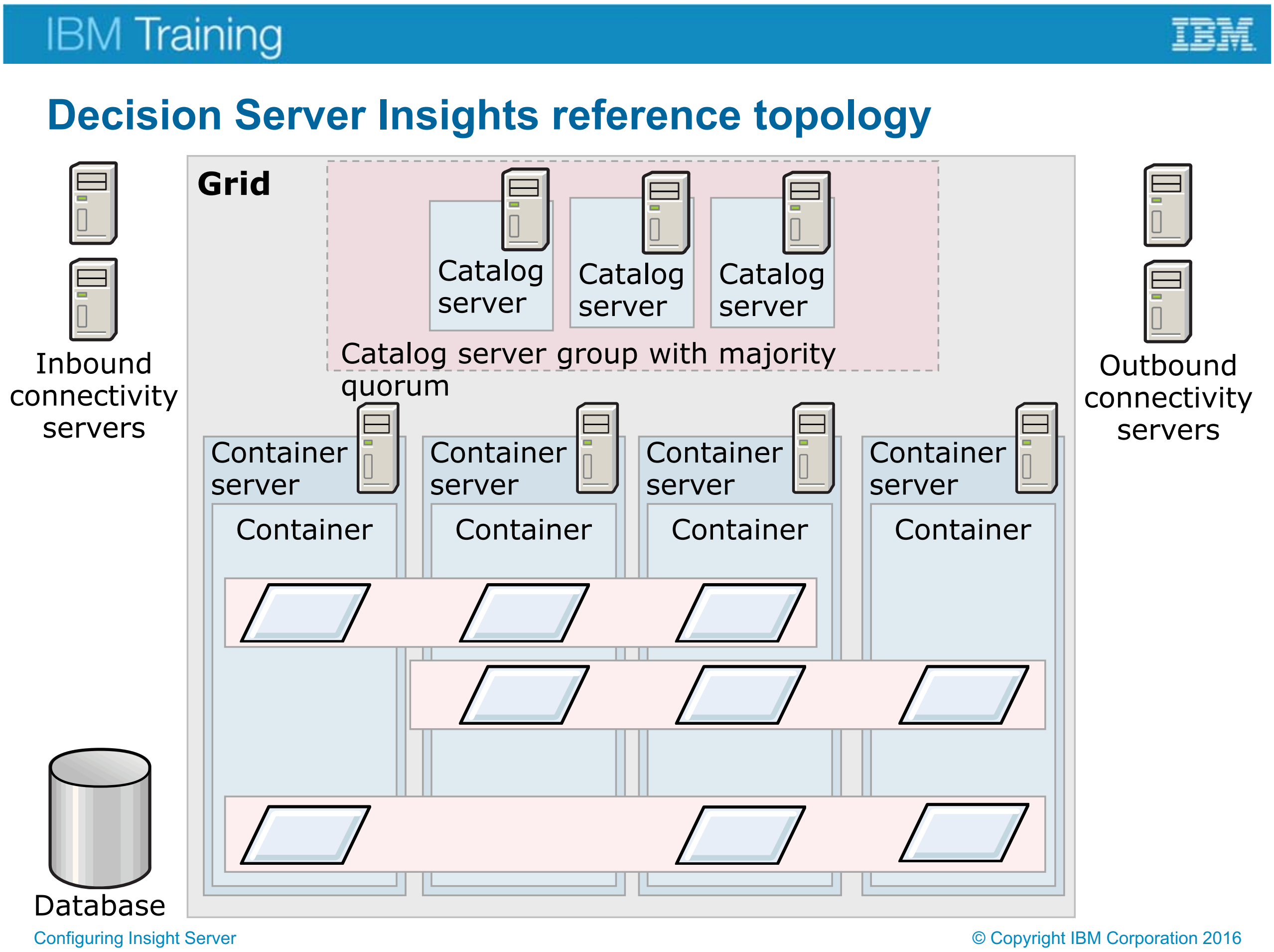


Figure 9-18. Decision Server Insights reference topology

The reference topology uses a different physical server for each Decision Server Insights server in the topology. This configuration provides a high degree of isolation between components.

The high availability and recoverability objectives can be met with a minimum of three physical servers, not counting the database. However, a topology with four servers would be more appropriate as a minimal high availability topology. The reference topology uses four container servers, one on each host machine. Three machines can have a catalog server. Additionally, you should have inbound and outbound servers on at least 2 of the 4 physical servers.

9.3. Sizing

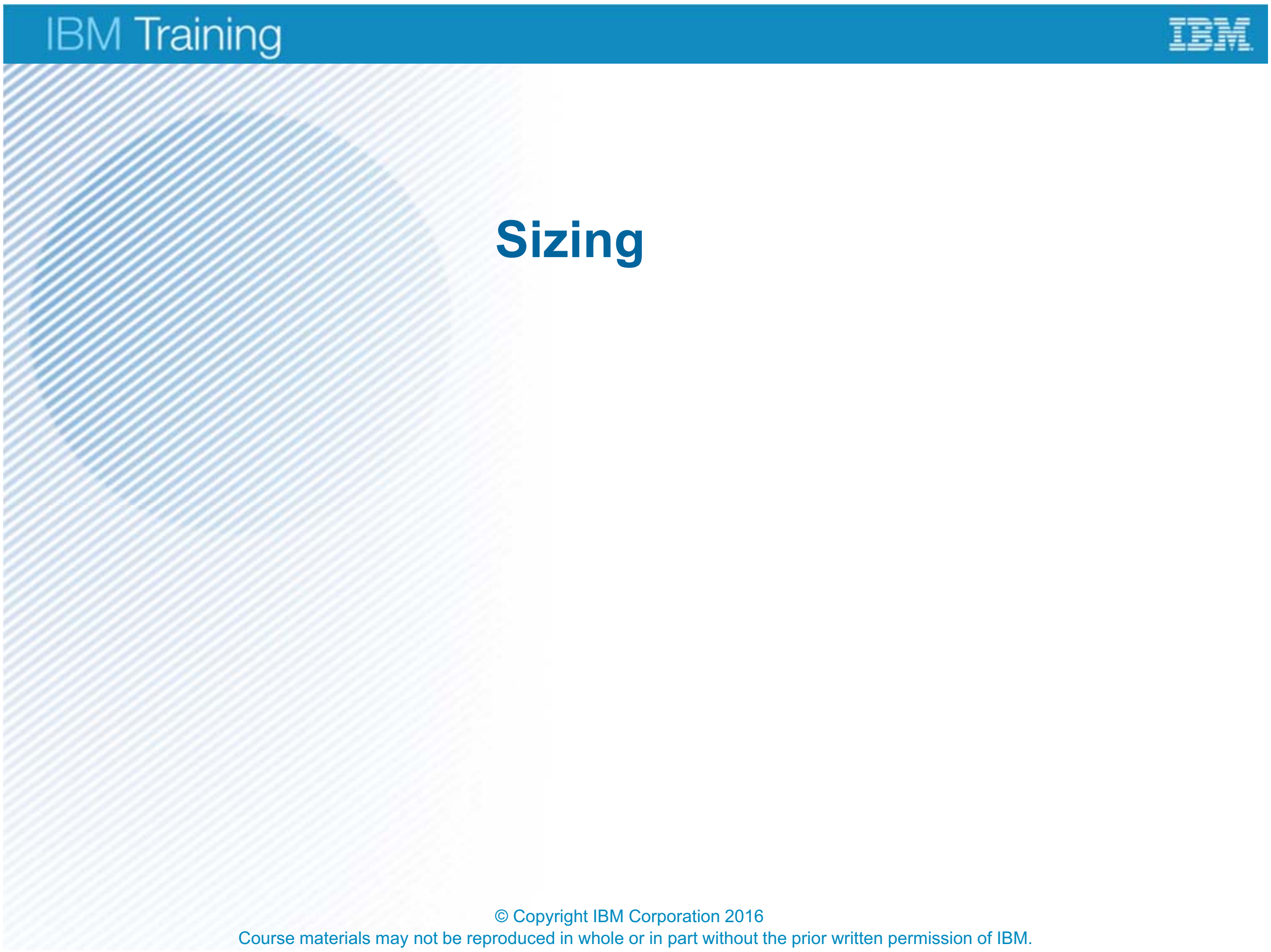


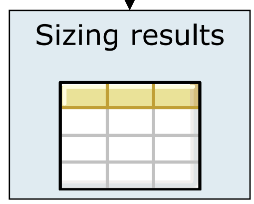
Figure 9-19. Sizing

IBM Training

Capacity planning for containers in the grid

- Steps to estimate the hardware, JVM, and grid configuration

1. Collect the information about the data (Java objects) to be stored
2. Determine the memory requirements and partition count
3. Determine the CPU size and server count



- Sizing results include:
  – Number of servers (physical machines)
  – Number of CPUs
  – Total physical memory
  – Number of containers
  – Maximum heap size per JVM (Xmx)
  – Number of partitions

- JVM recommendation for each container: 60% heap utilization

Configuring Insight Server

Figure 9-20. Capacity planning for containers in the grid

To compute the container memory requirements, catalog servers should not share the host with containers.

IBM Training

Calculating the required memory for the data in the grid

- To determine the maximum size of the data to be stored in the grid
  - Estimate the maximum number of objects that are needed at usage peaks (*numberOfObjects*)
  - Estimate the average size of each object (*averageObjectSize*)
- Use values for total object data to be stored, event data, and number and type of replicas to determine physical requirements

Sizing variable	Calculation
Total object data to be stored	averageObjectSize * numberOfObjects * (1 + number of replicas)
Maximum memory per container	maxHeapSizePerJVM * maxHeapUsage – ExtremeScale footprint
Minimum number of containers	totalObjectMemory * containerObjectMemory
Total physical memory for grid	physicalMemoryPerJVM * numberOfContainers
Container object memory	Total object memory / Maximum number of containers

Configuring Insight Server

Figure 9-21. Calculating the required memory for the data in the grid

To compute container object memory, you must know how much memory the object data (primary and replica) uses in each container JVM.

Make sure that your estimate is based on real data.

IBM Training

Determining number of partitions

- Set the number of partitions (P) to a prime number
  - Maximum number of containers (C)
  - Number of cores per container host (M)

Number of partitions = (C * M * 2) rounded up to next prime number

- You must limit the number of containers that are running to match the number of partitions
  - If number of partitions is P, you cannot use more than P containers to host data
  - Additional containers start but do not receive any partition to host

- One container per host of physical server

- Consider the number of cores on the host machine
  - With more cores, increase the number of partitions to increase the work load on each machine

Configuring Insight Server

Figure 9-22. Determining number of partitions

If feasible, use one container per host of physical server so that failure of a machine does not bring down more than one container. However, it is common to have multiple containers per server in cases where there are at least 3 or more servers.

IBM Training

Example: Calculating memory storage

- Assumptions for the solution:
  - Number of primary objects to be stored: 18
  - Average size of primary objects: ~10K per object
  - Total size of PrimaryObjectMemory: 18 GB
  - Number of required replicas: 1
  - Physical host machines: 4
  - Footprint for WebSphere eXtreme Scale (WXS) + WebSphere Application Server: 100 MB
  - Maximum heap size per JVM = (Container object memory + WebSphere eXtreme Scale / WebSphere Application Server footprint) * 1.7 (60% utilization)

Step	Calculations	Data (GB)
1	Total object memory = primary object memory * 2 (primary data + 1 replica)	36
2	Container object memory: total object memory / number of containers (4 hosts)	9
3	Maximum heap size per JVM = (9 GB + 0.1 GB) * 1.7	15.47

Configuring Insight Server

Figure 9-23. Example: Calculating memory storage

For this example, the solution must store object data for 18 types of primary objects. This data includes event data that the customer wants to keep a history of. Other event data is deleted. The average size per object is 10 KB.

1. To calculate the total amount of memory that must be stored across all containers, you calculate the amount of memory that is required for the objects that are processed by your grid. You then multiply that value by the number of replicas that you require.
2. The amount of memory that you need to store in each container is calculated by dividing the total object memory by the number of containers that you can use. In this example, the organization allotted a maximum of 4 physical hosts, and the plan is to use one container per host.
3. The maximum memory that can be used per container is limited by the JVM heap size. The container memory should not exceed 60% usage of the heap. To ensure that you remain under 60%, you can multiply the total content memory value by 1.7.

   Based on the calculations for this example, by using 4 containers, one per physical host, the maximum heap size is almost 16 GB, which is too large for a 64-bit operating system.

   To accommodate this memory requirement, more JVMs are needed to reduce the load per JVM.

As a general guideline, keep the maximum heap size for each JVM to under 5 - 6 GB.

IBM Training

Example: Determining heap size

- Goal: Maximum heap size should be 5-6 GB per JVM recommendation
  - Increase the number of containers from 4 to 11
  - All containers to run on the 4 physical hosts

Step	Calculations	Data (GB)
4	Maximum heap size per JVM • containerObjectMemory = 36 GB / number of containers (11) = 3.2727 GB • maxHeapSizePerJVM = (3.2727 + WebSphere eXtreme Scale / WebSphere Application Server footprint (.1 GB)) * 1.7 = **5.7336** GB • **Result** = Next highest multiple of 256 MB	6.144
5	Operating system (OS) tax • For a 64-bit Linux platform: 0.5 GB • Other native methods or tools that run on the OS: ~0.5	1
6	Physical Memory per JVM: OS tax + maximum heap size = **7.144**; round up to 7.5	7.5
7	How many JVM containers on each server? • maxNumberJVMsPerServer = 32 GB available memory per server / 7.5 GB • physicalMemoryPerJVM = 4 JVMs per server • numberOfServers = 12 JVMs / 4 JVMs per server	4

Configuring Insight Server © Copyright IBM Corporation 2016

Figure 9-24. Example: Determining heap size

MaxHeapUsage is 60% heap utilization. To represent the 60% heap utilization factor, this calculation uses a 1.7 multiplier.

1. To keep the heap size within the suggested range of 5 - 6 GB, the topology would require 11 containers as a minimum.

   The number of physical hosts is still 4, but multiple containers can run on a single host.

   You divide the total object memory by 11 containers. The result is the amount of memory to be stored on each container (`containerObjectMemory`), which is then added to the footprint and multiplied by 1.7 to determine maximum heap size.

   The final result can then be rounded up to the next highest multiple of 256 MB. In this example, the next highest multiple is 6.144 GB (which is 256 * 24).

2. In addition to the calculation for the grid object memory, you must also account for memory usage by the operating system (OS) and other native methods or tools. Use 1 GB as the OS tax value.

3. The physical memory per JVM is calculated by adding the OS tax to the maximum heap size.

4. To determine how many containers to run per server, you divide the available memory per server by the physical JVM memory. The result shows that you can host 4 container JVMs on a single host.

   To allow for an even distribution of 4 containers per host across the physical hosts, the number of containers is rounded up to 12. So only 3 physical hosts are required to store the total object memory.

IBM Training

Example: Planning for failover

- Consider failure and maintenance shutdown
  - Physical memory per JVM requires 4 JVM per server
  - Total containers: 11, rounded up to 12 for even distribution of the 4 JVMs
  - Physical hosts that are required: 3
- To plan for failure or maintenance, add one host that also runs four containers

Step	Calculations	Data (GB)
8	Failure scenario or maintenance shutdown: Loss of 1 JVM • 12 – 1 = 11 • 11 * containerObjectMemory (3.2727 GB)	35.9997
9	Recommendation to ensure performance: Add 1 host to run 4 additional containers • Total physical hosts: 4 • Total containers: 16 • Use previously calculated maxHeapSizePerJVM and physicalMemoryPerJVM values, so no change in physical memory per server	

Configuring Insight Server

Figure 9-25. Example: Planning for failover

1. The failure and maintenance scenarios account for the failure of a single JVM. In this case, because there are 12 JVMs, losing one JVM means that the 11 remaining containers must be able to hold the object memory. Each container is expected to hold 3.2727 GB. The maintenance scenario accounts for shutdown of one server at a time, if required, without affecting the grid.
2. To ensure grid performance, the recommendation is to increase the number of physical hosts from 3 to 4. Each host can run 4 container servers. The grid would now have 16 container JVMs, which continue to use the previously calculated heap size and physical memory values. The amount of physical memory per server is unchanged.

IBM Training

Example: Number of partitions

- Number of partitions = C * M * 2
  - C: Maximum number of containers
  - M: Number of cores on the container host
  - To account for growth: Multiply result by 2
  - Final result is rounded up to next prime number

Calculations	Data (GB)
C	16
M	2
(C * M * 2) * 2	64
Number of partitions (result rounded to next prime number)	**67**

Configuring Insight Server

Figure 9-26. Example: Number of partitions

To calculate the partitions, you multiply (the maximum number of containers) by (the number of cores per container host), and double the result. The final result is rounded to the next prime number.

IBM Training

Example: Sizing results

Results	Extended growth: 7+ years
Number of servers (physical machines)	4
Number of CPUs	4 * 2 = 8 cores
Total physical memory	32 GB x 4 = 128 GB
Number of WebSphere eXtreme Scale container JVMs	16
MaxHeapSize per JVM, also known as Xmx	6144 MB or 6.144 GB
numberOfPartitions	67

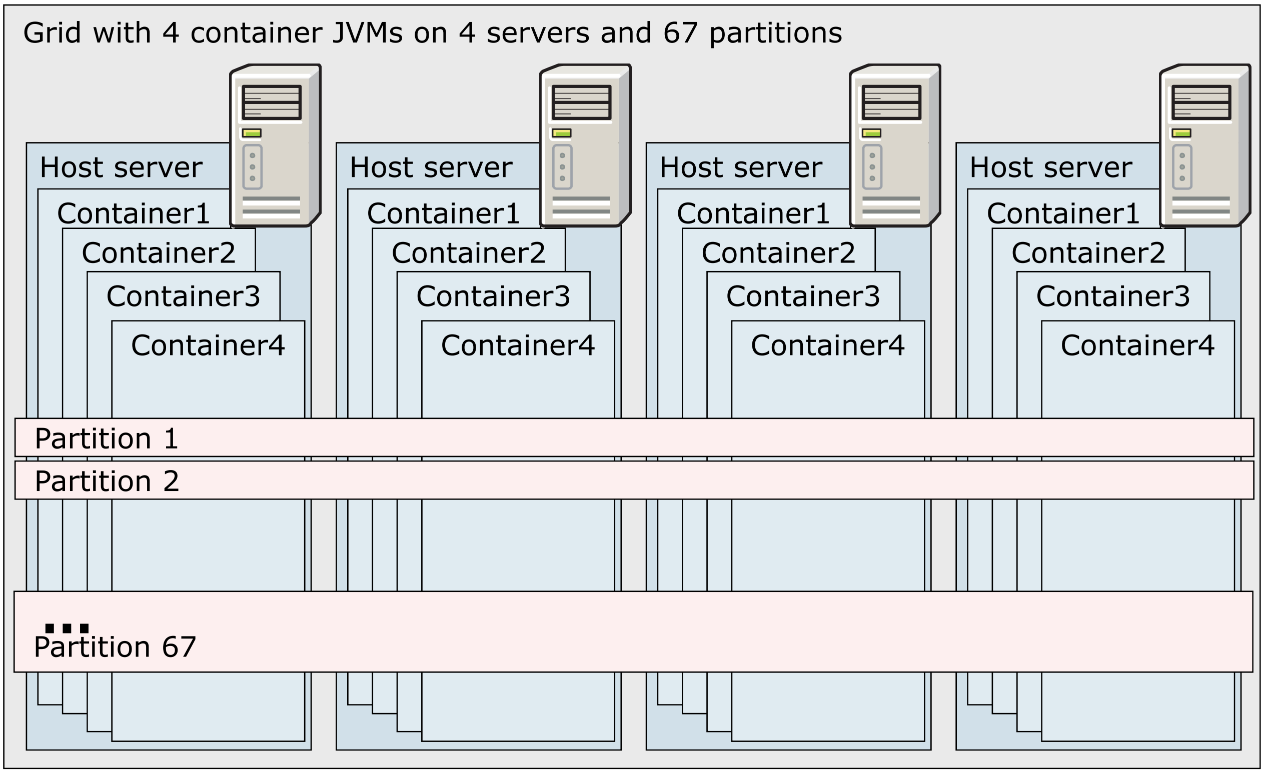
Configuring Insight Server

Figure 9-27. Example: Sizing results

This slide outlines the final result for the sizing example. These results account for growth over the next 7 years.

IBM Training

Example: Resulting grid



Configuring Insight Server

Figure 9-28. Example: Resulting grid

The final sizing results for the example are depicted on this slide:

- 4 container hosts
- 4 servers on each host
- 67 partitions

9.4. Configuring a production topology

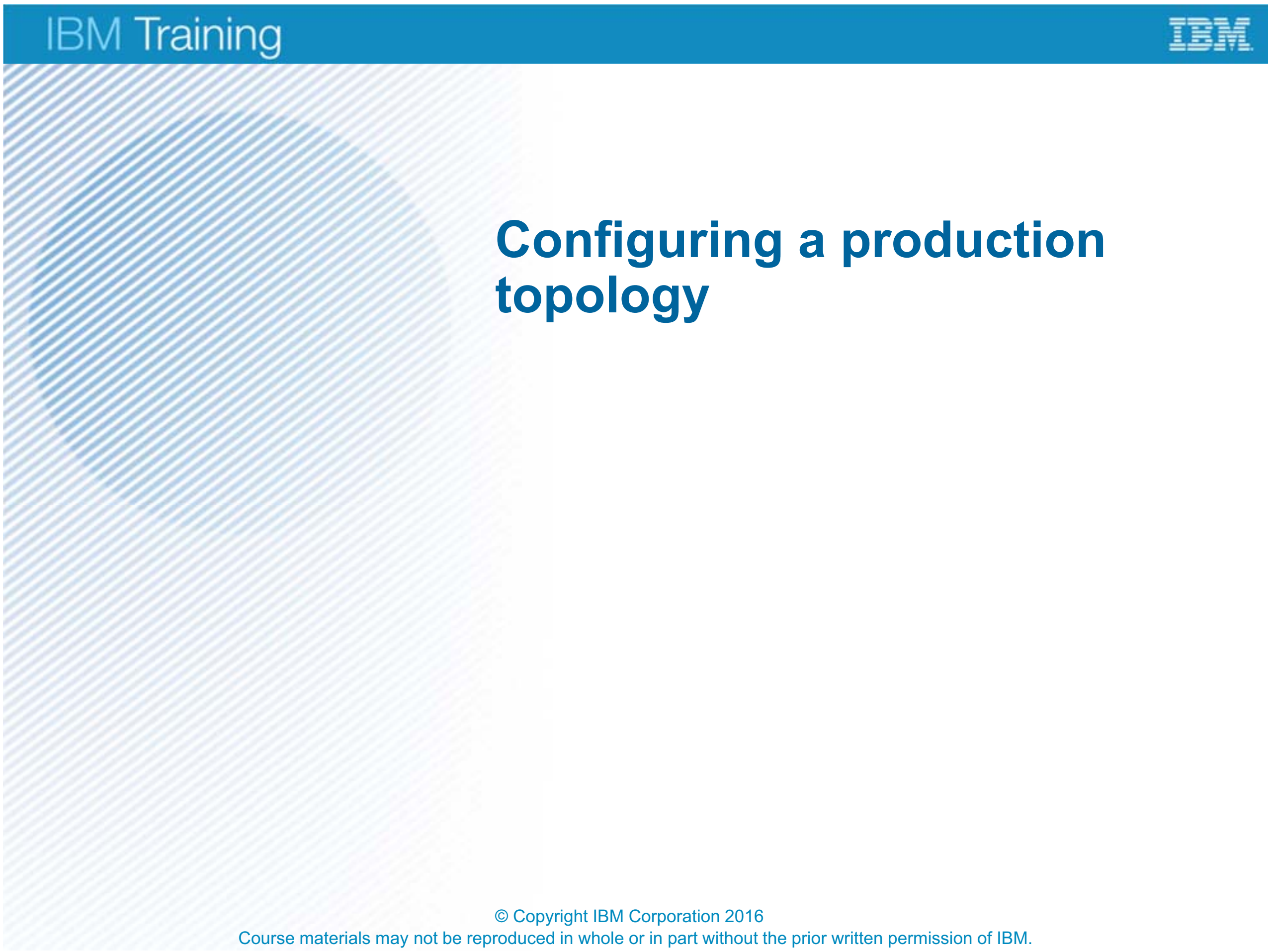


Figure 9-29. Configuring a production topology

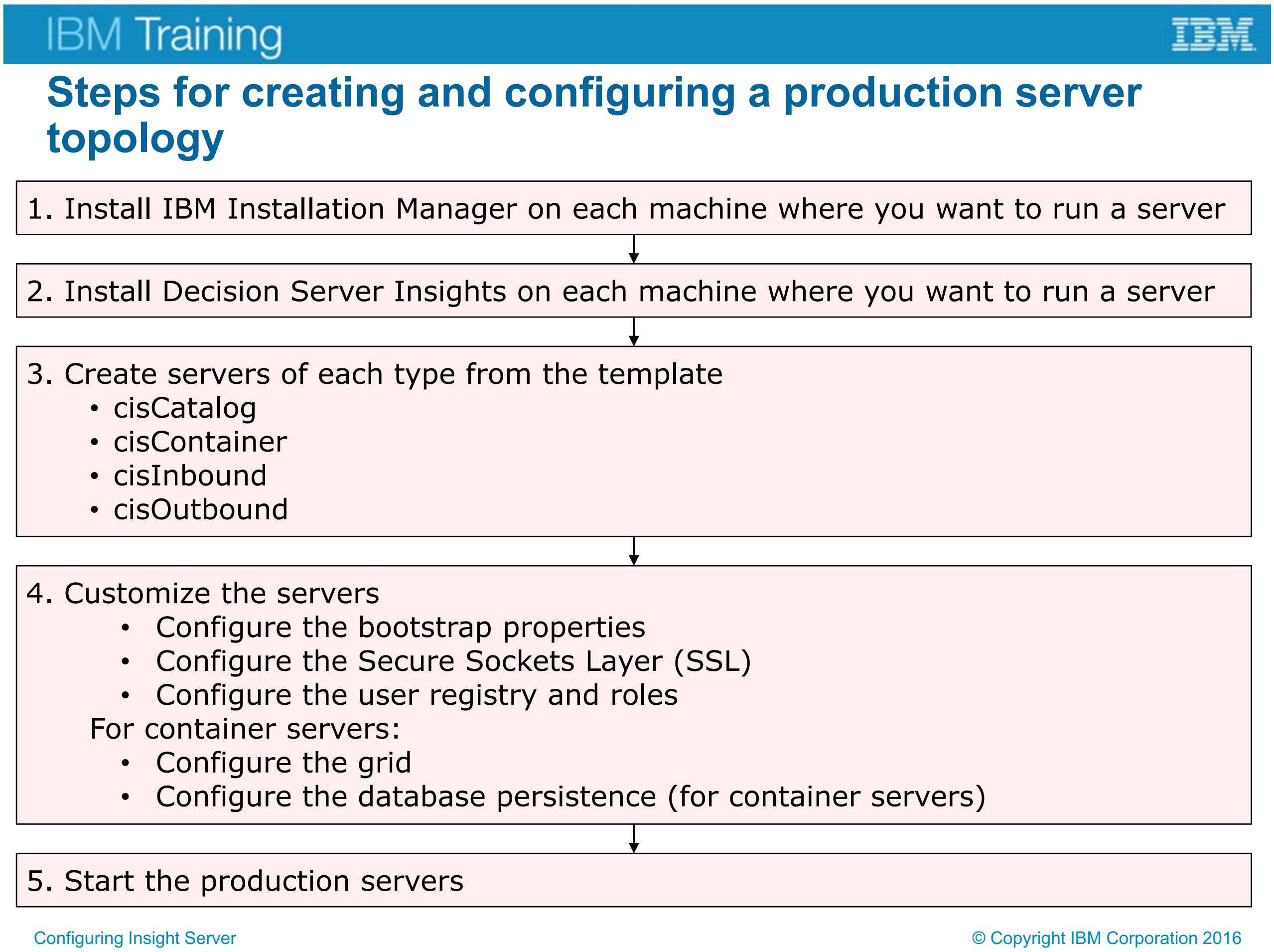


Figure 9-30. Steps for creating and configuring a production server topology

This slide outlines the steps for creating and configuring a production topology.

IBM Training

Server templates

- Create servers with Decision Server Insights server templates

Server templates	Description
Catalog `cisCatalog`	• Required for high availability • A production topology contains a minimum of three catalog servers • Catalog servers and container servers cannot coexist on the same host
Container `cisContainer`	• Stores entities and system information, and runs agents and analytics jobs • A production topology contains a minimum of three container servers • Container servers and catalog servers cannot coexist on the same host
Inbound `cisInbound`	• Submits inbound events to the grid either through the gateway API, through HTTP, or through JMS • A production topology contains at least one inbound server
Outbound `cisOutbound`	• Emits outbound events through HTTP or through JMS • A production topology contains at least one outbound server

Configuring Insight Server

Figure 9-31. Server templates

Decision Server Insights provides templates for you to generate the various server types that you use to set up your topology.

After you create a server from the template, you customize it by configuring the bootstrap properties, security, and JVM options.

IBM Training

Customizing servers

- For each server that you create, you edit these files:
  - `bootstrap.properties`
  - `server.xml`
- For catalog servers, edit `bootstrap.properties` to configure `ia.catalogClusterEndpoints`
  - Declares the host and server names, and peer connection ports of all catalog servers in the topology
- For container servers, inbound servers, and outbound servers, edit `bootstrap.properties` to configure `ia.bootstrapEndpoints`
  - Declares the host names and client listener ports of the catalog servers in your topology

Configuring Insight Server

Figure 9-32. Customizing servers

For catalog servers, edit `bootstrap.properties` to configure `ia.catalogClusterEndpoints`. This property declares the host and server names, and peer connection ports of all catalog servers in the topology.

Set this property by using the following format:

```
computer_name-server_name:computer_name:port_number:port_number,
computer_name-server_name:computer_name:port_number:port_number
```

For container servers, inbound servers, and outbound servers, edit `bootstrap.properties` to configure `ia.bootstrapEndpoints`. This property declares the host names and client listener ports of the catalog servers in your topology.

Set this property by using the following format:

```
computer_name:port_number,computer_name:port_number
```

IBM Training

Configuring security (1 of 2)

- To configure SSL, security and roles, you edit the `server.xml` file for each of the servers
- Generate a keystore (`key.jks`) by using the securityUtility
  - It is not necessary for all of the servers to use the same keystore
  - The keystores contain the certificates that are required for establishing trust
  - Example:
    `securityUtility createSSLCertificate --server=cisCatalog1 --password=ins1ghts`
- You must also provide a user registry configuration that defines which users are authorized to access the server
  - The user registry can also authorize the administrator to use the JMX and REST APIs, if necessary

Configuring Insight Server

Figure 9-33. Configuring security (1 of 2)

To configure security, you first generate a keystore, and then you edit the `server.xml` file for each server to provide authentication credentials.

IBM Training

Configuring security (2 of 2)

- Provide a basic or LDAP user registry configuration
  - Defines which users are authorized to access the server
  - Example:

```
<basicRegistry id="basic" realm="SimpleRealm">
   <user name="SimpleAdmin" password="abcdefg"/>
   <group name="SimpleAdministratorsGroup">
      <member name="SimpleAdmin"/>
   </group>
</basicRegistry>
```

- Configure authorization roles for server administration
  - Example:

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

Configuring Insight Server

Figure 9-34. Configuring security (2 of 2)

This slide shows and example of a basic or LDAP registry configuration and an example of an authorization role for server administration.

IBM Training

Configuring heap size for container servers

- Customize the heap size for container servers in the `jvm.options` file by setting the `-Xms` and `-Xmx` properties
  - Default `-Xms28g` and `-Xmx28g`
- Configure the grid in the `objectGridDeployment.xml` file, including these properties:
  - `numberOfPartitions`
  - `numInitialContainers`
  - `maxSyncReplicas`
  - `maxAsyncReplicas`

Configuring Insight Server

Figure 9-35. Configuring heap size for container servers

For container servers, you edit the `jvm.options` file to set the heap size.

You use the `objectGridDeployment.xml` file to define various grid properties.

IBM Training

IBM

Starting the servers

- Start all servers from the `runtime/wlp/bin/server` directory
  - `server start <serverName>`
- Startup sequence:
  - Catalogs
  - Containers
  - Outbound connectivity
  - Inbound connectivity
- When using multiple catalog servers, at least two must be started concurrently

Configuring Insight Server

Figure 9-36. Starting the servers

To manage the servers, you use the `server` and the `serverManager` utilities.

- Use `server` to start your three catalog servers; all catalogs should be started concurrently
- Use `serverManager` to suspend balancing
- SSH into your four containers and start them by using the `server` command.
- Use `serverManager` to resume balancing
- Use `server` to start your outbound server
- Use `server` to start your inbound server

IBM Training

Unit summary

- Describe WebSphere eXtreme Scale basics
- Describe the Decision Server Insights reference topology
- Design and configure a production topology

Configuring Insight Server

Figure 9-37. Unit summary

IBM Training

Review questions

1. **True or False:** A Decision Server Insights reference topology requires a minimum of 9 servers.

2. **True or False:** To reduce hardware requirements, you can run inbound and outbound servers on the same physical machine.

3. **True or False:** To compute the number of partitions for containers, you should consider the name of container hosts * the number of cores per host.

Configuring Insight Server

Figure 9-38. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. **True**

2. **True**

3. **False.** *The number of partitions is calculated by the name of container hosts * the number of cores per host * 2. The result should be rounded up to the next prime number.*

Configuring Insight Server

Figure 9-39. Review answers

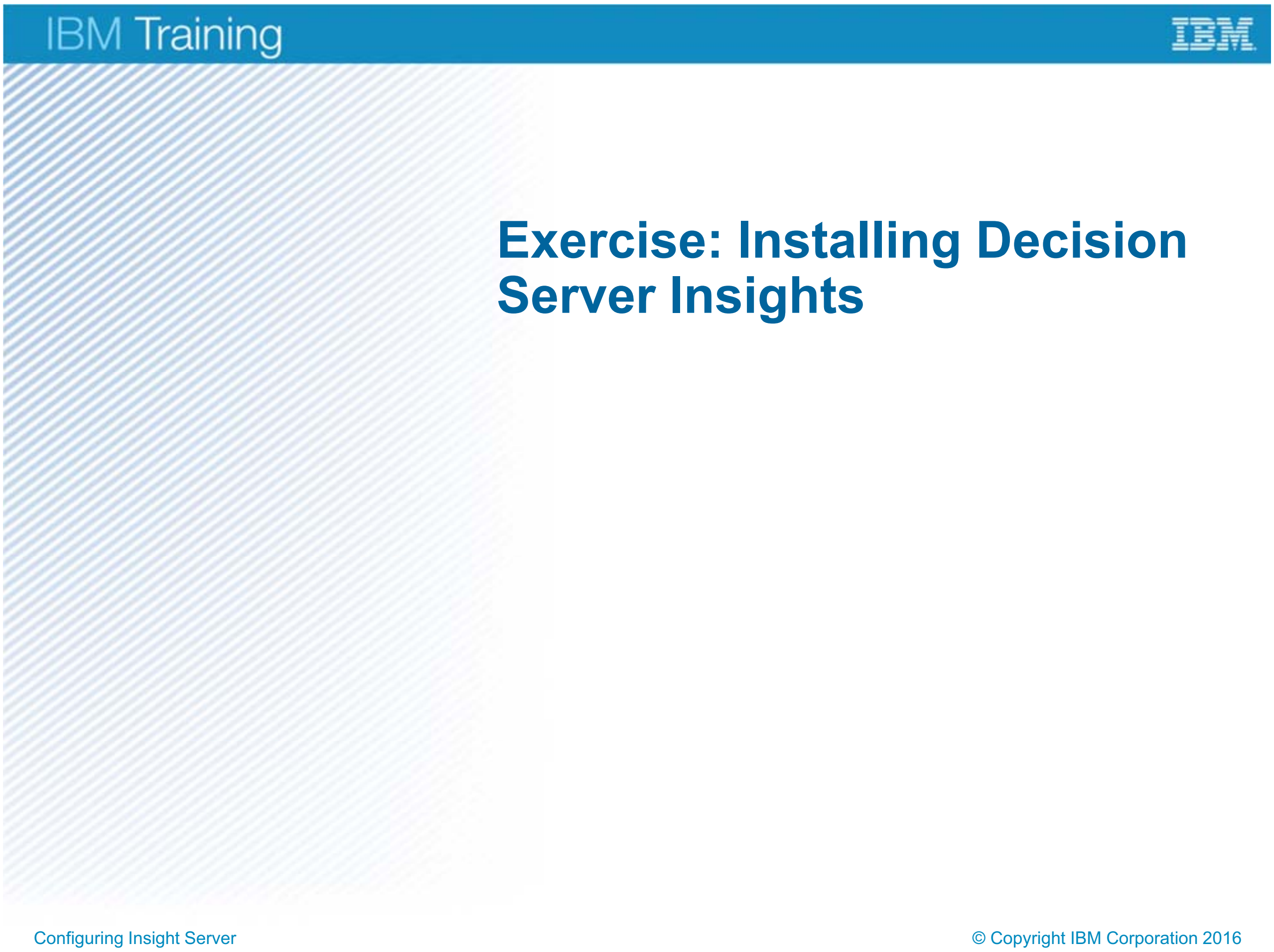


Figure 9-40. Exercise: Installing Decision Server Insights

IBM Training

IBM

Exercise objectives

- 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

Configuring Insight Server

Figure 9-41. Exercise introduction

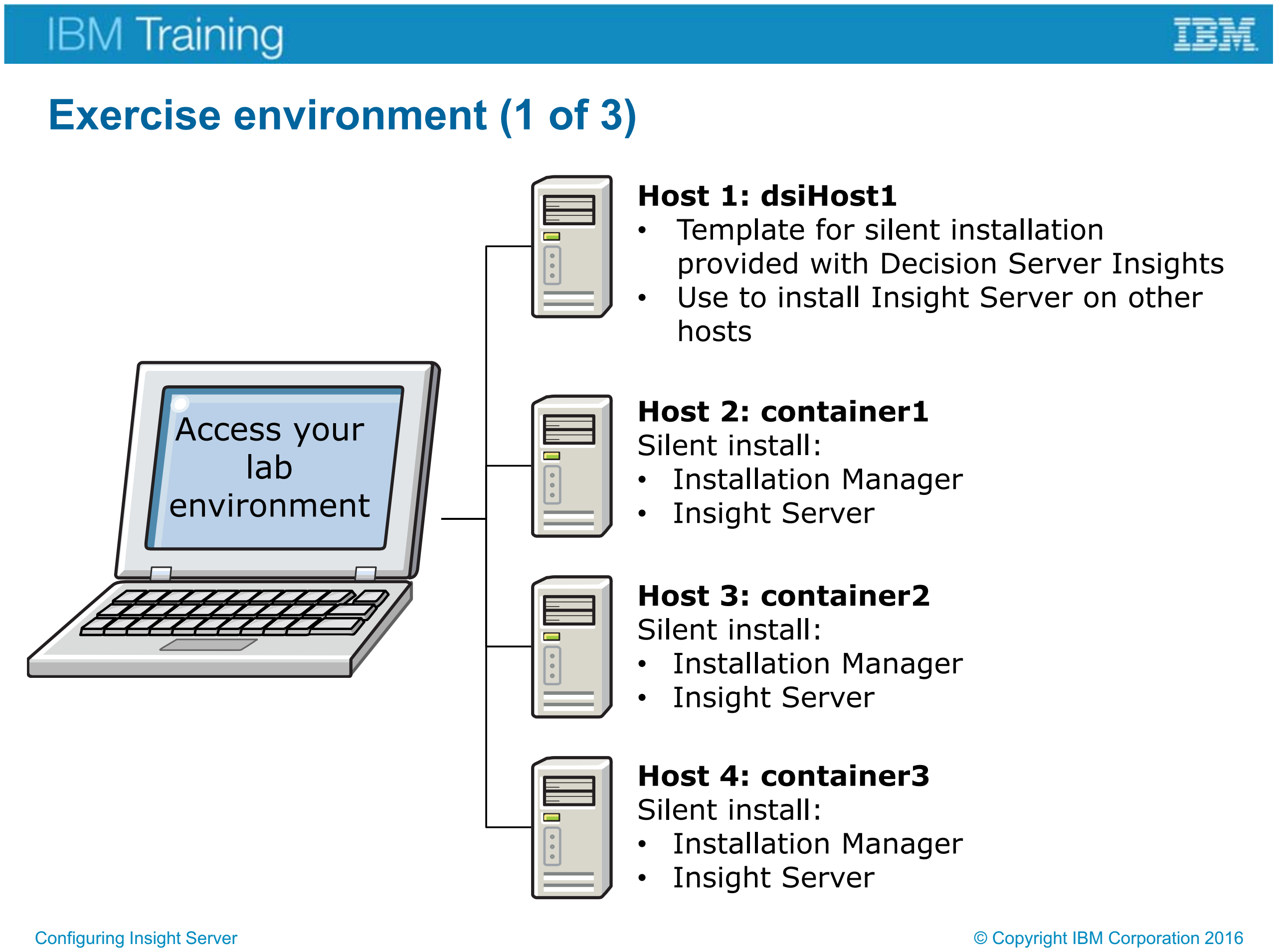


Figure 9-42. Exercise environment (1 of 3)

During the lab, you install Decision Server Insights on each of your container hosts.

Stop

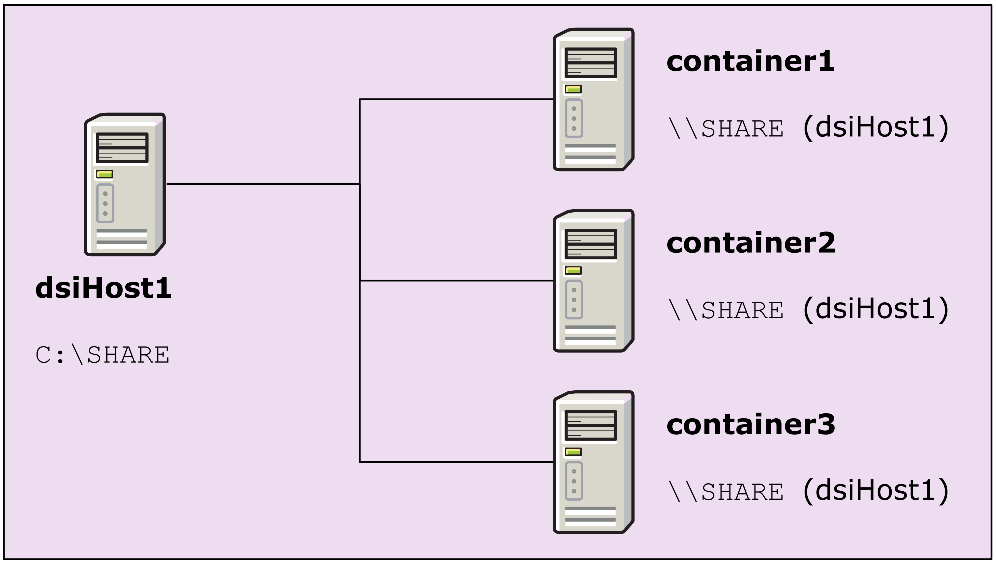
The default host names are: dsiHost1, container1, container2, and container3.

Your hosts might be assigned different unique host names. Make sure that you know and use the actual host names for your environment during the exercises.

You perform a silent installation of Installation Manager and Decision Server Insights (Insight Server only) on Host 2, Host 3, and Host 4, which are your container hosts.

IBM Training

Exercise environment (2 of 3)



Configuring Insight Server

Figure 9-43. Exercise environment (2 of 3)

To transfer files from one virtual machine to another, you use a shared directory.

On your main host (dsiHost1), the `C:\SHARE` folder is used to move files from the main host to the remote host. The other machines have a drive that is mapped to the `SHARE` folder.

IBM Training

Exercise environment (3 of 3)

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: ____________________

Configuring Insight Server

Figure 9-44. Exercise environment (3 of 3)

This graphic is in the final appendix of your exercise guide. Use that appendix as a reference to make a note of the IP addresses and host names that are assigned to your virtual machines.

IBM Training

Exercise: Configuring Decision Server Insights

Configuring Insight Server

Figure 9-45. Exercise: Configuring Decision Server Insights

IBM Training

Exercise objectives

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

Configuring Insight Server

Figure 9-46. Exercise introduction

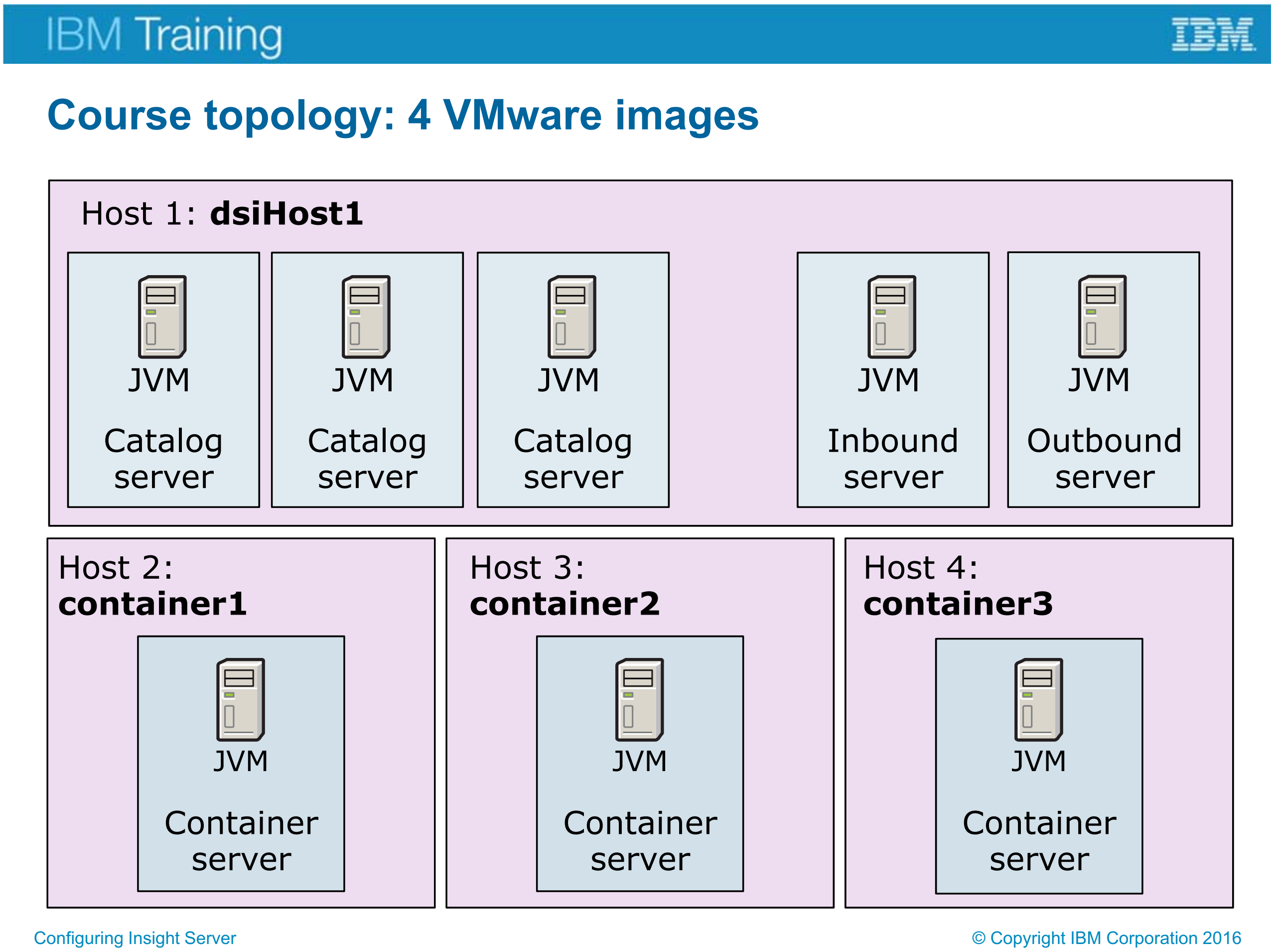


Figure 9-47. Course topology: 4 VMware images

For this lab, you use four VMware images.

- **dsiHost1**: Hosts the complete ODM Advanced installation, including Decision Server Insights. On this host, you create the catalog, inbound, and outbound servers. Each catalog server is configured to be aware of the other catalogs and the other server types in the grid.

  For this course, because these servers are all on the same host, you must modify the ports so the catalogs, inbound, and outbound servers are not all listening to the same ports.

- **container1, container2, and container3**: Each of these hosts is used as a container server.

Questions

Configuring the grid: How many partitions would this topology use, with each container running on a single core machine?

C * M * 2 = <next prime number>

Answer: 3 containers * 3 cores * 2 = 18, which must be rounded up to the next prime number: 19.

Stop

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

Your hosts might be assigned different unique host names. Make sure that you know and use the actual host names for your environment during the exercises.

Unit 10. Managing deployment

Estimated time

01:00

Overview

This unit explains how to deploy to Insight Server.

How you will check your progress

- Review
- Exercise

IBM Training

Unit objectives

- Explain how to export and deploy solutions
- Describe how to manage solutions with the solutionManager script
- Manage deployment to multiple hosts

Managing deployment

Figure 10-1. Unit objectives

IBM Training

Topics

- Exporting and deploying

Managing deployment

Figure 10-2. Topics

10.1. Exporting and deploying

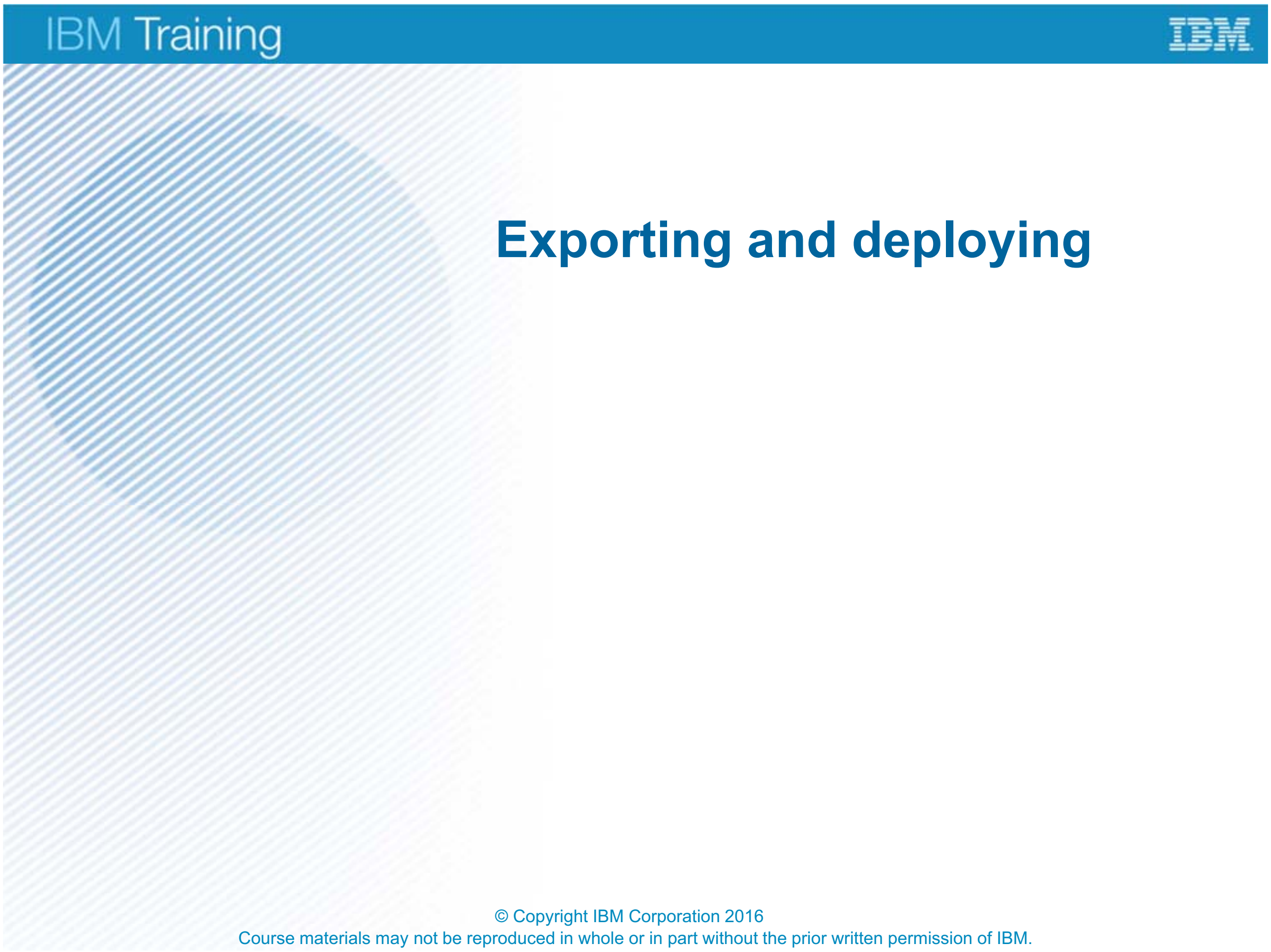


Figure 10-3. Exporting and deploying

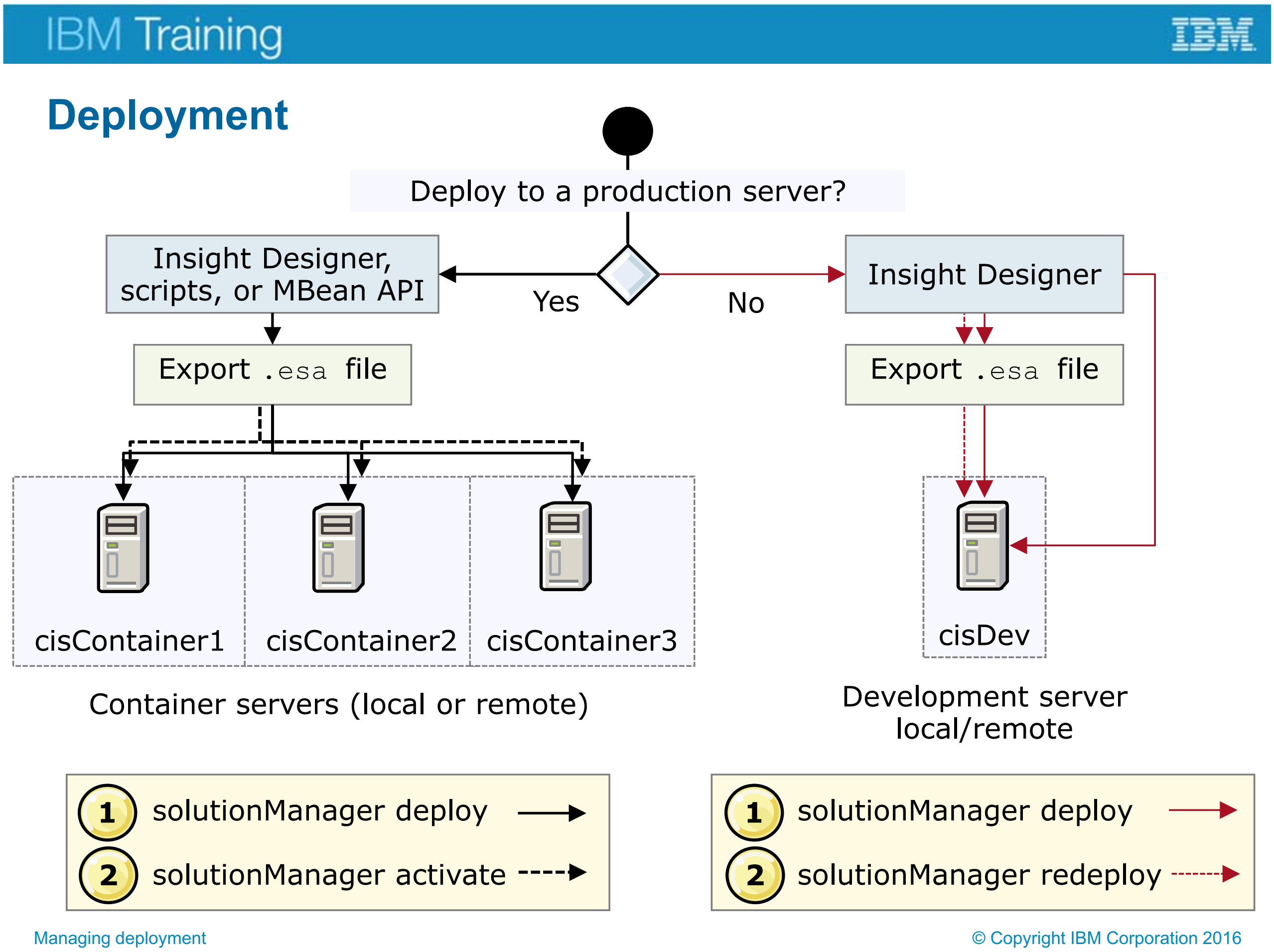


Figure 10-4. Deployment

You can deploy a solution from Insight Designer or by using scripts.

Deployment includes exporting the solution to an archive (`.esa`), which can be stored and managed by administrators.

Decision Server Insights provides several ways to deploy solutions to both local and remote servers.

In the development environment, you can use Insight Designer to deploy to the development server, cisDev, either locally or remotely. If you have a test or development environment that is set up on multiple hosts to match your production environment, you can test both local and remote deployment.

To deploy to multiple container servers on remote hosts, you can use scripts to automate deployment. The deployment scripts can automate the export of the solution to an archive, deployment of the archive to the servers, and activation of the solution on the servers.

IBM Training

Exporting solutions and agents

- When you export a full solution or a single agent, Insight Designer creates an `.esa` archive
- Solution and agent archives are OSGi subsystem archives that contain OSGi bundles
  - Deployable to Insight Server
- Exporting solutions
  - By default, the archive uses the solution name plus version as the archive name
  - All agents that are referenced in the solution project are packaged into the solution archive
  - Example: `banking_solution-1.0.esa`
- Exporting agents
  - Only the agent is packaged in the archive, but the name includes the solution name plus the archive name plus version
  - Example: `banking_solution-0.1-banking_solution.banking_agent_fraud_detection-0.1.0.esa`

Managing deployment

Figure 10-5. Exporting solutions and agents

When you export a solution and its agents to an archive, it creates a `.esa` file. You then deploy the archive to an Insight Server.

If you modify an agent, you can export the agent independently from the solution to update it. The solution must exist on the server before you deploy a single agent.

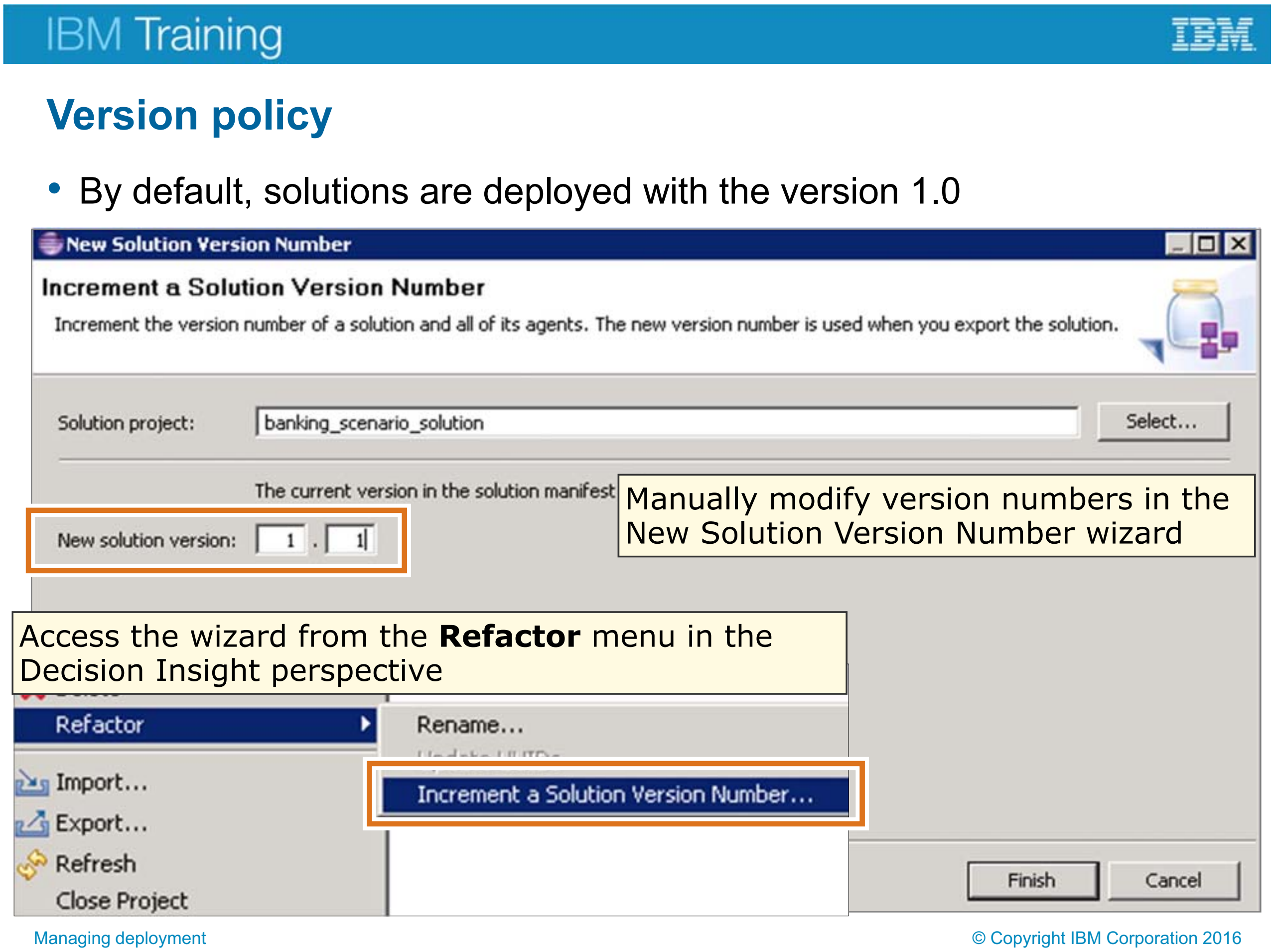


Figure 10-6. Version policy

Before you export a new or updated version of a solution, you can use the New Solution Version Numbers wizard to increment the version number of the solution project and its agents.

Incrementing the version before you export prevents overwriting a deployed solution.

When you export an agent, the Export wizard increments the micro number, for example: 1.0.1. This version in recorded in the manifest file of the agent project.

The `MANIFEST.MF` files of Java agent and OSGi projects might contain import and export declarations that include a version. When you increment the solution version, the version in the `Import-Package` and `Export-Package` headers is synchronized with the new version. This update happens only if the export and import declarations have a `version` attribute that is the same as the bundle version, and if it ends with the `.qualifier` property.

IBM Training

Deploying solution archives

- To deploy a solution from Insight Designer, you can:
  - Create a deployment configuration for the solution
  - Run the solutionManager script
  - Use JMX MBeans
- Deployment installs the solution files on the server and updates the `server.xml` file to activate the solution on the Liberty server
- Run the `solutionManager` script on Windows from the `<InstallDir>/runtime/ia/bin` directory
- You can deploy locally or remotely

Managing deployment

Figure 10-7. Deploying solution archives

You can deploy a solution by running the `solutionManager` script or by using JMS MBeans. Deployment installs the solution files on the target server and updates the `server.xml` file. When you deploy to a grid, the same solution must be installed on all the runtime servers in the grid.

You can run the `solutionManager` script to deploy locally or remotely. For example:

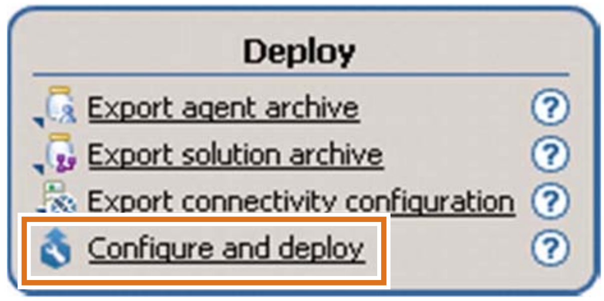
```
solutionManager deploy remote C:\solution.esa --host=someRemoteHost --port=9080
--username=user1 --password=user1
--trustStoreLocation=<InstallDir>\runtime\wlp\usr\servers\cisDev\resources\
security\key.jks --trustStorePassword=truststore
```

The `solutionManager` script is run from the `<InstallDir>/runtime/ia/bin` directory.

IBM Training

Using a deployment configuration

- Use the **Configure and Deploy** wizard to generate a deployment configuration for your solution
  - Define local or remote servers
  - Define connection properties
- Use the **Configure and Deploy** wizard with either of these methods:
  - Right-click the solution and select **Deploy > Configure and Deploy**
  - In the Deploy goal of the Solution Map, click **Configure and deploy**



- Configurations (`config_name-config.xml`) are stored in the **Deployment Configurations** folder of the solution project

Managing deployment

Figure 10-8. Using a deployment configuration

IBM Training

Local deployment

- Using the `local` parameter
  - Run the script with the Liberty server started or stopped
  - If the server is stopped when you deploy an archive, the solution or agent is accounted for when the server starts

- Example

```
solutionManager deploy local C:\solution.esa --
server=cisDev
```

Managing deployment

Figure 10-9. Local deployment

IBM Training

Remote deployment

- Using the `remote` parameter
  - The script establishes an HTTP connection to a running Liberty server on the remote host
  - You must provide administrator authentication credentials to run the script
- Example

```
solutionManager deploy remote C:\solution.esa --
host=cisContainer1 --port=9080 --username=user1
--password=ins1ghts
--trustStoreLocation=C:\IBM\ODMInsights88\runtime\wlp\usr\
servers\cisCatalog1\resources\security\key.jks
--trustStorePassword=ins1ghts
```

Managing deployment

Figure 10-10. Remote deployment

IBM Training

Using connection properties files

- Create a unique `connection.properties` file for each remote server
  ```
  solutionManager deploy remote C:\mySolution-0.0.esa
  --propertiesFile=../etc/connectionC01.properties
  ```
- Example connection properties for a container
  - server=`cisContainer1`
  - host=`containerHost1`
  - port=`9443`
  - username=`admin`
  - password=`ins1ghts`
  - trustStoreLocation=`${wlp.user.dir}/servers/cisCatalog1/resources/security/key.jks`
  - trustStorePassword=`ins1ghts`
  - sslProtocol=`TLS`
  - disableSSLHostnameVerification=`true`

Managing deployment

Figure 10-11. Using connection properties files

To simplify passing the remote host parameters to the remote `deploy` command, you can pass a properties file through the `propertiesFile` parameter.

```
--propertiesFile=InstallDir/runtime/ia/etc/myconnection.properties
```

You can define unique `connection.properties` files for each server that you are using.

IBM Training

Deploying to multiple servers

- Multiple servers that are hosted on the same computer
  1. Deploy a solution to one runtime server
  2. Deploy the same solution to the other runtime servers by using the `activateOnly=true` parameter with the `solutionManager deploy` command
- Multiple servers on remote hosts
  - Set the `solutionAutoStart` property in the `server.xml` files to `false`
  - The auto-start property places a new version of a deployed solution on hold so that you can deploy the new solution version on all the container servers in the topology before you make it active

Managing deployment

Figure 10-12. Deploying to multiple servers

In a topology where multiple servers are hosted on the same host, you first deploy a solution to one container server. You then deploy the same solution to the other runtime servers by using the `activateOnly=true` parameter with the `solutionManager deploy` command. This parameter ensures that the `server.xml` file is updated, without attempting to redeploy the solution archives. Another use of the `activateOnly` parameter is when you undeploy a solution version and you then want to deploy it again. The `undeploy` command does not delete the archives from the host.

In a production topology with multiple servers, the `solutionAutoStart` property in the `server.xml` files is set to false. This auto-start property places a new version of a deployed solution on hold. You can then deploy the new solution version on all the container servers in the topology before you make it active by running the `activate` command.

If the `deploy` command fails and displays an error because the solution was previously deployed but not activated, you can rerun the command with the `activateOverride` parameter to force the deployment of the solution.

IBM Training

Deployed solutions files

- After you deploy, the solution is copied to the `<InstallDir>/runtime/wlp/usr/extension/lib` directory
  - Agents and aggregates are copied to the `/runtime/wlp/usr/extension/lib` folder
  - The solution manifest file is copied to the `/runtime/wlp/usr/extension/lib/features` folder
- Use the REST API to verify deployment by typing this URL in a browser:

  http://*hostname*:*port*/ibm/ia/rest/solutions

Managing deployment © Copyright IBM Corporation 2016

Figure 10-13. Deployed solutions files

After you deploy a solution, you can open Windows Explorer to find the solution files that are stored in the `runtime/wlp/usr/extension/lib` folder. The solution manifest file is stored in the `lib/features` folder.

You can also verify deployment with the REST API in a browser, as you saw during the exercises.

IBM Training

Undeploying (1 of 2)

- Before you undeploy a solution, you must stop it
- Run the `solutionManager stop` command to stop and deactivate a solution
  - Example:
    `solutionManager stop banking_solution`
- If you try to undeploy a solution while it is active, you get errors

Managing deployment

Figure 10-14. Undeploying (1 of 2)

If you need to undeploy a solution, you first need to stop the solution and deactivate it.

You might choose to stop and deactivate the active solution if, for example:

- The solution is not working correctly and must be stopped, deployed again, and then restarted
- You want to undeploy and remove the active version of the solution
- An update to the business object model (BOM) makes the previous and active solution versions incompatible
- You want to stop the solution from processing events without undeploying the solution

If you attempt to undeploy a solution version when it is the active version and it is processing events, an error occurs.

IBM Training

Undeploying (2 of 2)

- After running the `stop` command, you can undeploy by running the `undeploy` command
  - Example:
    `solutionManager undeploy local MySolution --server=cisDev`
- The `undeploy` command removes the solution feature from the `server.xml` file
- Solution artifact files, such as the manifest file and the feature `.jar` files are left in the `<InstallDir>/runtime/wlp/usr/extension/lib` directory
  - Before you redeploy, you must delete these files
- If you try to deploy the solution again without incrementing the solution version number, the script detects the solution feature in the manifest files and displays an error

Managing deployment

Figure 10-15. Undeploying (2 of 2)

When you undeploy a solution, the solution feature is removed from the `server.xml` file.

The solution artifact files, such as `.jar` files and the manifest files, remain in the file system. If the solution artifact files are not used by other installed solutions, you can delete these files by running the `solutionManager delete` command.

IBM Training

Deleting solution files

- To remove a solution from the server after running the `undeploy` command, use the `solutionManager delete` command
  - Example: `solutionManager delete banking_solution-0.1`
- The `solutionManager delete` command deletes the solution manifest (`.mf`) file and the solution feature `.jar` files
  - After removing these files, you can deploy the same version of the solution again
- To delete the solution:
  1. Stop the server
     `server stop cisDev`
  2. Run the solutionManager `delete` command
     `solutionManager delete banking_solution-0.1`
  3. Restart the server with the `–clean` option to remove any cached files
     `server start cisDev --clean`

Managing deployment

Figure 10-16. Deleting solution files

To delete these files, open the `<InstallDir>/runtime/wlp/usr/extension/lib` directory and delete them when you want to remove the solution from the server completely. To delete the retained solution files, run the `solutionManager delete` command.

You can stop and start the server by using the server management script, which is stored in the `<InstallDir>/runtime/wlp/bin` directory. This script includes these actions:

- **create**: Creates a server
- **start**: Starts a server as a background process
- **stop**: Stops a server

IBM Training

Unit summary

- Explain how to export and deploy solutions
- Describe how to manage solutions with the solutionManager script
- Manage deployment to multiple hosts

Managing deployment

Figure 10-17. Unit summary

IBM Training

Review questions

1. True or false: Decision Server Insights supports hot deployment.

2. True or false: You can undeploy a solution without stopping it.

3. True or false: An agent archive can be deployed by itself.

4. True or False: You use the `solutionManager` deploy command for local deployment only.

Managing deployment

Figure 10-18. Review questions

Write your answers here:

1.

2.

3.

4.

IBM Training

Review answers

1. True or false: Decision Server Insights supports hot deployment.
   Answer: True.

2. True or false: You can undeploy a solution without stopping it.
   Answer: False. You must stop the solution first.

3. True or false: An agent archive can be deployed by itself.
   Answer: True.

4. True or False: You use the `solutionManager` deploy command for local deployment only.
   Answer:* False. *You use the solutionManager deploy command for both local and remote deployment.

Managing deployment

Figure 10-19. Review answers

IBM Training

Exercise: Deploying solutions

Managing deployment

Figure 10-20. Exercise: Deploying solutions

IBM Training

Exercise objectives

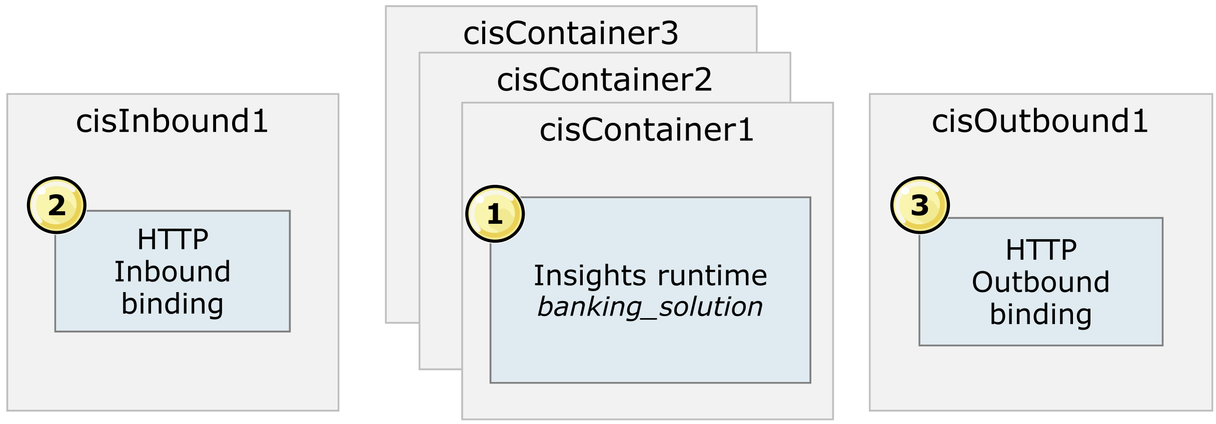
- Use solutionManager to deploy solutions
- Manage deployment and connectivity for a grid environment

Managing deployment

Figure 10-21. Exercise introduction

IBM Training

Exercise overview: Deploying a solution and connectivity



1. Deploy solution to runtime servers on container hosts
2. Deploy inbound configuration to inbound server
3. Deploy outbound configuration to outbound server

Managing deployment

Figure 10-22. Exercise overview: Deploying a solution and connectivity

During the exercise, you deploy a solution to the grid containers. You deploy inbound connectivity to the inbound server, and outbound connectivity to the outbound server.

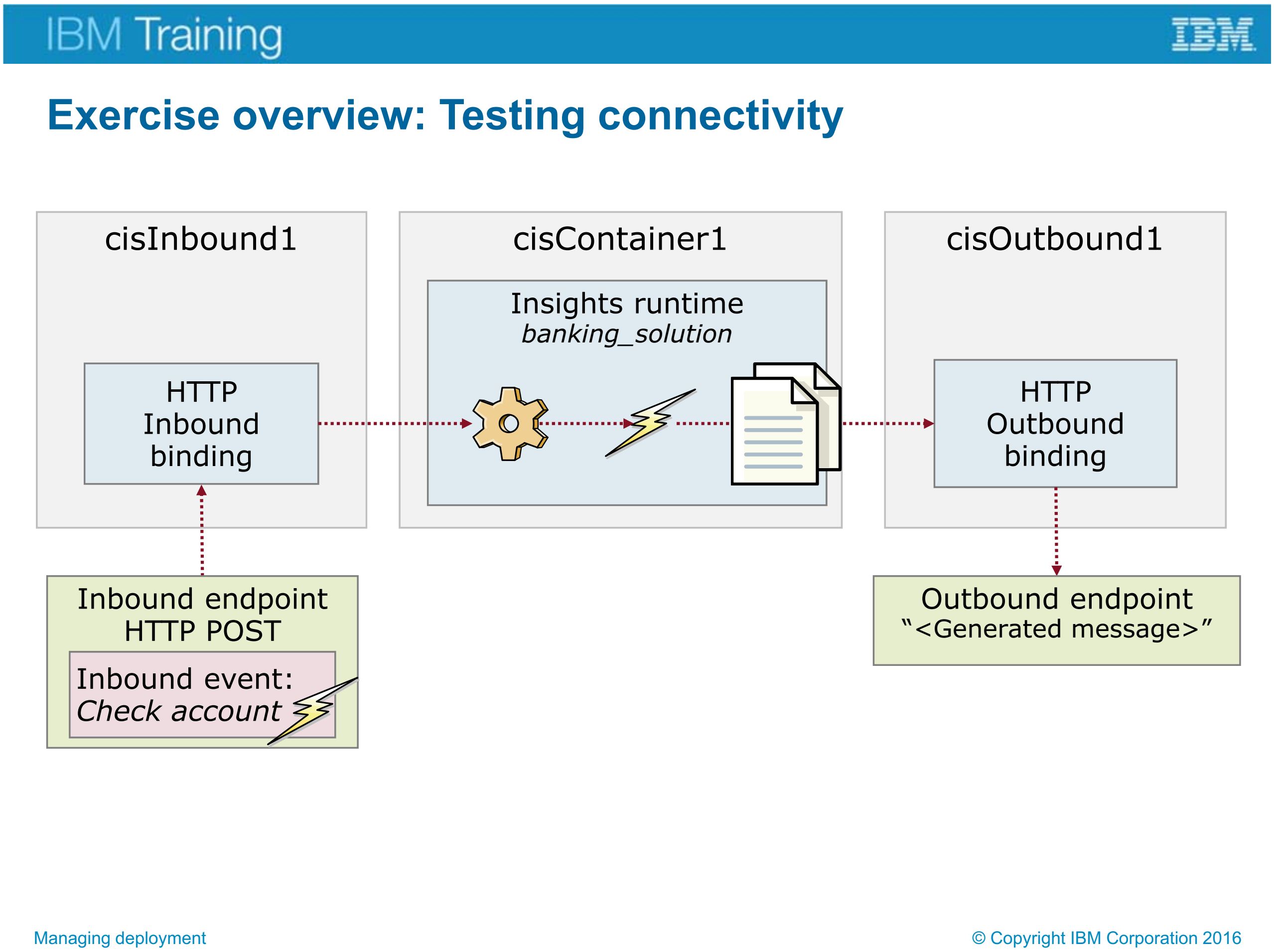


Figure 10-23. Exercise overview: Testing connectivity

To test the deployment, you use an HTTP tool to submit an event through the inbound server. The result of successful processing of the event is that an outbound message is generated by the runtime and the outbound connectivity server sends that message to the outbound endpoint.

Unit 11. Administering Decision Server Insights

Estimated time

01:00

Overview

This unit explains how to administer Decision Server Insights.

How you will check your progress

- Checkpoint
- Exercise

IBM Training

Unit objectives

- Use administration scripts to monitor status and activity of your servers and grid
- Use trace files and logging to monitor Decision Server Insights
- Describe how to monitor WebSphere eXtreme Scale and WebSphere MQ

Administering Decision Server Insights

Figure 11-1. Unit objectives

IBM Training

Topics

- Administration tools
- Logging
- Monitoring WebSphere eXtreme Scale
- Monitoring WebSphere MQ

Administering Decision Server Insights

Figure 11-2. Topics

11.1. Administration tools

IBM Training

Administration tools

Figure 11-3. Administration tools

IBM Training

Liberty profile monitoring

- JVM monitoring
  - Use the JvmStats MXBean for JVM monitoring in the Liberty profile
- ThreadPool monitoring
  - Use a ThreadPool MXBean
- Multiple components monitoring with Insight Monitor
  - Use the `monitor-1.0` feature

Administering Decision Server Insights

Figure 11-4. Liberty profile monitoring

Some of the important monitor components that you want to monitor as part of Decision Server Insights and WebSphere eXtreme Scale are JVM and ThreadPool. ThreadPool is critical when you have a large number of partitions on a server or on a single host server. If you have hundreds of partitions per server, thousands of threads are allocated, which can quickly lead to resource contention.

IBM Training

Server administration scripts

- Decision Server Insights administration scripts to:
  - Deploy and manage solutions
  - Deploy and manage connectivity
  - Manage server properties and activity

Property	Description
propertyManager	Use the propertyManager script to get, set, and manage server properties
connectivityManager	Use the connectivityManager script to generate, deploy, and activate connectivity for your solution
serverManager	Use the serverManager script to pause or shutdown server processes, and check the `isonline` status for a server
solutionManager	Use solutionManager to deploy and manage solutions across your servers

Administering Decision Server Insights

Figure 11-5. Server administration scripts

Here you see listed some of the administration scripts that you use for managing servers, solutions, and connectivity. You worked with the `solutionManager` and `connectivityManager` scripts during the connectivity and deployment exercises.

For more information about using these scripts, see the “Reference” section of the product documentation.

IBM Training

Managing servers

- Use `serverManager isonline` command to determine whether a server is running

- Run from the *InstallDir*\runtime\ia\bin directory

```
serverManager isonline [--propertiesFile=properties_file]
[--username=username] [--password=password]
[--host=hostname] [--port=port]
[--keyStoreLocation=keystore_location]
[--keyStorePassword=keystore_password]
[--sslProtocol=sslProtocol]
[--trustStorePassword=truststore_password]
[--trustStoreLocation=truststore_location]
[--disableSSLHostnameVerification=true|false]
```

  - Example

```
serverManager isonline --propertiesFile=../etc/
connectionC1.properties
```

Administering Decision Server Insights

Figure 11-6. Managing servers

The `serverManager` script includes the `isonline` command, which checks for running and active local or remote servers. Use the command to identify a local or remote server where you can deploy or activate solutions.

You must provide administrator authentication credentials to run the script. Specify the server host name and port to determine the status of a remote server. You can pass these credentials and server details through a unique `connection.properties` file with the `propertiesFile` parameter.

IBM Training

Managing server properties

- Use `propertyManager` to set and retrieve server properties and log settings

- Run from the *InstallDir*\runtime\ia\bin directory

```
propertyManager list [--propertiesFile=properties_file]
[--username=username] [--password=password]
[--host=hostname] [--port=port]
[--keyStoreLocation=keystore_location]
[--keyStorePassword=keystore_password]
[--sslProtocol=sslProtocol]
[--trustStorePassword=truststore_password]
[--trustStoreLocation=truststore_location]
[--disableSSLHostnameVerification=true|false]
```

  - Example

```
propertyManager list --propertiesFile=../etc/
connectionC1.properties
```

Administering Decision Server Insights

Figure 11-7. Managing server properties

You use the **`propertyManager`** script to manage log settings and server properties. This script includes the **`list`** command to list which properties can be set on your server.

To view the list of properties that you can set with the **`propertyManager`** script, run the **`list`** command.

As with the **`serverManager`** script, you can pass credentials and server details through a unique `connection.properties` file with the `propertiesFile` parameter.

11.2. Log analysis

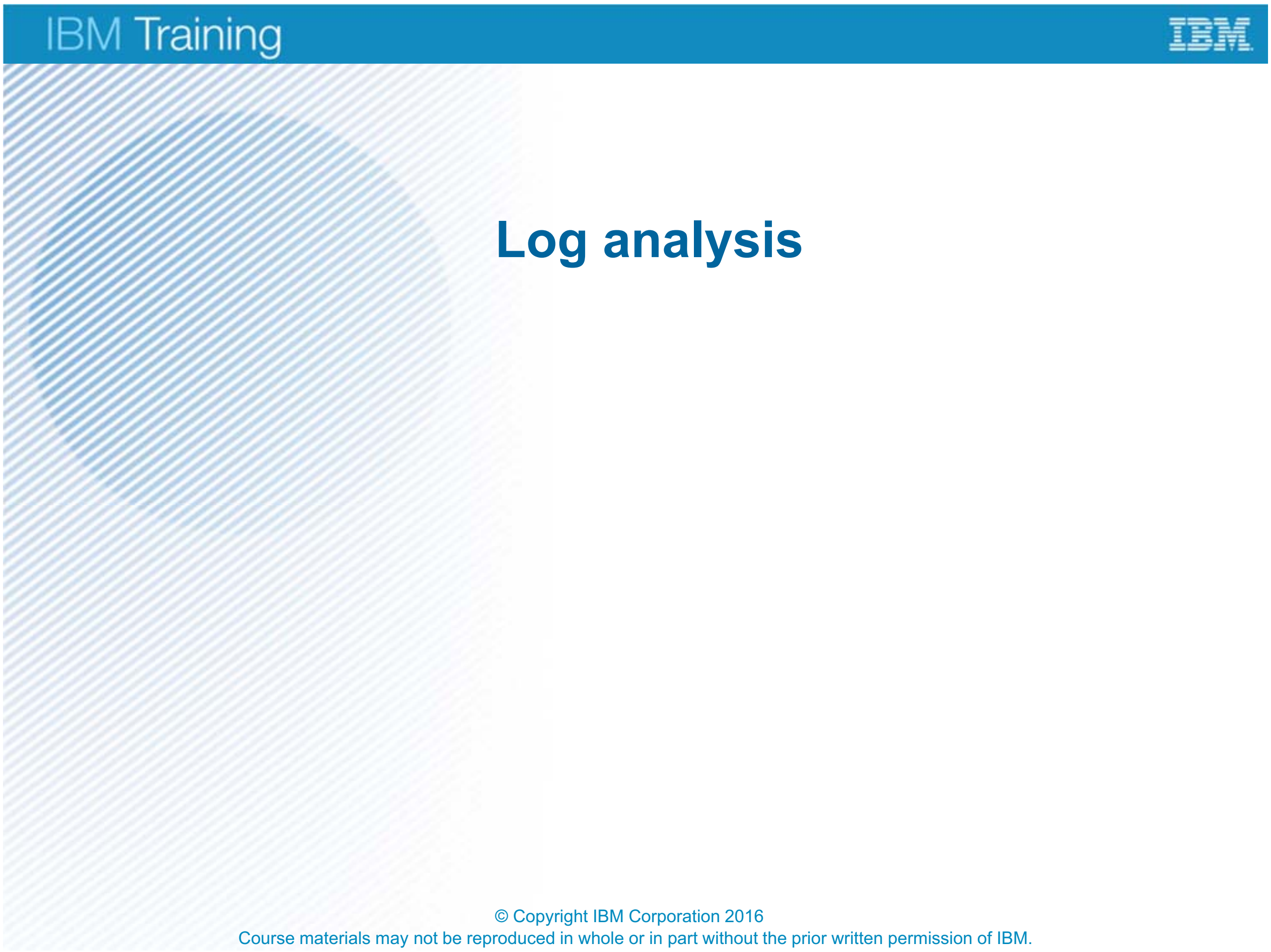


Figure 11-8. Log analysis

IBM Training

Why are log files important?

- Runtime components (Insights Runtime, eXtreme Scale, Rule Engine, Connectivity) all log to the Liberty server logs
- The processing of an “update account” event on the Getting Starting solution in a cisDev server: about 1000 lines in the `trace.log` file
  - The answer is likely to be within those lines
- Key information for finding your way in the logs:
  – JSON serialization of events and entities
  – Keywords to spot the processing by agents and rules
- Beware of null handling in rule agents, prevents NPE but renders rules silent

Administering Decision Server Insights

Figure 11-9. Why are log files important?

Log analysis for runtime components is essential for troubleshooting. The runtime components, including the Insights runtime, Extreme Scale, the rule engine, and connectivity all log to the Liberty server logs. You need to go through these logs when you run into problems.

The trace log is verbose. For example, say that you run the Getting Started tutorial solution that is provided with the product and you check the logs after processing just one event. For the update account event, the log generates about 1000 lines. Your answer to a problem is probably in that log, but it can be difficult to go through it. You can use something like the JSON serialization of events and entries to make the log more readable. Or if you search for the processing that is done by agents and rules, you can use keywords to help focus your search.

If you look for the execution of a specific rule but you cannot find that rule in the log (as if the execution did not happen), the reason might be a Null Pointer Error (NPE). The rule might have been called, then silently ignored because the entity that it was supposed to test either did not exist or was not initialized. Keep in mind that an NPE might have occurred, which would be why you cannot find the execution log entry for the specific rule.

IBM Training

Where and what to gather

- The `trace.log` and First Failure Data Capture (FFDC) files
- Also, collect the configuration files of the servers:

```
wlp/
+-usr/
    +-servers/
       +-<my_server>
      +-grids/
      +-bootstrap.properties
      +-jvm.options
      +-server.xml
      +-logs/
```

- Collect these files for each server of the grid

Administering Decision Server Insights

Figure 11-10. Where and what to gather

When you gather files for troubleshooting, make sure that you include the trace log and the FFDC files.

FFDC files capture the exception stack and any additional data that is generated when an exception error occurs. For example, if an NPE occurs, it generates an FFDC record.

Along with the trace log and FFDC files, you should also gather the configuration files that you see listed here for all your runtime nodes.

IBM Training

Log analysis

- To search for incoming events: the keyword is **"received [event]"**
- To search for the agent that is processing the event: the keyword is "**begin processing**"
- Note the ThreadID: **000001ae**
  - It differentiates agents that are processing events simultaneously
- Bound entity, if any, is visible a few lines before
  - The keyword is **"retrieved entity"**
- Exception in the action part: generates an FFDC
- What if you do not see the “Rule […] execution started”?
  - “Null handling” might be responsible
  - In conditions of the rules, any potential NPE is detected and makes the condition false

Administering Decision Server Insights

Figure 11-11. Log analysis

To help you analyze the log, you can search for these keywords.

Examples:

- To see incoming events, you can search with the keywords: `received [`*`events`*`]`
- To find a bound entity, you can search: `retrieved entity`
- To the start point for where an agent processes and event, use the keywords: `begin processing`

  Pay attention to the thread ID, which distinguishes agents that work simultaneously.
- If you need to find a particular rule execution, you can search: `Rule […] execution started`

  If the execution does not exist, remember it might be due to an NPE. You need to find entity value before that agent starts processing, the agent that calls your missing rule, and check the value of that entity before the execution should start.

IBM Training

Log and trace files

- Use these files to monitor the activity and status of your servers

File	When to use	Supported tasks
console.log	Use to view standard process output and error messages from the runtime environment, for run and debug actions. Messages are redirected to the *server_name*/logs/console.log file when you start the server.	View a message that confirms that a solution is deployed and ready to use, for example: `CWMBD0060I: Solution ConnectivitySolution-0.1 ready` View other standard process output and error messages.
messages.log	Use to view operational messages and information that is generated by the system.	View INFO, AUDIT, WARNING, ERROR, and FAILURE messages. View time stamps and the IDs of issuing threads.
trace.log	Use to gather and view debug information.	Gather and view debug information that is obtained by basic, enhanced, or advanced tracing.

Administering Decision Server Insights

Figure 11-12. Log and trace files

On this slide, you see some of the main files that you use to monitor the activity and status of your servers.

For more information, see “Trace and logging” in the WebSphere Application Server, Liberty core documentation.

IBM Training

Server log settings

Attribute	Description
maxFileSize	If an enforced maximum file size exists, this setting is used to determine how many of each of the log files are kept. This setting also applies to the number of exception logs that summarize exceptions that occurred on any particular day. The default value is 2. **Note:** maxFiles does not apply to the console.log file.
consoleLogLevel	This filter controls the granularity of messages that go to the console.log file. The valid values are INFO, AUDIT, WARNING, ERROR, and OFF. The default level is AUDIT.
traceSpecification	This attribute controls the format of the trace log. The default format for the Liberty profile is ENHANCED. You can also use BASIC and ADVANCED formats as in the full profile.

Administering Decision Server Insights

Figure 11-13. Server log settings

The logging component can be controlled through the server configuration. The primary location for the logging configuration is in the server.xml file. Occasionally, you might need to configure trace to diagnose a problem that occurs before the server.xml file is processed. In this case, the equivalent configuration properties can be specified in the bootstrap.properties file. If a configuration property is specified in both the bootstrap.properties file and the server.xml file, the value in bootstrap.properties is used until the server.xml file is processed. Then, the value in the server.xml file is used. Avoid specifying different values for the same configuration property in both the bootstrap.properties and the server.xml file.

You can set logging properties in the server configuration file by selecting **Logging and Tracing** in the Server Configuration view in the developer tools. You can also add a logging element to the server configuration file as follows:

<*component*> = <*level*>

In this case, <*component*> is the component for which to set a log detail level, and <*level*> is one of the valid log levels (off, fatal, severe, warning, audit, info, config, detail, fine, finer, finest, all). Separate multiple log detail level specifications with colons (:).

For more information, see “Trace and logging” in the WebSphere Application Server, Liberty core documentation.

IBM Training

Server log settings

- Default log levels are set for each server type
- The following `server.xml` elements define the default log levels for each server type

Server type	Log level
cisDev (default development server)	`<logging traceSpecification="*=info" maxFiles="5" />`
cisCatalog	`<logging traceSpecification="*=info" maxFiles="5" />`
cisContainer, **cisInbound**, **cisOutbound**	`<logging traceSpecification="com.ibm.ia.*=info:com.ibm.rules.*=info:*=info" maxFiles="5" />`

- You can set the logging level to one of the following values: `severe, warning, audit, info, fine, finer, finest, off`

Administering Decision Server Insights

Figure 11-14. Server log settings

You change the level for a log by editing the `server.xml` files, and in some cases the `bootstrap.properties` file.

See
`http://www.ibm.com/support/knowledgecenter/SSQP76_8.8.0/com.ibm.odm.itoa.admin/topics/con_logging_settings.html`

IBM Training IBM

Changing log and trace settings

- Edit the trace and logging properties in the `server.xml` file on each of the containers, inbound and outbound servers, and catalogs
  - **Note:** Make a backup of the `server.xml` file before you edit

- Create a trace file for a solution with `messageFileName` parameter to the <`logging`> entry
  - Example:

```
<logging
traceSpecification="com.ibm.rules.generated.dataie.banking
_
scenari.* =detail:com.ibm.ia.*=warning:
com.ibm.ia.runtime.SolutionProviderMgr=finest:com.ibm.rule
s.*=info:*=warning" maxFiles="10"
messageFileName="bankingSolutionMessags.log"/>
```

Administering Decision Server Insights © Copyright IBM Corporation 2016

Figure 11-15. Changing log and trace settings

To change log settings for your grid, you edit the `server.xml` files for each of the containers, the inbound and outbound servers, and the catalogs. You can make these changes while the servers are running. You do no not need to restart the servers because the changes are detected and applied automatically.

To generate more extensive traces of a solution, you can add or modify logging entries in the server.xml file. For example, you can create a trace file to capture logs for a particular solution by adding the **`messageFileName`** parameter. You can also increase the number of files that are generated for each log by setting the **`maxFiles`** parameter. If you increase this number to 10, for example, you might have 10 message logs, 10 trace logs, and 10 exception summaries in the `ffdc` directory. The trace file is created only if you enable additional trace.

IBM Training

Finding troubleshooting information in log and trace files

- Analyze the `trace.log` and `messages.log` files
- Search for key phrases

Key phrase	Information
received *event_name*	ID, timestamp, and other attributes of an incoming event
begin processing	Agent processing activity
retrieved entity	Class name, ID, and other attributes of a bound entity
rule *rule_name* execution started	Rule processing and output

Administering Decision Server Insights

Figure 11-16. Finding troubleshooting information in log and trace files

See:
http://www.ibm.com/support/knowledgecenter/SSQP76_8.8.0/com.ibm.odm.itoa.admin/topics/con_find_trb_log_files.html

IBM Training

Server administration properties

- Server properties include the following properties

Property	Description	Default
debugPort	Indicates one or more ports that the server monitors for connections from a test driver instance. The value is a single port, or a range or ports, which matches the `httpport` property as specified in the `testdriver.properties` file.	None
LogSuppressionThreshold	Defines the number of occurrences of any log message that is allowed in a specified time period. The value of this property determines the number of times a log message is produced before it is suppressed.	10
LogSuppressionThresholdPeriod	Defines the time period during which log messages are counted. A log message rate is calculated from the value of the log threshold and the time period for the threshold. When the message rate is deemed too high, all subsequent messages are suppressed. Messages are produced again only when the rate falls to an acceptable volume.	20000

Administering Decision Server Insights

Figure 11-17. Server administration properties

The `propertyManager set` command sets properties that modify the behavior and characteristics of the server. The properties apply across to all solutions on the server.

In the development environment, the `<ia runtime>` entry might include the `debugPort` property and `solutionAutoStart="true"`.

```
<ia_runtime debugPort="6543" solutionAutoStart="true"/>
```

You can use the `LogSuppressionThreshold` to limit the number of times a log message is produced before it is suppressed. For example, you can reduce the threshold from `10` to `2`. This value means that after 2 log messages are generated within the threshold period, subsequent messages are suppressed.

You can also change other values, such as `LogSuppressionThresholdPeriod`, which defines the time period over which log messages are counted. The default is 20000 ms. If you increase this value, the same number of messages are produced, but over a longer period. As a result, you receive fewer messages.

11.3. Monitoring WebSphere eXtreme Scale

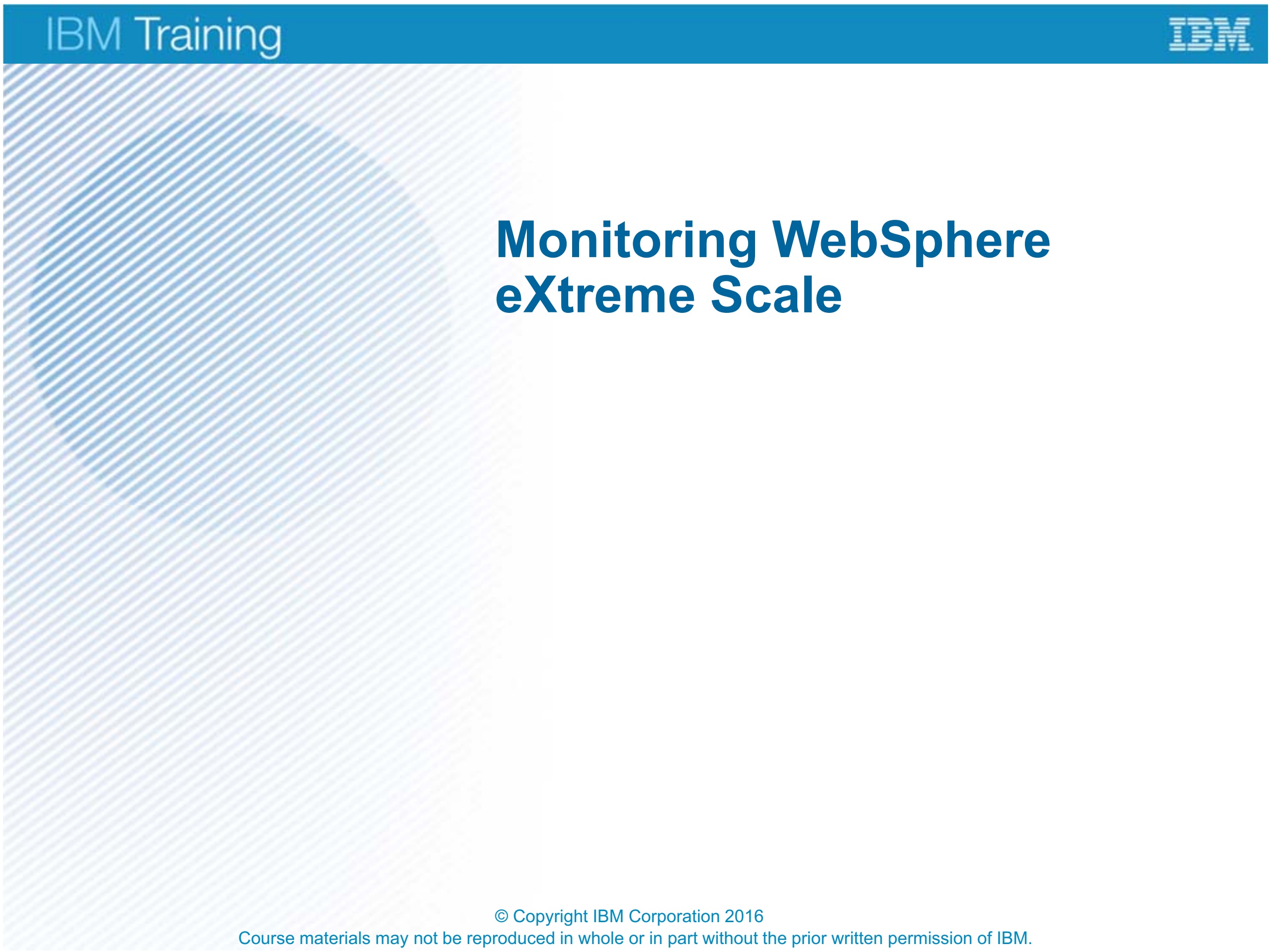


Figure 11-18. Monitoring WebSphere eXtreme Scale

IBM Training

WebSphere eXtreme Scale: xscmd

- The `xscmd` utility displays textual information about the grid topology
  - Make sure that catalog servers and container servers are started before using this tool
  - If your catalog servers are set up with majority quorum enabled, at least two catalog servers must be started

Administering Decision Server Insights

Figure 11-19. WebSphere eXtreme Scale: xscmd

With `xscmd`, you can display textual information about your Decision Server Insights WebSphere eXtreme Scale grid topology.

You can use a number of tools and techniques to determine what is happening in an eXtreme Scale environment during run time.

As mentioned earlier, you want to monitor the JVM and ThreadPool, especially when you have a large number of partitions that are on a single host server to avoid resource contention.

IBM Training

Monitoring that containers and catalogs are communicating

- To verify whether the catalogs are running, use the `xscmd routetable` command
  - Example:
    `xscmd -c routetable -cep localhost:2810`
- This command returns the list of containers that are associated with the catalog service and the route table of partitions

Administering Decision Server Insights

Figure 11-20. Monitoring that containers and catalogs are communicating

You can use the extreme scale `xscmd` utility to verify that all your servers are running.

As you see here, you can use the `routeable` command to verify that the catalogs and containers are communicating. This command also lists all the partitions on the container servers.

IBM Training

Monitoring the status of quorum (1 of 2)

- To verify the status of your catalogs, use the eXtreme Scale `xscmd showQuorumStatus` command
  - Example:

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

- This command returns the status of the catalogs that are associated with the catalog service

```
C:\IBM\ODMInsights87\runtime\wlp\bin>xscmd -c showQuorumStatus -cep localhost:2809
Starting at: 2015-04-30 02:48:58.181
CWXSI0068I: Executing command: showQuorumStatus
Server                Host    Quorum   Quorum Size Active Servers
------                ----    ------   ----------- --------------
localhost-cisCatalog1 dsiMain DISABLED -              localhost-cisCatalog1,
                                                   localhost-cisCatalog2,
                                                   localhost-cisCatalog3
localhost-cisCatalog2 dsiMain DISABLED -              localhost-cisCatalog1,
                                                   localhost-cisCatalog2,
                                                   localhost-cisCatalog3
localhost-cisCatalog3 dsiMain DISABLED -              localhost-cisCatalog1,
                                                   localhost-cisCatalog2,
                                                   localhost-cisCatalog3
CWXSI0040I: The showQuorumStatus command completed successfully.
Ending at: 2015-04-30 02:48:59.884
```

Administering Decision Server Insights © Copyright IBM Corporation 2016

Figure 11-21. Monitoring the status of quorum (1 of 2)

Here, you see an example of running the eXtreme Scale utility to check quorum status. The `showQuorumStatus` command returns the status of all catalogs that are associated with this catalog service.

In this example, three catalogs are in the service, but quorum is disabled.

IBM Training

Monitoring the status of quorum (2 of 2)

- Quorum includes these statuses:

Status	Descriptions
`TRUE`	The server has quorum enabled and the system is working normally. Quorum is met.
`FALSE`	The server has quorum enabled, but quorum is lost. The catalog servers do not allow changes to the catalog service domain.
`UNAVAILABLE`	The server cannot be contacted. It is either not running, or the server cannot be reached because of a network problem.
`DISABLED`	The server does not have quorum enabled.

Administering Decision Server Insights

Figure 11-22. Monitoring the status of quorum (2 of 2)

This slide shows a list of quorum statuses that you can expect to see for your catalog servers. Later, during the exercises, you learn how to enable majority quorum.

11.4. Using Insight Monitor

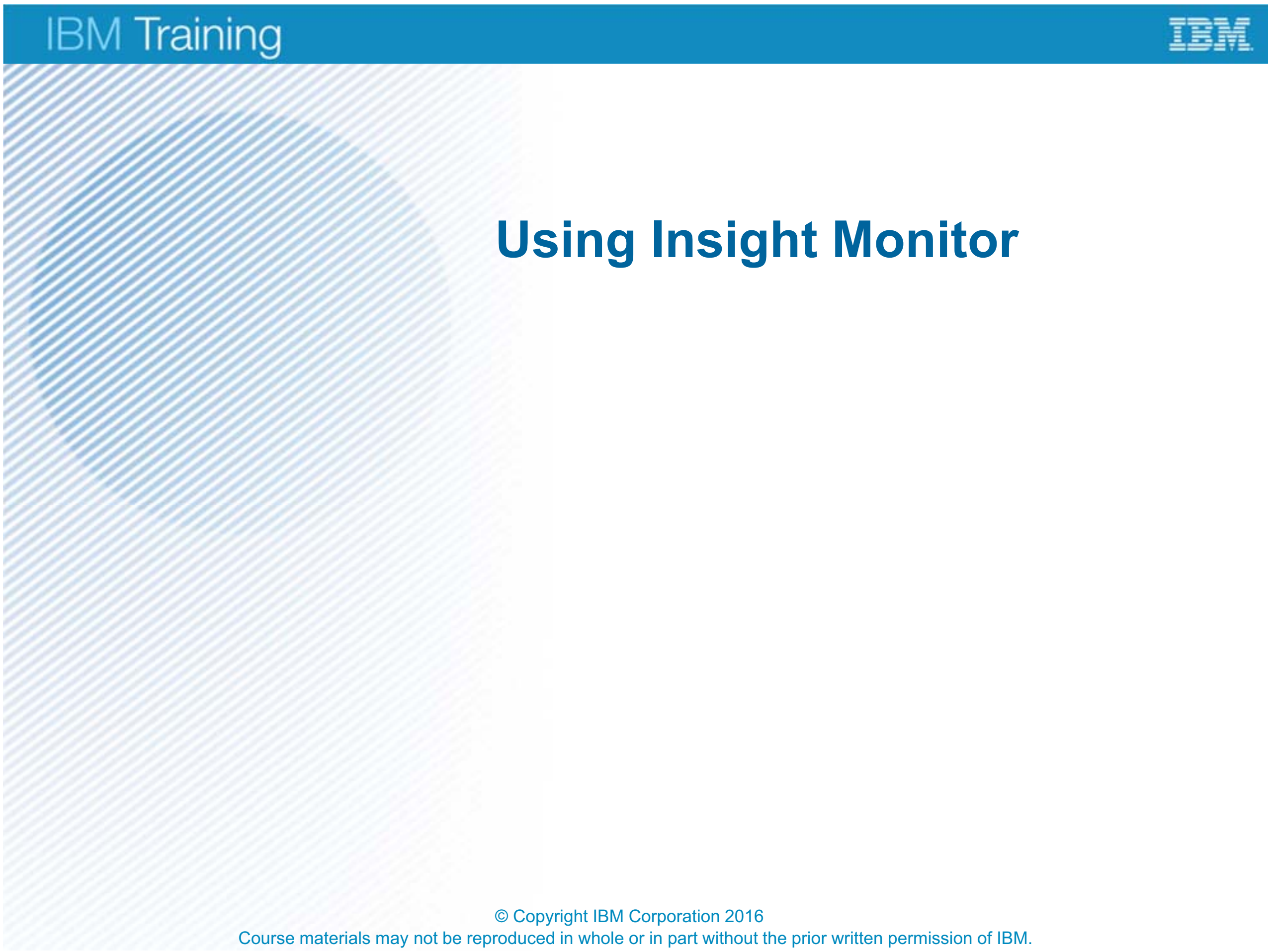
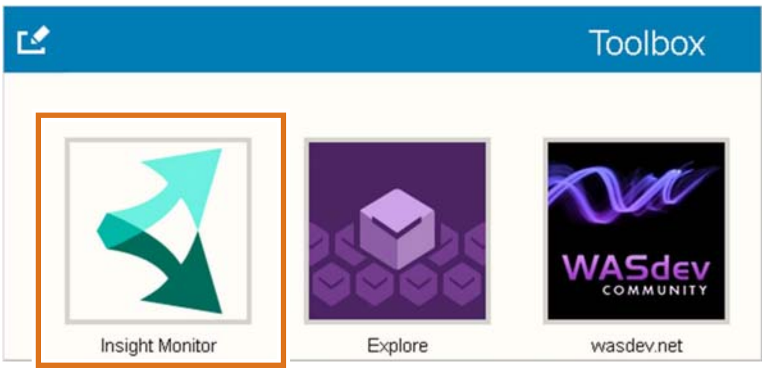


Figure 11-23. Using Insight Monitor

IBM Training

Overview

- Web interface for administrators
  - View event rates
  - Monitor CPU and memory usage on all runtime servers
  - Check system health
- Implemented as a tool in the Liberty admin center



- Launch URL:

  `http:/`*`hostname`*`:`*`port`*`/adminCenter`

Administering Decision Server Insights

Figure 11-24. Overview

Insight Monitor provides a browser-based interface for administrators to monitor event activity and system health.

Insight Monitor is implemented as a tool in the Liberty Admin Center.

IBM Training

Enabling Insight Monitor

- Not enabled by default
- Enable Liberty admin center feature on catalog server by adding feature to `server.xml`

```
<feature>adminCenter-1.0</feature>
```

- Enable Liberty monitor feature on each container server by adding feature to server.xml

```
<feature>monitor-1.0</feature>
```

Administering Decision Server Insights

Figure 11-25. Enabling Insight Monitor

Insight Monitor is not enabled by default, but requires manual setup. You enable the admin center on the primary catalog server by adding the `adminCenter` feature to the server.xml file.

You enable monitoring of each of the runtime servers by adding the `monitor` feature to the `server.xml` files for the container servers, the remaining catalog servers, and the inbound and outbound servers.

IBM Training

Authentication between catalog and container servers

- Certificate-based authentication
  - Use keytool
  - Export certificate from catalog/container server
  - Import certificate to each container/catalog server
  - Add `clientAuthenticationSupported="true"` attribute to the `<ssl>` tag to each catalog and container server
- User name and password authentication
  - Set in `server.xml` of catalog server
  - Add `<ia_admincenter>` tag

```
<ia_admincenter
   http.ssl.config="defaultSSLConfig"
   user="tester"
   password="tester" />
```

  - You can also use this tag to set non-default port values

Administering Decision Server Insights

Figure 11-26. Authentication between catalog and container servers

To enable authentication between the catalog and container servers, you can use these methods:

- Certificate-based authentication, by using the Java keytool
- User name and password by using the `ia_admincenter` tag

When you use the `ia_admincenter` tag to define the user name and password authentication, these properties are required:

- `http.ssl.config`
- `user`
- `password`

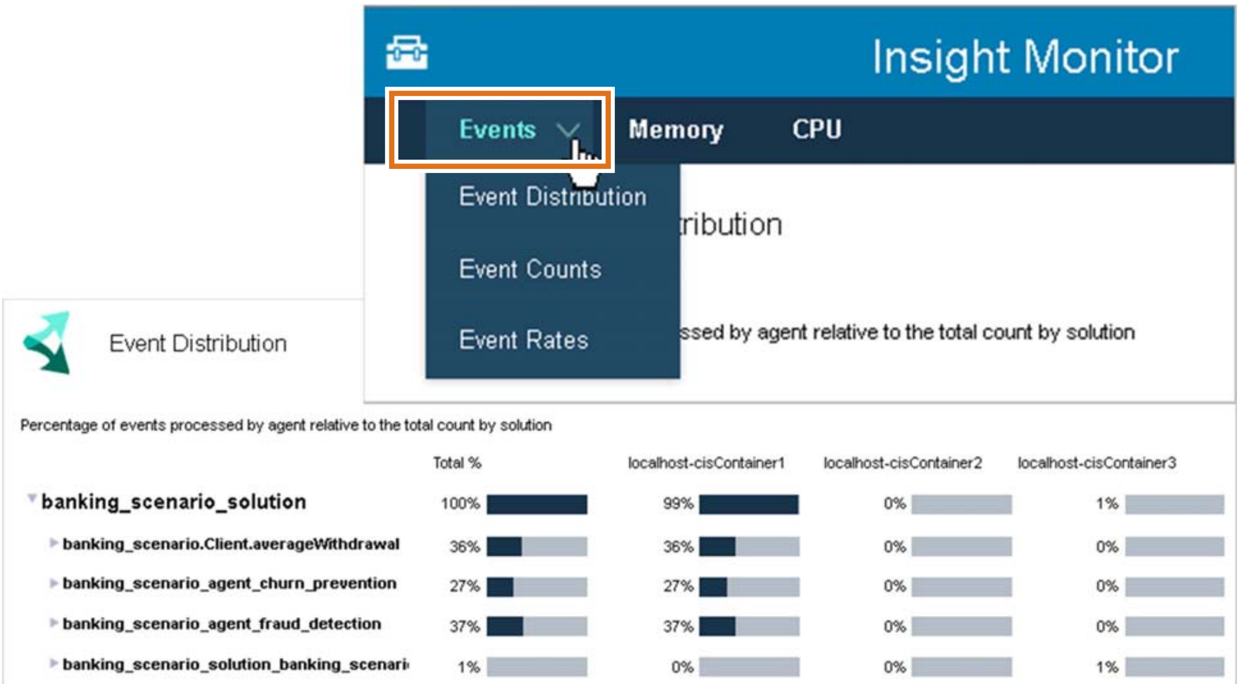
You can also use the `ia_admincenter` tag define other properties, such as non-default port values. For example:

```
<ia_admincenter
http.ssl.config="defaultSSLConfig"
user="tester"
password="tester"
alternate.ports="serverX:9443,serverY:9442" />
```

IBM Training

Monitoring events

- Monitor event distribution, count, or rates of occurrence



Administering Decision Server Insights

Figure 11-27. Monitoring events

The Events pages display event totals from when the runtime server was started. If a server is restarted, the event count is reset. The event count also includes time triggers, which the system schedules, so it can be higher than expected.

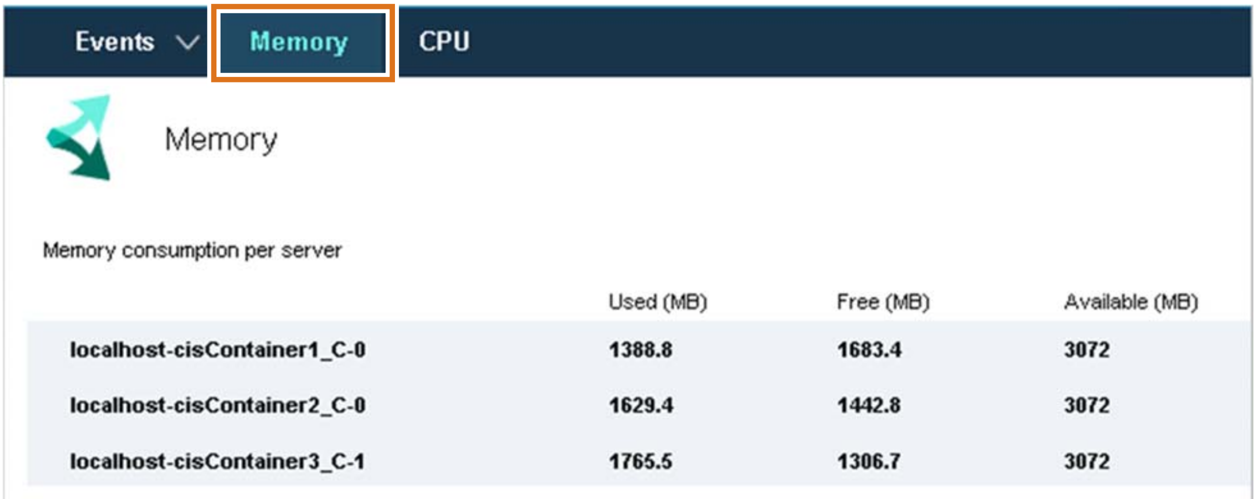
The **Events** menu includes these submenus.

- Event Distribution: Lists the events according to solution across all runtime hosts. The distribution is a percentage that is calculated by the event count per server divided by the total number of events processed.
- Event Count: Shows the number of events that are processed per server. The count includes retries of event processing as a result of processing errors.
- Event Rates: Shows the number of events that are processed per second.

IBM Training

Monitoring runtime server memory

- Monitor memory consumption on each runtime server



	Used (MB)	Free (MB)	Available (MB)
localhost-cisContainer1_C-0	1388.8	1683.4	3072
localhost-cisContainer2_C-0	1629.4	1442.8	3072
localhost-cisContainer3_C-1	1765.5	1306.7	3072

Figure 11-28. Monitoring runtime server memory

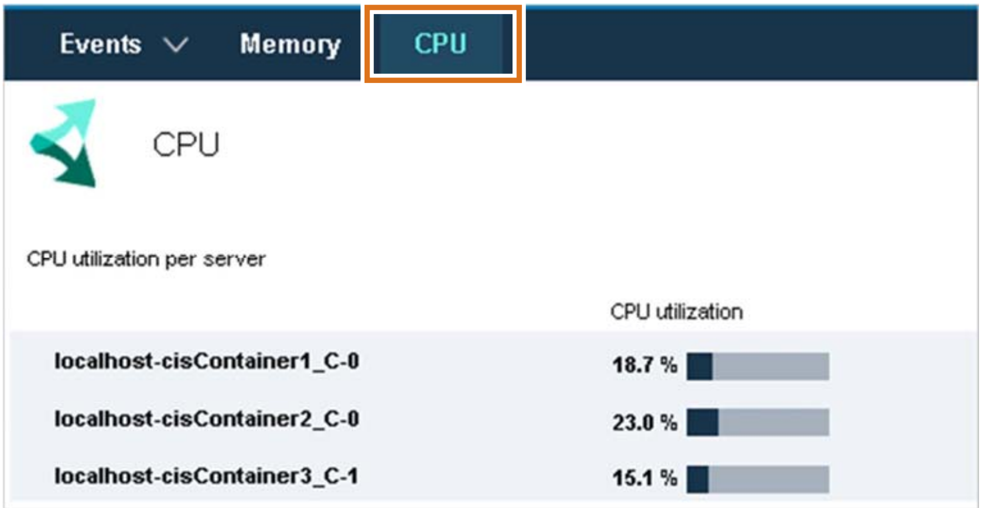
The Memory page lists the memory usage for each runtime server.

Use the information on this page to determine whether the system is running low on Java heap. If the memory usage is near the configured amount, you might want to add more RAM and increase the size of the Java virtual machines.

IBM Training

Monitoring CPU

- Monitor memory consumption on each runtime server



Administering Decision Server Insights

Figure 11-29. Monitoring CPU

The CPU page shows a view of CPU utilization on each runtime server.

Use this page to determine whether the system is running efficiently. If the system has a high event rate, the CPU utilization must be near 100%, which indicates that the system is processing the backlog of events as quickly as possible.

11.5. Monitoring WebSphere MQ

IBM Training

Monitoring WebSphere MQ

Figure 11-30. Monitoring WebSphere MQ

IBM Training

Tracing the WebSphere MQ resource adapter (1 of 2)

Property	Description	Default
logWriterEnabled	A flag to enable or disable the sending of a diagnostic trace to a LogWriter object provided by the application server. If the value is true, the trace is sent to a LogWriter object instead of the location that is specified by the `traceDestination` property. If the value is `false`, any LogWriter object that is provided by the application server is not used.	False
traceEnabled	A flag to enable or disable diagnostic tracing. If the value is `false`, tracing is turned off. If the value is `true`, a trace is sent to the location specified by the `traceDestination` property.	False
traceDestination	The location to where a diagnostic trace is sent • If the value is `System.err`, the trace is directed to the system error stream instead of a file • If the value is `System.out`, the trace is directed to the system output stream Example: `/tmp/wmq_jca.trace` **Note:** If `logWriterEnabled` is set to `True`, the log redirects to the LogWriter	

Administering Decision Server Insights

Figure 11-31. Tracing the WebSphere MQ resource adapter (1 of 2)

With the WebSphere MQ resource adapter, you can configure diagnostic trace as a property on the resource adapter. The resource adapter RAR file contains a file that is called `META-INF/ra.xml`, which contains a deployment descriptor for the resource.

To change the behavior of the log and the settings, you must change the `ra.xml` that is included in the `wmq.jmsra.rar` file.

You extract the files to locate the `ra.xml` file in the **META-INF** folder. After you make the change, you replace the `ra.xml file` back in the `wmq.jmsra.rar` file.

IBM Training

Tracing the WebSphere MQ resource adapter (1 of 2)

- Example

```
<config-property>
  <config-property-name>logWriterEnabled</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>false</config-property-value>
</config-property>
<config-property>
  <config-property-name>traceEnabled</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>true</config-property-value>
</config-property>
<config-property>
<config-property-name>traceLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>6</config-property-value>
</config-property>
<config-property>
<config-property-name>traceDestination</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>/tmp/wm_jca.trace</config-property-value>
</config-property>
```

Administering Decision Server Insights

Figure 11-32. Tracing the WebSphere MQ resource adapter (1 of 2)

In the example that is shown here, the `traceDestination` property is added as the location where the log file should be stored. The `traceLevel` property is changed from `3` (error and warning messages) to `6` (error, warning, and information messages).

For more information, see the IBM Knowledge Center about enabling and changing the trace level for JMS logging and about tracing the WebSphere MQ resource adapter.

IBM Training

Unit summary

- Use administration scripts to monitor status and activity of your servers and grid
- Use trace files and logging to monitor Decision Server Insights
- Describe how to monitor WebSphere eXtreme Scale and WebSphere MQ

Administering Decision Server Insights

Figure 11-33. Unit summary

IBM Training

Review questions

1. You run the Decision Server Insights server administration scripts from which directory:
   A. `<InstallDir>\runtime\ia\bin`
   B. `<InstallDir>\runtime\wlp\bin`

2. **True or False:** After modifying logging properties in the `server.xml` file for your containers and catalogs, you must restart the servers.

3. **True or False:** You use `xscmd` to manage the WebSphere eXtreme Scale grid.

4. You run the `xscmd` script from which directory:
   A. `<InstallDir>\runtime\ia\bin`
   B. `<InstallDir>\runtime\wlp\bin`

Administering Decision Server Insights

Figure 11-34. Review questions

Write your answers here:

1.

2.

3.

4.

IBM Training

Review answers

1. **A.** <*InstallDir*>\runtime\ia\bin

2. **False.** *Changes to the logging properties in the* `server.xml` *file are automatically applied without restarting the server.*

3. **True.**

4. **B.** <*InstallDir*>\runtime\wlp\bin

Figure 11-35. Review answers

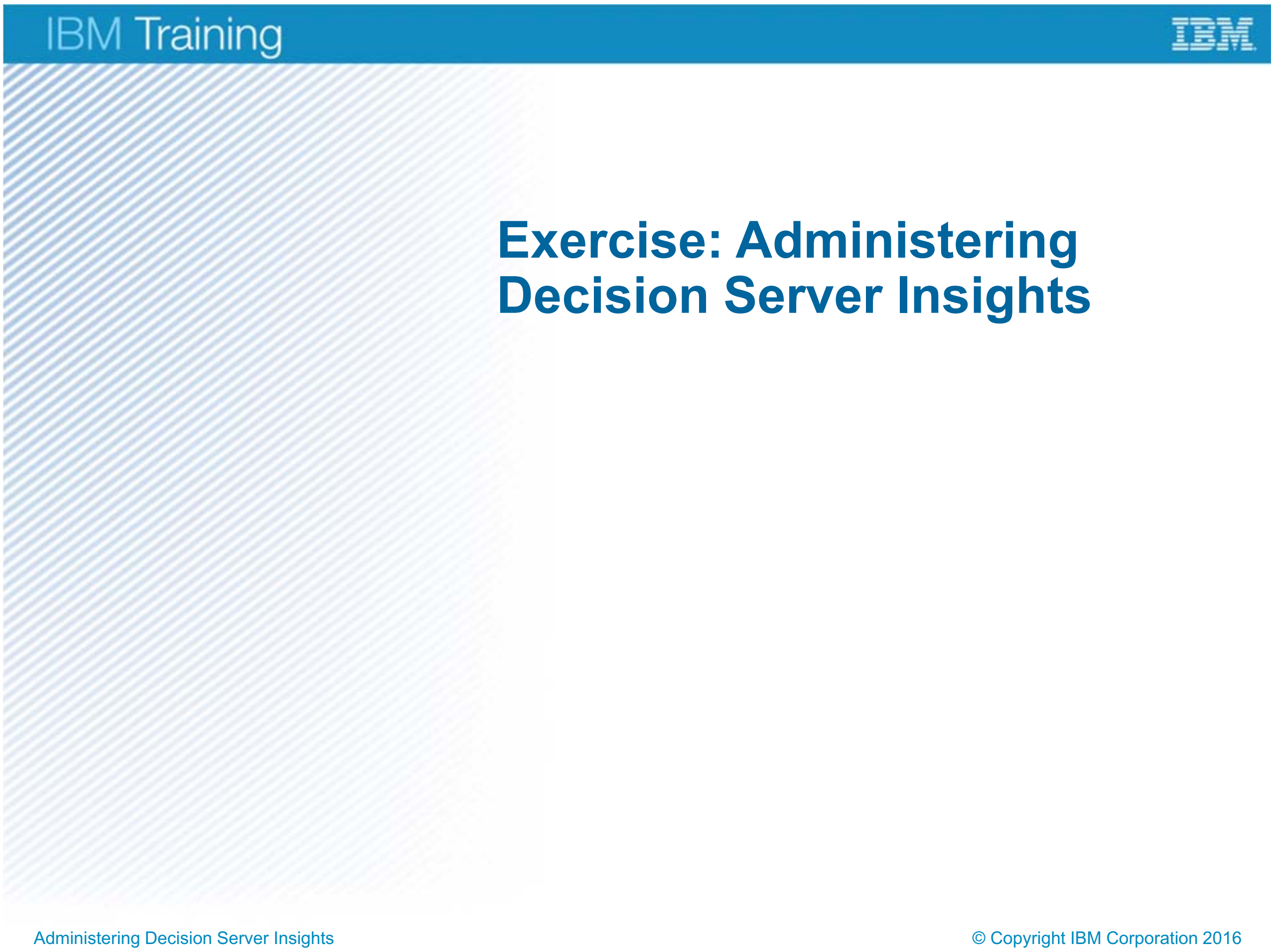


Figure 11-36. Exercise: Administering Decision Server Insights

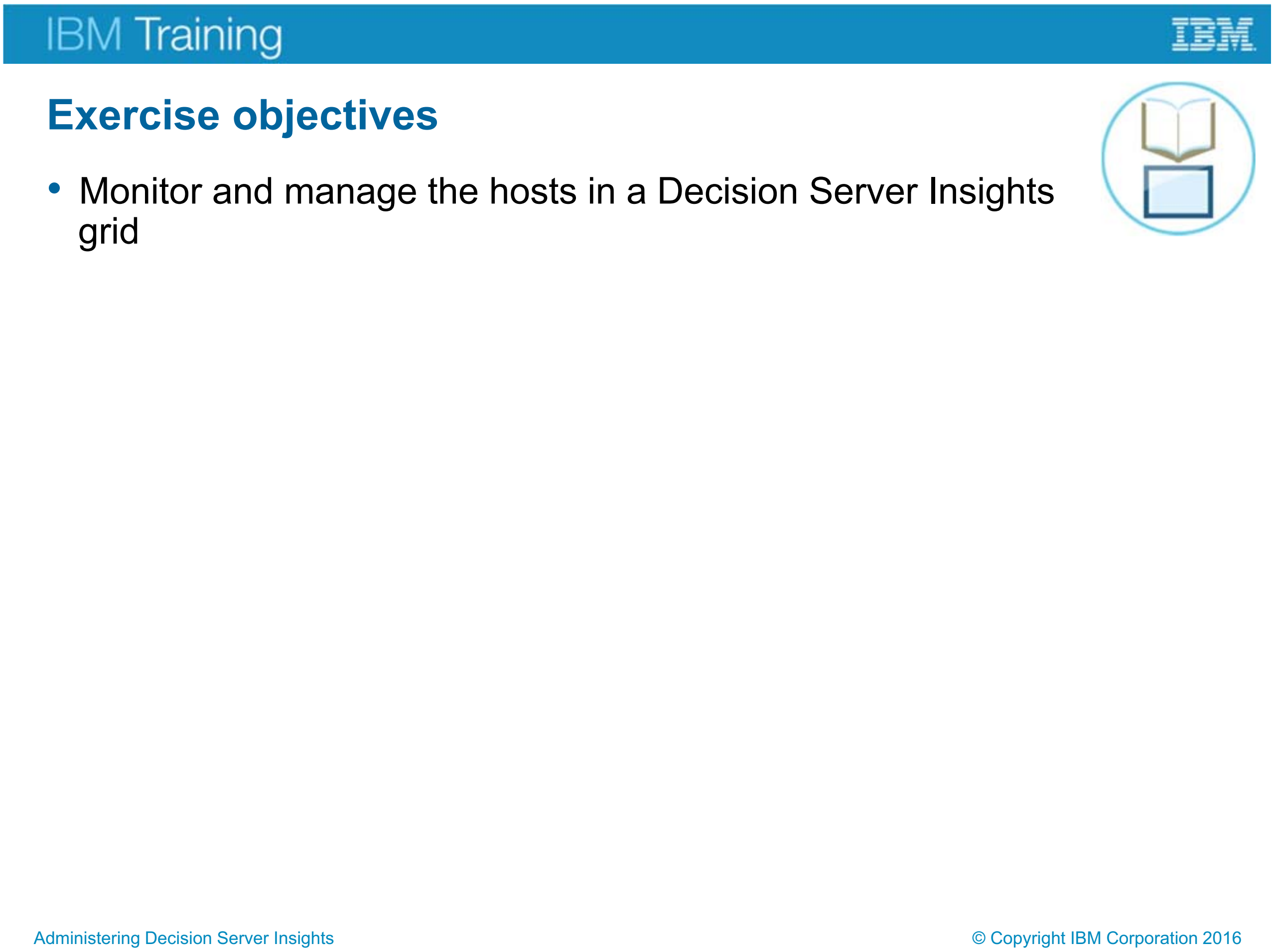


Figure 11-37. Exercise introduction

IBM Training

General administration utilities

- WebSphere eXtreme Scale monitoring
  - Use `xscmd` utility to view information about the grid
- Liberty profile server management
  - Use `server` and `serverManager` utilities to manage servers within your grid
  - Run these utilities from the <*InstallDir*>`/runtime/wlp/bin` directory
- Decision Server Insights solutions and connectivity
  - Use `solutionManager` and `connectivityManager`
  - Run these utilities from the <*InstallDir*>`/runtime/ia/bin` directory

Figure 11-38. General administration utilities

Unit 12. Course summary

Estimated time

00:30

Overview

This unit summarizes the course and provides information for future study.

IBM Training

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Course summary

Figure 12-1. Unit objectives

IBM Training

Course objectives

- Describe the Decision Server Insights programming model and architecture
- Design and create a Decision Server Insights solution
- Define the business model for the events, entities, and concepts that are relevant to your domain
- Use global aggregates for calculations across all events or a population of entities
- Implement business logic with rule agents and rules to detect and respond to business situations
- Deploy solutions to the Insight Server runtime and test runtime behavior
- Explain Decision Server Insights integration capabilities

Course summary

Figure 12-2. Course objectives

IBM Training

To learn more on the subject

- IBM Training website:
  www.ibm.com/training

- For more information about Decision Server Insights and decision management
  - Redbooks: *Systems of Insight for Digital Transformation: Using IBM Operational Decision Manager Advanced and Predictive Analytics*
    `http://www.redbooks.ibm.com/abstracts/sg248293.html?Open`
  - Redpaper: *Systems of Insight Overview*
    `http://www.redbooks.ibm.com/abstracts/redp5299.html?Open`
  - DeveloperWorks tutorial: *Install and configure a Decision Server Insights reference topology*
    `http://www.ibm.com/developerworks/bpm/bpmjournal/1503_defreitas1/1503_defreitas1.html`
  - Kolban's Book on IBM Decision Server Insights
    `http://neilkolban.com/ibm/decision-server-insights`

Course summary

Figure 12-3. To learn more on the subject

IBM Training

Earn an IBM Badge

- After completing this course, you might be ready to take an IBM Badge test
- Use IBM Badges to share verified proof of your IBM credentials
- Find your Badge test on this site:
  - https://www.ibm.com/services/learning/ites.wss/zz-en?pageType=badgesearch
- The *IBM Operational Decision Manager Advanced V8.8 Developer* Badge test requires these courses:
  - **WB395/ZB395**: *Developing Rule Solutions with IBM Operational Decision Manager V8.8*
  - **WB399/ZB399**: *Developing Solutions with IBM Decision Server Insights V8.8*
  - After completing **both** courses, take the Badge test

Course summary

Figure 12-4. Earn an IBM Badge

IBM Training

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
  - Product information
  - Training and certification
  - Documentation
  - Support
  - Technical information

- To learn more, see the IBM Cloud Education Resource Guide:
  - www.ibm.biz/CloudEduResources

Course summary

Figure 12-5. Enhance your learning with IBM resources

IBM Training

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Course summary

Figure 12-6. Unit summary

IBM Training

Course completion

You have completed this course:

Developing Solutions with IBM Decision Server Insights V8.8

Any questions?

Course summary

Figure 12-7. Course completion

Appendix A. List of abbreviations

ABRD	agile business rule development
API	application programming interface
ATM	automated teller machine
B2X	BOM-to-XOM mapping
BAL	Business Action Language
BEP	business event processing
BERL	Business Event Rule Language
BMD	business model definition
BOM	business object model
BPM	business process management
BPMN	Business Process Modeling Notation
BQL	Business Query Language
BRM	business rule management
BRMS	business rule management system
CEP	complex event processing
CICS	Customer Information Control System
CPU	central processing unit
CVS	Concurrent Versions System
Db	Database
DSI	Decision Server Insights
DVS	Decision Validation Services
DW	Decision Warehouse
EAR	enterprise archive
EE	Enterprise Edition (Java EE)
EJB	Enterprise JavaBeans
ERC	edition revision code
ESB	enterprise service bus
FFDC	First Failure Data Capture
GUI	Graphical User Interface
HMM	Hidden Markov Models

HTML	Hypertext Markup Language
HTDS	hosted transparent decision service
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDE	integrated development environment
IIB	IBM Information Bus
IP	Internet Protocol
IRL	ILOG Rule Language
IT	information technology
JAR	Java archive
JAXB	Java Architecture for XML Binding
JCA	Java EE Connector Architecture
JDK	Java Development Kit
JMS	Java Message Service
JMX	Java Management Extension
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JVM	Java virtual machine
KPI	key performance indicator
LAN	local area network
MBean	message bean
MDB	message-driven bean
MQ	message queue
MTDS	monitored transparent decision service
NPE	null pointer error
ODM	Operational Decision Manager
OSGi	Open Services Gateway Initiative
POJO	plain old Java object
QA	quality assurance
RAR	resource adapter archive
RES	Rule Execution Server
REST	Representational State Transfer
RMI	Remote Method Invocation

RQL	Rule Query Language
RSO	Rule Solutions for Office
SCA	Service Component Architecture
SCC	source code control
SDO	Service Data Object
SE	Standard Edition (Java SE)
SME	subject matter expert
SOA	service-oriented architecture
SOAP	Usage note: SOAP is not an acronym; it is a word in itself (formerly an acronym for Simple Object Access Protocol)
SPSS	Statistical Product and Service Solutions
SSP	Scenario Service Provider
TCP	Transmission Control Protocol
TRL	Technical Rule Language
UI	user interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
WADL	Web Application Description Language
WAN	wide area network
WAR	web archive
WDT	WebSphere Developer Tools
WSDL	Web Services Description Language
WSE	Workgroup Server Edition
WTDS	web transparent decision service
XML	Extensible Markup Language
XOM	execution object model
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformation.
XU	Execution Unit
z/OS	Z Series Operating System

Appendix B. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: `http://www.ibm.com/training`
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: `http://bit.ly/IBMTrainEN`
- **IBM Certification**
  - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
  - For more information, see: `http://www.ibm.com/certify`
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
  - For more information, see: `http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0003096`

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the latest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
  - Receive concise updates from Middleware Education a few times each week.

  - Follow Middleware Education at: `twitter.com/websphere_edu`
- **Facebook**:
  - Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
  - Find IBM Training at: `facebook.com/ibmtraining`
- **YouTube**:
  - See the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: `youtube.com/IBMTraining`

Support

- **Middleware Support portal**
  - The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARS). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
  - For more information, see: `http://www.ibm.com/software/websphere/support`
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: `http://www.ibm.com/software/support/isa`
- **IBM Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.
  - For more information, see: `http://www.ibm.com/software/info/education/assistant/`

Middleware documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.
  - For more information, see: `http://www.ibm.com/redbooks`
- **IBM documentation and libraries**
  - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
  - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
  - For more information, see: `http://www.ibm.com/developerworks`

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see:
  `http://www-935.ibm.com/services/us/en/it-services/systems/middleware-services/`

IBM Training