

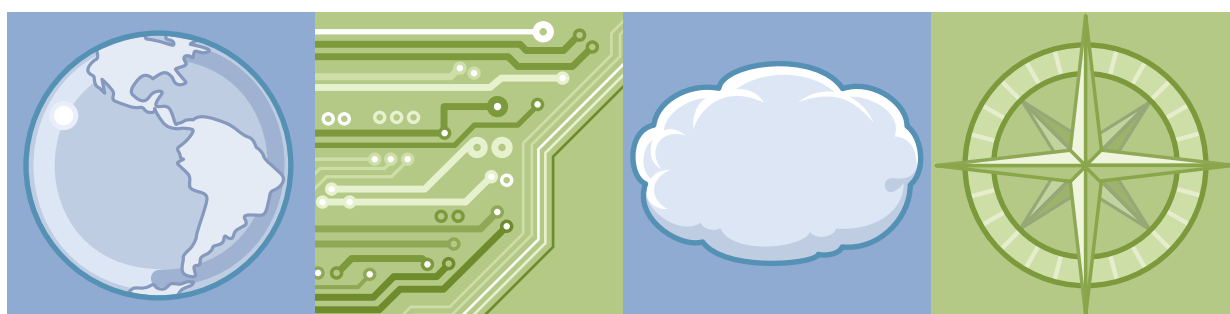


IBM Training

Student Notebook

Developing Solutions with IBM Decision Server Insights V8.7

Course code WB394 ERC 2.0



WebSphere Education

Trademarks

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

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

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

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

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

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

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

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

July 2015 edition

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

© Copyright International Business Machines Corporation 2015.

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

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

Contents

Trademarks	ix
Course description	xi
Agenda	xiii
Unit 1. Introducing IBM Decision Server Insights V8.7	1-1
Unit objectives	1-2
Topics	1-3
1.1. Overview of Decision Server Insights	1-5
Overview of Decision Server Insights	1-6
What is Decision Server Insights?	1-7
Four steps from data collection to intelligent business action	1-8
Sense what is happening	1-9
Build the context	1-10
Decide what to do	1-11
Act quickly and consistently	1-12
1.2. Decision Server Insights programming model	1-13
Decision Server Insights programming model	1-14
Core building blocks	1-15
Programming model	1-17
Analytic capabilities	1-18
1.3. Decision Server Insights architecture	1-19
Decision Server Insights architecture	1-20
IBM Operational Decision Manager Advanced offering	1-21
Decision Server Insights components	1-22
Decision Server Insights high-level architecture	1-24
Insight Server	1-26
1.4. Decision Server Insights roles	1-27
Decision Server Insights roles	1-28
Decision Server Insight roles	1-29
Unit summary	1-30
Checkpoint questions	1-31
Checkpoint answers	1-32
Exercise 1	1-33
Exercise objectives	1-34
Unit 2. Designing Decision Server Insights solutions	2-1
Unit objectives	2-2
Topics	2-3
2.1. Designing a solution	2-5
Designing a solution	2-6
Designing for a 360° view of your business entities	2-7
Solution diagrams	2-8
Agent diagrams	2-10
Choosing an agent type	2-11
Planning for event delivery	2-13
Understanding “time” in Decision Server Insights	2-14

Location-based reasoning	2-15
2.2. Planning for development and test environments	2-17
Planning for development and test environments	2-18
Setting up the development and test environments	2-19
Design concerns	2-20
Planning development and production environments	2-21
Planning development and production environments	2-22
2.3. Creating solutions	2-23
Creating solutions	2-24
Steps of building a Decision Server Insights solution	2-25
Getting started on an Insights solution	2-26
Decision Insight perspective	2-28
Setting up the Decision Insight perspective	2-29
Solution project folders and files	2-30
Solution Map view in Decision Insight perspective	2-32
Samples Console for Insights sample server	2-33
Unit summary	2-34
Checkpoint questions	2-35
Checkpoint answers	2-36
Exercise 2	2-37
Exercise objectives	2-38
Unit 3. Creating the business model.	3-1
Unit objectives	3-2
Topics	3-3
3.1. Overview of the business model	3-5
Overview of the business model	3-6
Modeling for a Decision Server Insights solution	3-7
Elements of a business model	3-8
Example entity model	3-9
Example event model with relationships	3-10
Example enumerations	3-11
Facets	3-12
3.2. Writing the business model	3-13
Writing the business model	3-14
Definition structure	3-15
Defining attributes	3-16
Entity definition	3-17
Event definition	3-18
Business model editor (1 of 3)	3-19
Business model editor (2 of 3)	3-20
Business model editor (3 of 3)	3-21
Unit summary	3-22
Checkpoint questions	3-23
Checkpoint answers	3-24
Exercise 3	3-25
Exercise objectives	3-26
Unit 4. Authoring the business logic	4-1
Unit objectives	4-2
Topics	4-3

4.1. Implementing business logic with agents	4-5
Implementing business logic with agents	4-6
Runtime architecture: Agents	4-7
Agents	4-8
Agents and entities	4-9
Agent types	4-10
Creating rule agents	4-11
Agent projects	4-12
Agent descriptors	4-13
Writing an agent descriptor	4-14
Example descriptor	4-15
Relationship between events and entities	4-16
4.2. Rule agents	4-17
Rule agents	4-18
Creating rule agents	4-19
Structure of an action rule	4-20
Defining the “when” part of a rule	4-21
Using “when” to postpone event processing	4-22
Example: Testing for the absence of an event	4-23
Defining rule variables	4-24
Calling global aggregate variables in rules	4-25
Defining conditions	4-26
Defining actions	4-27
Time-related tests	4-28
Location-related tests	4-29
4.3. Java agents and predictive scoring agents	4-31
Java agents and predictive scoring agents	4-32
Setting the target platform in Insight Designer	4-33
Java agents	4-34
Predictive agents	4-35
4.4. Global aggregates	4-37
Global aggregates	4-38
Runtime architecture: Global aggregates	4-39
What is a global aggregate	4-40
Types of global aggregate	4-41
Aggregates in the programming model	4-42
Global entity aggregate	4-43
Global event aggregates	4-44
Global aggregate functions	4-45
Defining aggregates	4-46
Retrieving global aggregate value	4-47
Unit summary	4-48
Checkpoint questions	4-49
Checkpoint answers	4-50
Exercise 4	4-51
Exercise objectives	4-52
Exercise 5	4-53
Exercise objectives	4-54
Exercise 6	4-55
Exercise objectives	4-56
Exercise 7	4-57

Exercise objectives	4-58
Exercise 8	4-59
Exercise objectives	4-60
Exercise 9	4-61
Exercise objectives	4-62
Unit 5. Deploying solutions	5-1
Unit objectives	5-2
Topics	5-3
5.1. Exporting and deploying	5-5
Exporting and deploying	5-6
Deploying from Insight Designer	5-7
Exporting solutions and agents	5-8
Version policy	5-9
Deploying solution archives	5-10
Deployed solutions files	5-11
Undeploying (1 of 2)	5-12
Undeploying (2 of 2)	5-13
Deleting solution files	5-14
Unit summary	5-15
Checkpoint questions	5-16
Checkpoint answers	5-17
Exercise 10	5-18
Exercise objectives	5-19
Unit 6. Testing solutions	6-1
Unit objectives	6-2
Topics	6-3
6.1. TestDriver API	6-5
TestDriver API	6-6
TestDriver API	6-7
TestDriver properties	6-8
Example test driver properties	6-9
Insert entities and submit events (1 of 2)	6-10
Insert entities and submit events (2 of 2)	6-11
Receiving and storing debug information (1 of 2)	6-12
Receiving and storing debug information (2 of 2)	6-13
6.2. REST API	6-15
REST API	6-16
REST methods	6-17
REST resources	6-18
6.3. Log analysis	6-19
Log analysis	6-20
Why are log files important?	6-21
Where and what to gather	6-22
Log analysis	6-23
6.4. Insight Inspector	6-25
Insight Inspector	6-26
A debugging tool	6-27
Web UI	6-28
Capabilities	6-29

How it works	6-30
6.5. Generic Client	6-31
Generic Client	6-32
Highlights	6-33
Plugging the generic client to your own solution (1 of 3)	6-34
Plugging the generic client to your own solution (2 of 3)	6-35
Plugging the generic client to your own solution (3 of 3)	6-36
Unit summary	6-37
Checkpoint questions	6-38
Checkpoint answers	6-39
Unit 7. Modeling and defining connectivity	7-1
Unit objectives	7-2
Topics	7-3
7.1. Event delivery	7-5
Event delivery	7-6
High-level event delivery architecture	7-7
Connectivity components	7-8
7.2. Modeling connectivity in your solution	7-9
Modeling connectivity in your solution	7-10
Event delivery: Inbound connectivity	7-11
Solution gateway for inbound connectivity	7-12
Event delivery: Outbound connectivity	7-13
7.3. Defining connectivity for your solution	7-15
Defining connectivity for your solution	7-16
Defining connectivity in the solution project	7-17
Define connectivity in your solution	7-19
Export solution connectivity server configuration	7-20
Create the connectivity definitions	7-21
Export solution connectivity server configuration	7-23
Create the connectivity definitions	7-24
Providers for JMS	7-25
7.4. Deploying connectivity in your solution	7-27
Deploying connectivity in your solution	7-28
Deploying a connectivity configuration	7-29
Transformations	7-30
Classifying inbound messages for transformation	7-31
Troubleshooting	7-32
Common issues and resources	7-33
Unit summary	7-34
Checkpoint questions	7-35
Checkpoint answers	7-36
Exercise 11	7-37
Exercise objectives	7-38
Unit 8. Integrating Decision Server Insights	8-1
Unit objectives	8-2
Topics	8-3
8.1. Overview of integration requirements	8-5
Overview of integration requirements	8-6
No title	8-7

Rolling out a Decision Server Insights solution today	8-8
8.2. Submitting events from an Integration Bus message flow	8-9
Submitting events from an Integration Bus message flow	8-10
Submitting events from an Integration Bus message flow	8-11
Events model	8-12
Java Compute Node	8-13
Deploying IBM Integration Bus message flows	8-14
8.3. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights	8-15
Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights	8-16
Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights	8-17
Step 1	8-18
Step 2	8-19
Step 3	8-20
Step 4	8-21
Step 5	8-22
Summary of the events to be imported	8-23
The result	8-24
8.4. Integration with SPSS	8-25
Integration with SPSS	8-26
SPSS	8-27
Predictive analytics	8-28
How it works	8-29
Creating a predictive scoring agent	8-30
Troubleshooting	8-31
Suggested practices	8-32
8.5. OSGi services	8-33
OSGi services	8-34
OSGi Services	8-35
Unit summary	8-36
Checkpoint questions	8-37
Checkpoint answers	8-38
Unit 9. Course summary	9-1
Unit objectives	9-2
Course learning objectives	9-3
To learn more on the subject (1 of 2)	9-4
To learn more on the subject (2 of 2)	9-5
Unit summary	9-6
Appendix A. List of abbreviations	A-1
Appendix B. Resource guide	B-1

Trademarks

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

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

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

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

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

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

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

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

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

Course description

Developing Solutions with IBM Decision Server Insights V8.7

Duration: 3 days

Purpose

This course teaches you the main features of the Decision Server Insights component of IBM Operational Decision Manager Advanced V8.7. 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 Decision Server Insight's multi-agent architecture by developing several agents that are bound to a single entity for different purposes.

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 WebSphere Application Server Liberty profile

Objectives

After completing this course, you should be able to:

- 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 V8.7, which is covered in the course: WB392, Developing Rule Solutions in IBM Operational Decision Manager V8.7.

Agenda

Day 1

Course introduction
Unit 1. Introducing IBM Decision Server Insights V8.7
Exercise 1. Getting started with Decision Server Insights
Unit 2. Designing Decision Server Insights solutions
Exercise 2. Creating a solution in Insight Designer
Unit 3. Creating the business model
Exercise 3. Defining the business model
Unit 4. Authoring the business logic (part 1)

Day 2

Unit 4. Authoring the business logic (part 2)
Exercise 4. Creating a rule agent
Exercise 5. Writing and testing rules
Exercise 6. Using global aggregates in rules
Exercise 7. Using event aggregates in rules
Exercise 8. Using time-based and location-based reasoning in rules
Exercise 9. Testing for the absence of event

Day 3

Unit 5. Deploying solutions
Exercise 10. Deploying solutions
Unit 6. Testing solutions
Unit 7. Modeling and defining connectivity
Exercise 11. Defining connectivity for a solution
Unit 8. Integrating Decision Server Insights
Unit 9. Course summary

Unit 1. Introducing IBM Decision Server Insights V8.7

What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

- Describe Decision Server Insights and explain how it works
- Introduce key terminology, including *entity*, *event*, *agent*, and *global aggregate*
- Describe the Decision Server Insights architecture
- Outline the user roles that are associated with Decision Server Insights

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Describe Decision Server Insights and explain how it works
- Introduce key terminology, including entity, event, agent, and global aggregate
- Describe the Decision Server Insights architecture
- Outline the user roles that are associated with Decision Server Insights

© Copyright IBM Corporation 2015

Figure 1-1. Unit objectives

WB3942.0

Notes:

Topics

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

© Copyright IBM Corporation 2015

Figure 1-2. Topics

WB3942.0

Notes:

1.1. Overview of Decision Server Insights

Overview of Decision Server Insights



© Copyright IBM Corporation 2015

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

9.1

Figure 1-3. Overview of Decision Server Insights

WB3942.0

Notes:

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
- Model-driven, scalable, complex 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

© Copyright IBM Corporation 2015

Figure 1-4. What is Decision Server Insights?

WB3942.0

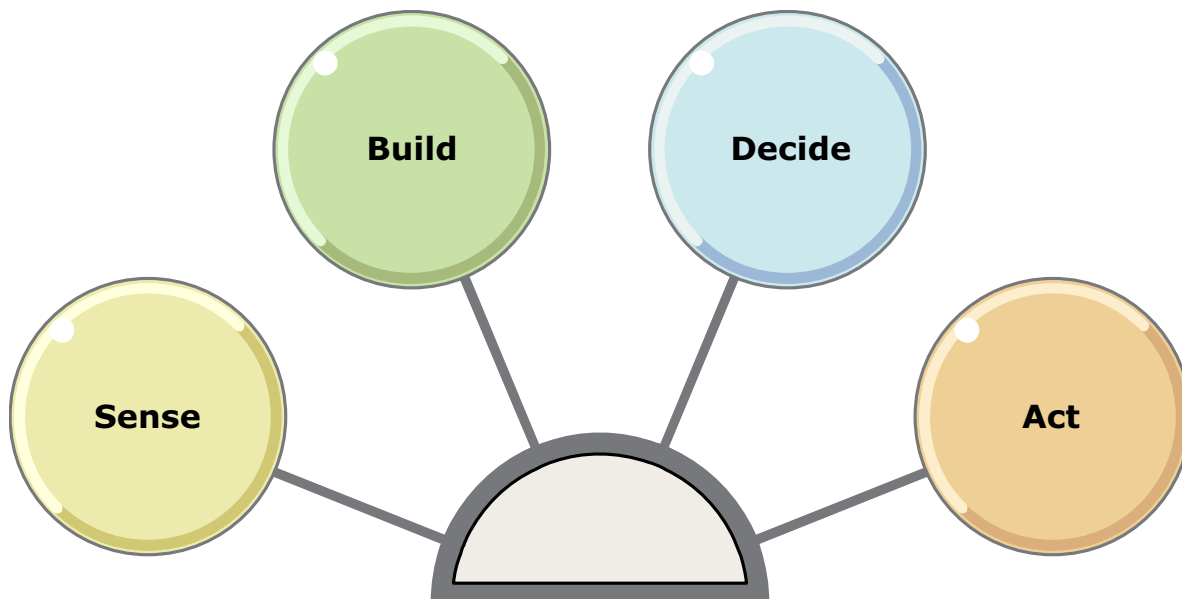
Notes:

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

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



© Copyright IBM Corporation 2015

Figure 1-5. Four steps from data collection to intelligent business action

WB3942.0

Notes:

In the interconnected, mobile-focused, cloud-enabled world of today, the amount of data is growing exponentially. Data is now the new natural resource. But data collection is merely a means to an end. The goal is to gain insight from the data that is collected in real time, so you can act and improve business results.

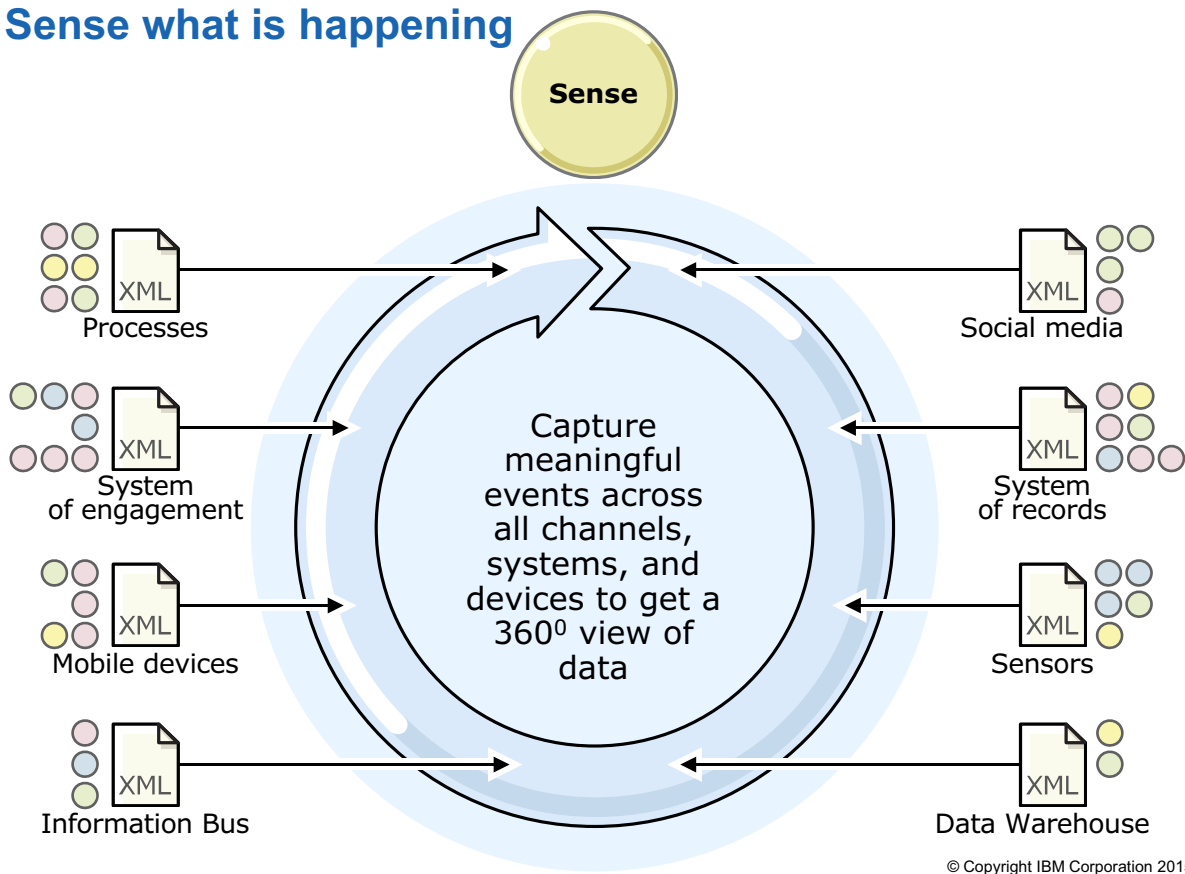
Decision Server Insights helps you to interact with your customers at the most advantageous time and place to optimize their experience and advance your interests.

Success depends on effectively capturing diversity in decisions and adapting to unanticipated changes at the pace of your business.

This real-time actionable insight is a four-step process:

1. Sense what is happening.
2. Build your context.
3. Decide what to do.
4. Act quickly and consistently.

Sense what is happening



© Copyright IBM Corporation 2015

Figure 1-6. Sense what is happening

WB3942.0

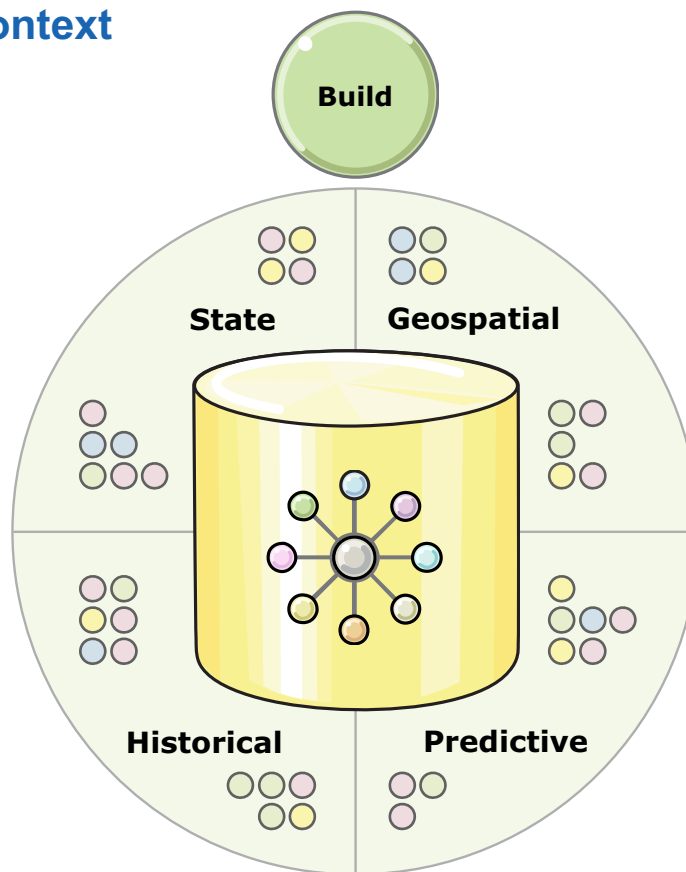
Notes:

First, you must sense, or receive, the events from the outside world. The Decision Server Insights runtime environment can listen to a wide range of event providers, as depicted here. Events are sent into the product as XML messages.

To detect interesting situations, you must correlate the events. 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.

Events are received as XML messages over a JMS queue. They can be posted by HTTP in a RESTful manner or submitted programmatically through the Java API.

Build the context



© Copyright IBM Corporation 2015

Figure 1-7. Build the context

WB3942.0

Notes:

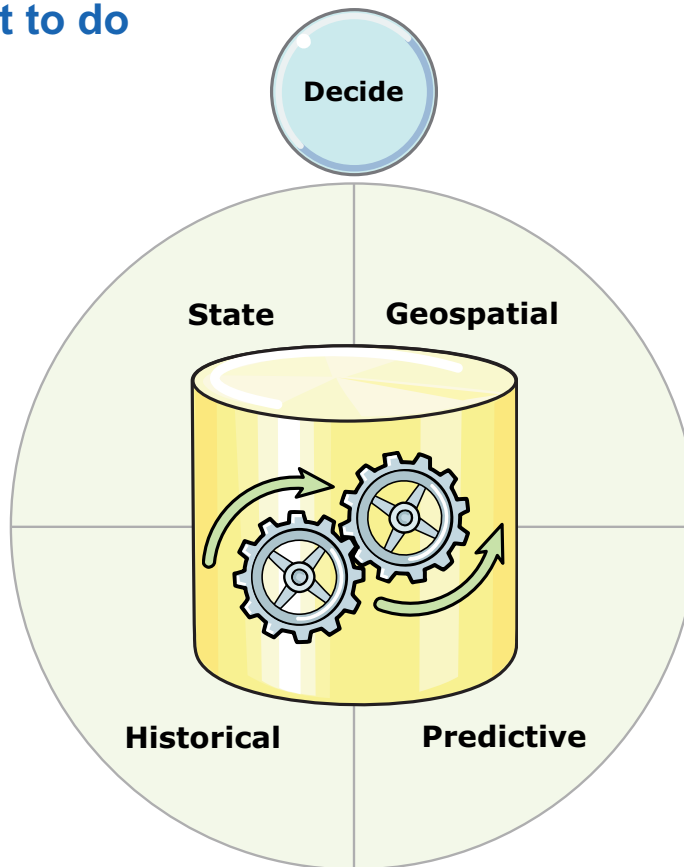
Building the context involves putting data and events into context to understand and evaluate correlations.

As events occur that involve business entities that you care about, like 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 a client to defect.

Decision Server Insights provides **global analytics** that you can use to extract valuable insights over entire populations of business entities in near real-time. You can use them to further optimize your decision models. You use global analytics to identify *outliers*, such as the clients with the highest risk of defecting, and take special action for them.

Decide what to do



© Copyright IBM Corporation 2015

Figure 1-8. Decide what to do

WB3942.0

Notes:

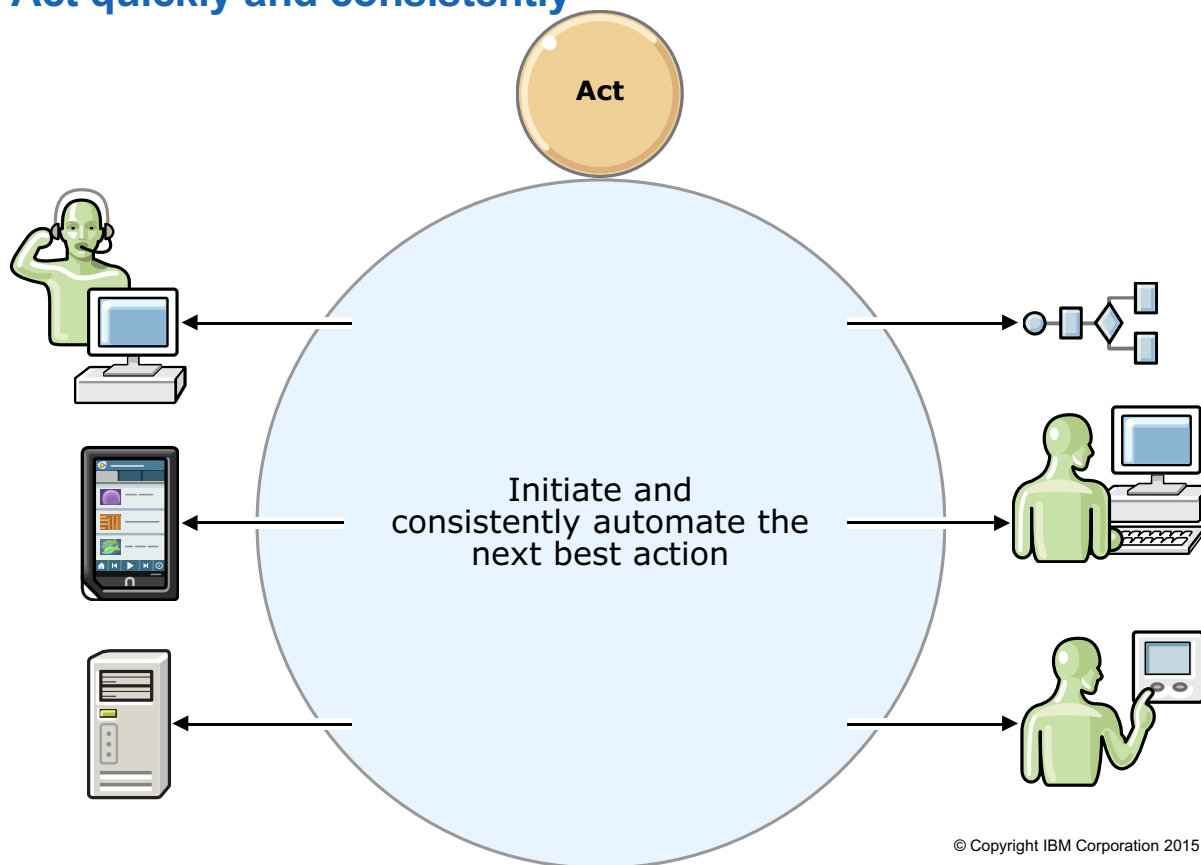
Decision Server Insights agents implement the business logic to apply the models, policies, and suggested practices that your subject-matter experts establish. Rules are expressed as situation detection patterns. When you identify a situation that is relevant to your business, you must decide on the next best action.

ODM Advanced generalizes the use of **natural language** technology to all aspects of decision modeling. It fosters the active participation of subject matter experts, business analysts, and data scientists in all activities that are associated with the design, implementation, and management of a proactive decision service.

Decision Server Insights provides **time and spatial modeling, reasoning**, and **analytics** to help you detect and respond to the most intricate patterns and trends.

You can use **global analytics** through aggregates to extract valuable insights over populations of business entities in near real-time. You can use these insights to further optimize your decision models.

Act quickly and consistently



© Copyright IBM Corporation 2015

Figure 1-9. Act quickly and consistently

WB3942.0

Notes:

Decision Server Insights emits a response event whenever an incoming event, in combination with past events, matches a rule.

The response event is published as an XML message on the outbound queue, or it is posted to HTTP as a URL for some external system to pick up that event. The event can be transformed into something that the external system can use to interact with 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

Decision Server Insights programming model



© Copyright IBM Corporation 2015

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

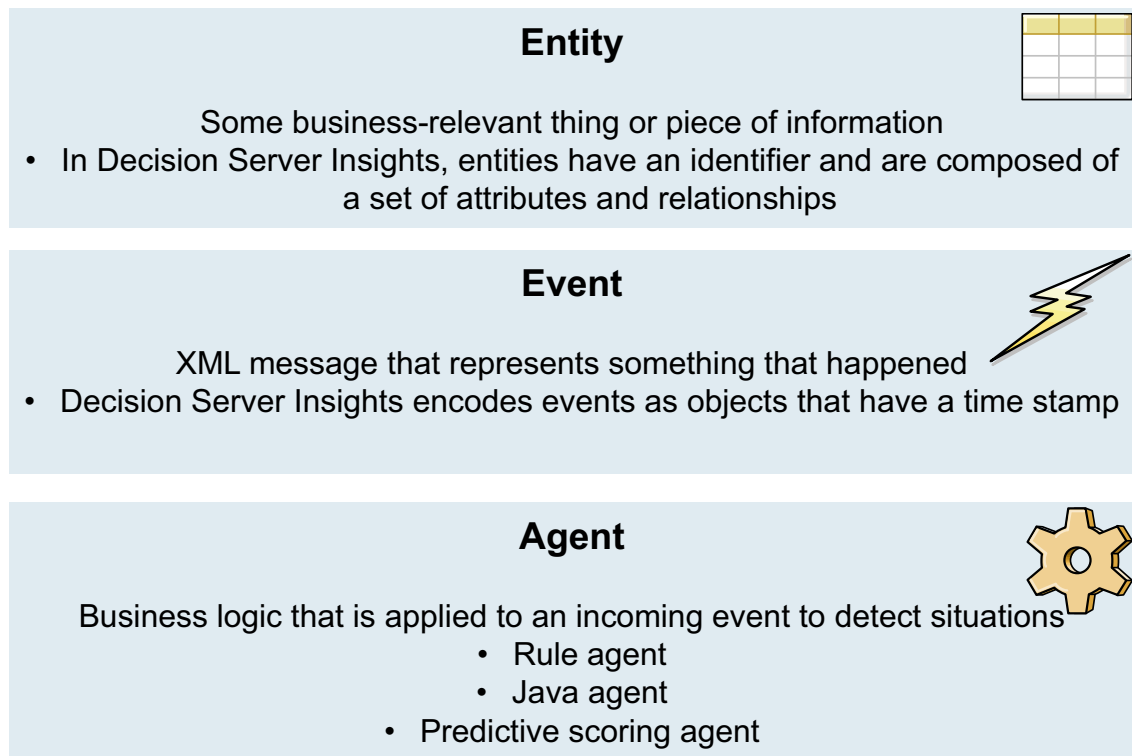
9.1

Figure 1-10. Decision Server Insights programming model

WB3942.0

Notes:

Core building blocks



© Copyright IBM Corporation 2015

Figure 1-11. Core building blocks

WB3942.0

Notes:

Entities are concepts that are relevant to your business and have an identifier. Entities are composed of a set of properties and relationships to other concepts, and can inherit from existing entities. For example, if you define a Person entity, you might also define a Customer entity or an Employee entity that inherit common properties from the Person entity.

An event is something that happens or can happen. An event relates to one or more entities. For example, a withdrawal event relates to an ATM entity and to a client entity. From a design point of view, an event is a specific action or measurement that occurred at a specific time and place. Events must be defined in the business model and processed before any action can take place. In Decision Server Insights, every event is assigned a time stamp.

Agents in Decision Server Insights define the business logic that is used to detect situations of interest. An agent is bound to an entity and subscribes to the events that might be involved in a business situation of interest. When an agent detects a business situation for its entity, it reacts.

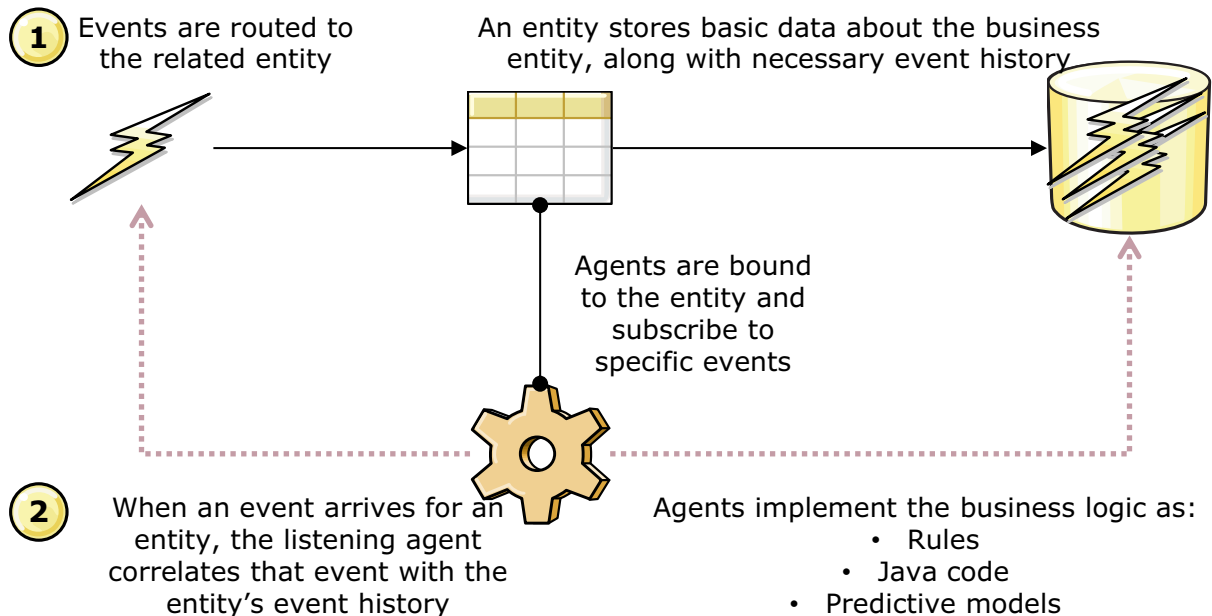
Types of agents include:

- Rule agents**, which are written with natural language (not code) so that business stakeholders can understand and validate them. Rule agents automatically manage the state for the

accumulated events. A rule agent can include rules that query, filter, and correlate events from the past and use them to detect situations of interest.

- **Java agents**, which are written in Java and are stateless by default. To maintain a state with Java agents, you must add some attributes to the entity so that the state is maintained in the entity.
- **Predictive scoring agents** are an extension of the Java agent, with a built-in API for invoking an SPSS Scoring Configuration.

Programming model



© Copyright IBM Corporation 2015

Figure 1-12. Programming model

WB3942.0

Notes:

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 can 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 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. These new events can be either internal to the solution, to trigger other event-entity-agent bindings; or external to the solution, to trigger system actions.

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

© Copyright IBM Corporation 2015

Figure 1-13. Analytic capabilities

WB3942.0

Notes:

A global aggregate is a value that is computed from a collection of objects. Decision 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 entities regularly (typically, every day).

For example, you can measure the average age of all customers or determine how many “change of address” events were received last week. This type of measurement can help you find entities or events that do not match the average. For example, if you receive an average from an aggregate, but have a customer that is below average, then you can take some action for this customer.

Decision Server Insights is not only reactive. To get ahead of the events, you can use SPSS predictive models.

Decision Server Insights manages the aggregates, and inputs aggregates data to model in SPSS. From SPSS, you receive a likelihood score, which you can use to trigger the prescribed next best action.

1.3. Decision Server Insights architecture

Decision Server Insights architecture



© Copyright IBM Corporation 2015

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

9.1

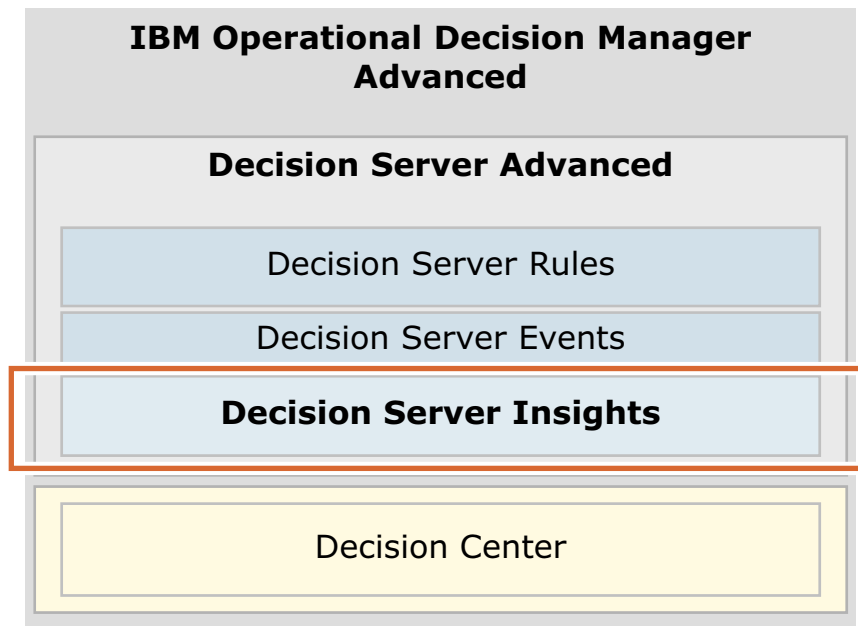
Figure 1-14. Decision Server Insights architecture

WB3942.0

Notes:

IBM Operational Decision Manager Advanced offering

- Decision Server Insights is a module in IBM Operational Decision Manager Advanced V8.7



© Copyright IBM Corporation 2015

Figure 1-15. IBM Operational Decision Manager Advanced offering

WB3942.0

Notes:

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

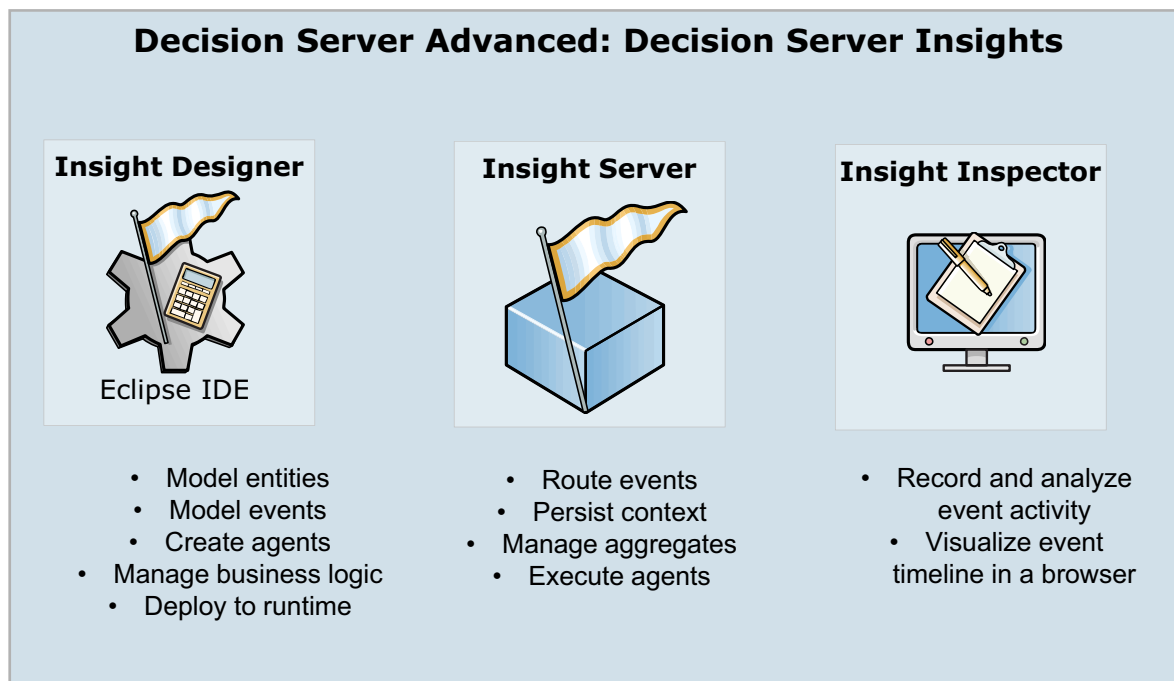


Note

When you install ODM Advanced, you cannot install Decision Server Insights to the same installation directory due to shell sharing limitations.

IBM ODM is also available as Standard and Express offerings, but without Decision Server Insights. For more information about the IBM ODM packaging, see the product documentation.

Decision Server Insights components



© Copyright IBM Corporation 2015

Figure 1-16. Decision Server Insights components

WB3942.0

Notes:

Decision Server Insights has a similar structure as ODM Decision Server Rules, but has no interaction with Decision Center.

Decision Server Insights includes:

- Insight Designer: A development environment in Eclipse.
- Insight Server: A runtime environment that handles complex event processing and agent execution.

Monitoring tools are also available, including Insight Inspector, which is a browser-based tool for visualizing event activity.

Insight Designer

- Includes an Eclipse interface to develop rule-based, event-driven solutions.
- Use to develop solutions that capture business models and logic through natural-language editors.
- Solutions route events to entities through agents or services and use business rules to process responses.

- Solutions include model definition, business rules, and analytics.
- Connectivity definitions determine inbound and outbound endpoints to receive and send events between solution and external systems.

The Insight Server runtime is based on WebSphere Liberty and Extreme Scale.

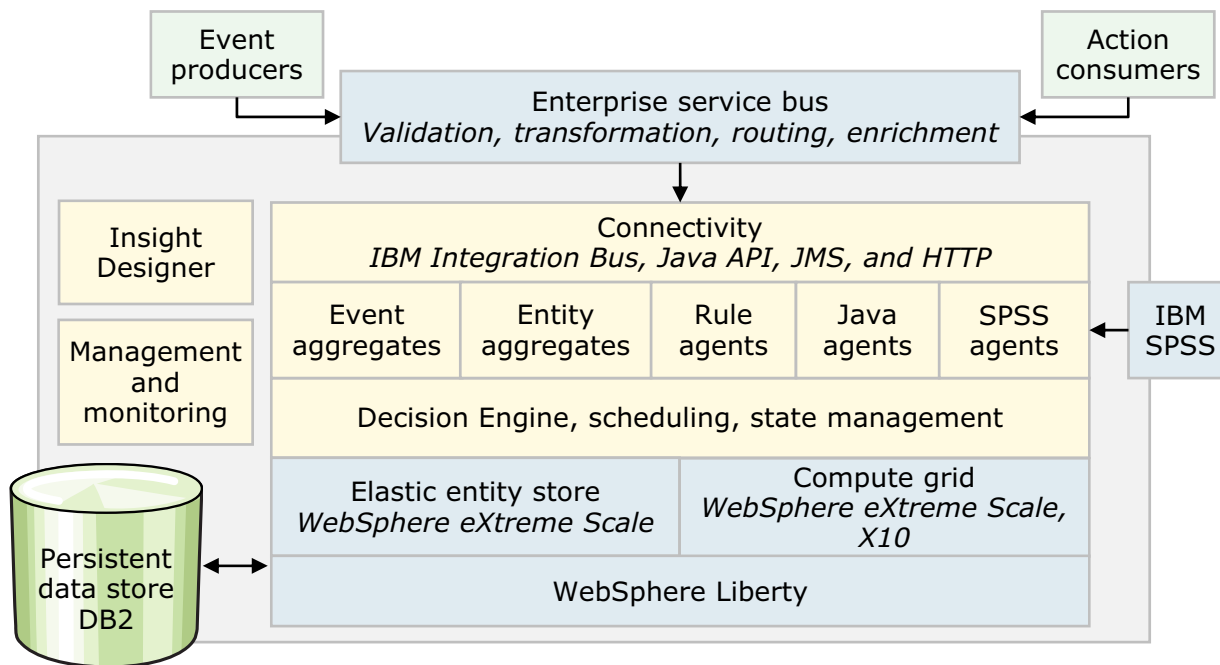
- Elastic and scalable in-memory compute and data grid
- Maintains stateful context of business entities
- Applies event-processing logic at the time of interaction

Insights Inspector

- Visualize a timeline of event activity for entities.

Decision Server Insights high-level architecture

- Integrates business rules, events, and predictive analytics capabilities in a single platform



© Copyright IBM Corporation 2015

Figure 1-17. Decision Server Insights high-level architecture

WB3942.0

Notes:

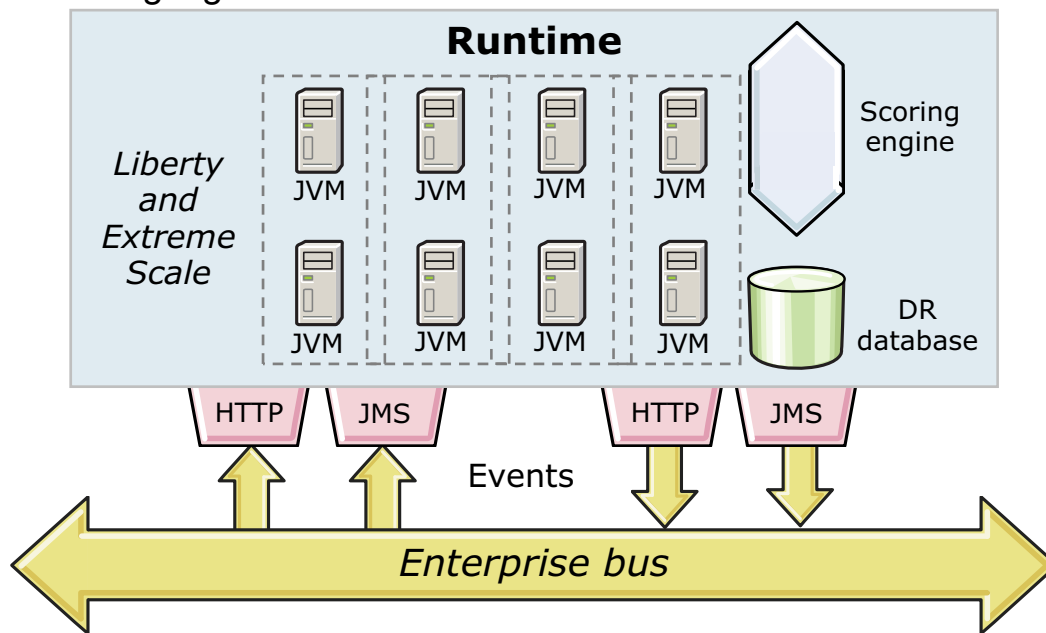
Insight Server is a clustered, modular, and distributed runtime that uses the WebSphere Liberty profile as an OSGi microkernel. WebSphere eXtreme Scale provides an elastic and scalable in-memory compute and data grid. WebSphere eXtreme Scale is used as a data grid for high-performance access to data, and it 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.

- Solutions are deployed from Insight Designer to the runtime.
 - The runtime receives events through the connectivity tier. Then, the solution gateway component routes the events to the agents that are interested (through subscription criteria for agents as defined in agent descriptors). The gateway is aware of the agents, and their descriptors so it routes events to the appropriate agents. Routing is done at the level of the gateway, consolidating the descriptors of each agent.
 - The runtime also maintains stateful context for the business entities and applies event-processing logic at the time of interaction. Context is a correlation of all historical events for an entity plus the current state of the entity, plus predictive scores that are associated with the entity.

- Global aggregates run calculations on all events or across a population of entities that the runtime manages.
- Decision Server Insights simplified much of the integration steps with the SPSS environments.
- Management is done by a set of JMX MBeans and RESTful web services.
 - Use IBM SmartCloud Analytics Embedded to run diagnostic tests, check logs, and generate analytical charts from real-time data.
- Persistence for disaster recovery.
- An event bus is not part of the product, but it is a good prerequisite for facilitating integration by having one place where all the events are published.

Insight Server

- Efficient, elastic, scalable, in-memory compute and data grid
- Maintains stateful context of business entities and applies event-processing logic at the time of interaction



© Copyright IBM Corporation 2015

Figure 1-18. Insight Server

WB3942.0

Notes:

The runtime is a clustered, modular, and distributed that uses the Liberty profile as an OSGi microkernel. The modular architecture of the runtime makes it scalable. You can run Insight Server with a single JVM, or, with potentially hundreds of servers in a large data center.

WebSphere eXtreme Scale provides a grid for high-performance data access and computation.

1.4. Decision Server Insights roles

Decision Server Insights roles



© Copyright IBM Corporation 2015

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





9.1

Figure 1-19. Decision Server Insights roles

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 1-20. Decision Server Insight roles

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Describe Decision Server Insights and explain how it works
- Introduce key terminology, including entity, event, agent, and global aggregate
- Describe the Decision Server Insights architecture
- Outline the user roles that are associated with Decision Server Insights

© Copyright IBM Corporation 2015

Figure 1-21. Unit summary

WB3942.0

Notes:

Checkpoint questions

1. True or false: Global entity aggregates provide the ability to run calculations across your entire population of entities or a subset of those entities.
2. The development component of Decision Server Insights is an Eclipse plug-in that is called
 - a. Insight Studio
 - b. Insight Designer
 - c. Event Designer
3. 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

© Copyright IBM Corporation 2015

Figure 1-22. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

Checkpoint answers

1. 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.

2. 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.

3. 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

Answer: All (Rule Agent, Java Agent, Predictive Scoring Agent).

© Copyright IBM Corporation 2015

Figure 1-23. Checkpoint answers

WB3942.0

Notes:

Exercise 1

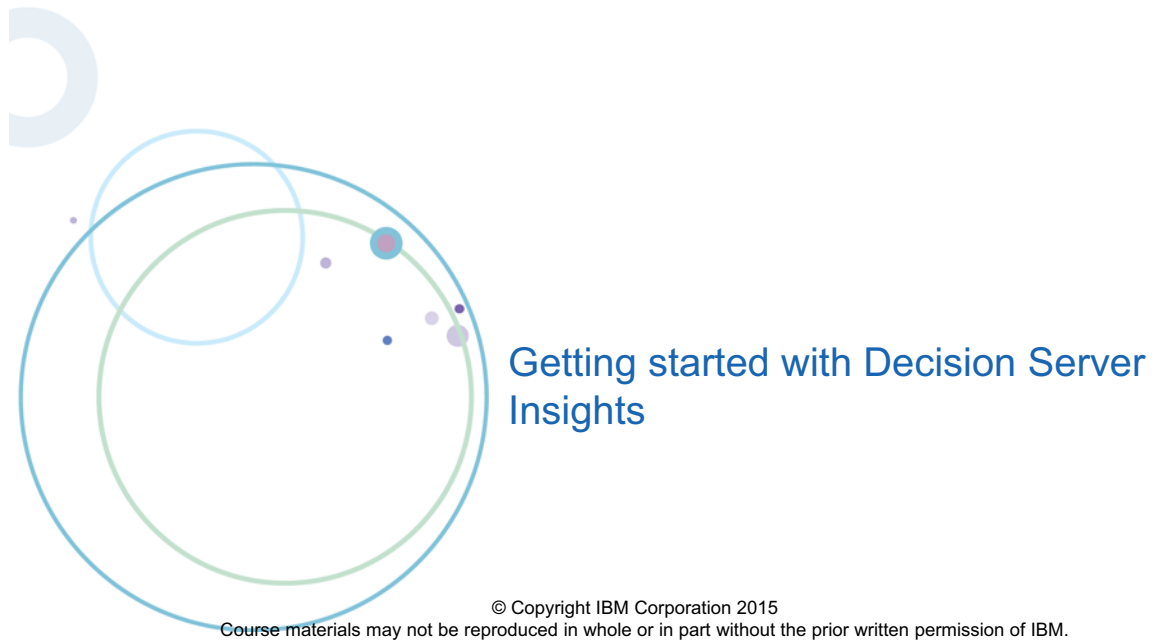


Figure 1-24. Exercise 1

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 1-25. Exercise objectives

WB3942.0

Notes:

Unit 2. Designing Decision Server Insights solutions

What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

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

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 2-1. Unit objectives

WB3942.0

Notes:

Topics

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

© Copyright IBM Corporation 2015

Figure 2-2. Topics

WB3942.0

Notes:

2.1. Designing a solution

Designing a solution



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

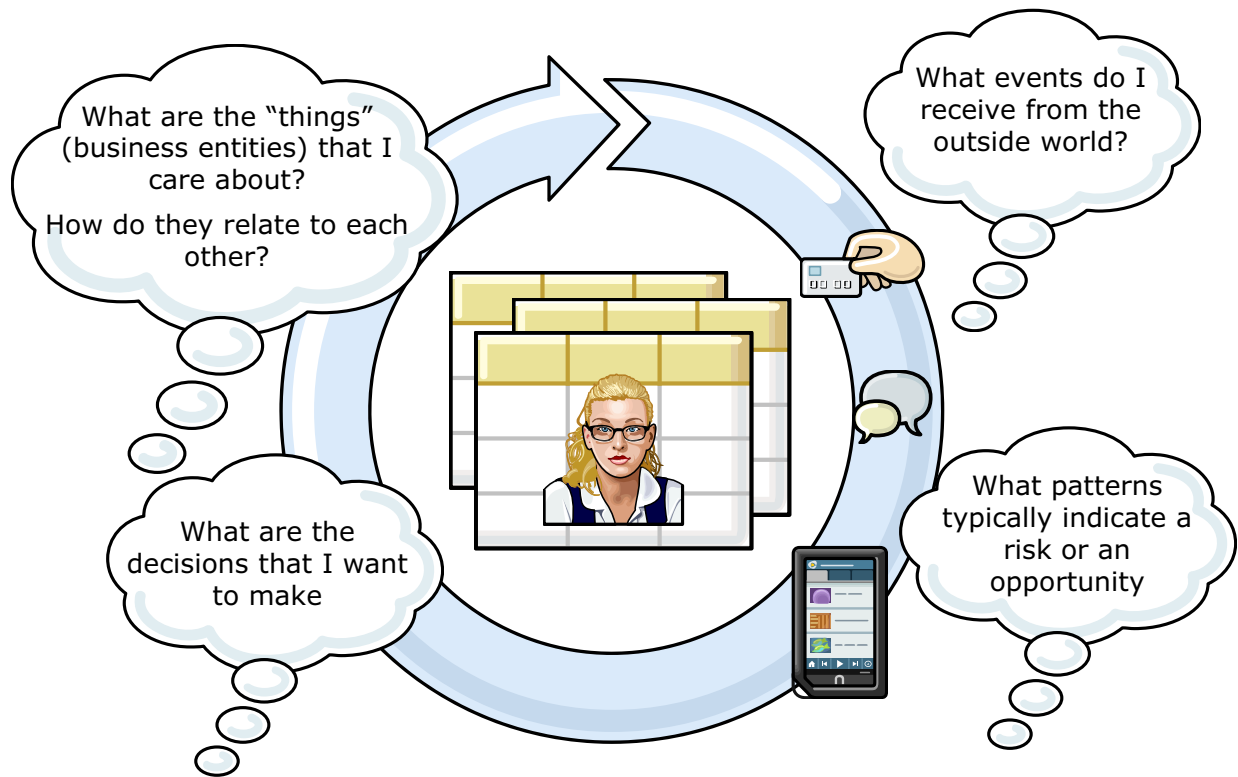
9.1

Figure 2-3. Designing a solution

WB3942.0

Notes:

Designing for a 360° view of your business entities



© Copyright IBM Corporation 2015

Figure 2-4. Designing for a 360° view of your business entities

WB3942.0

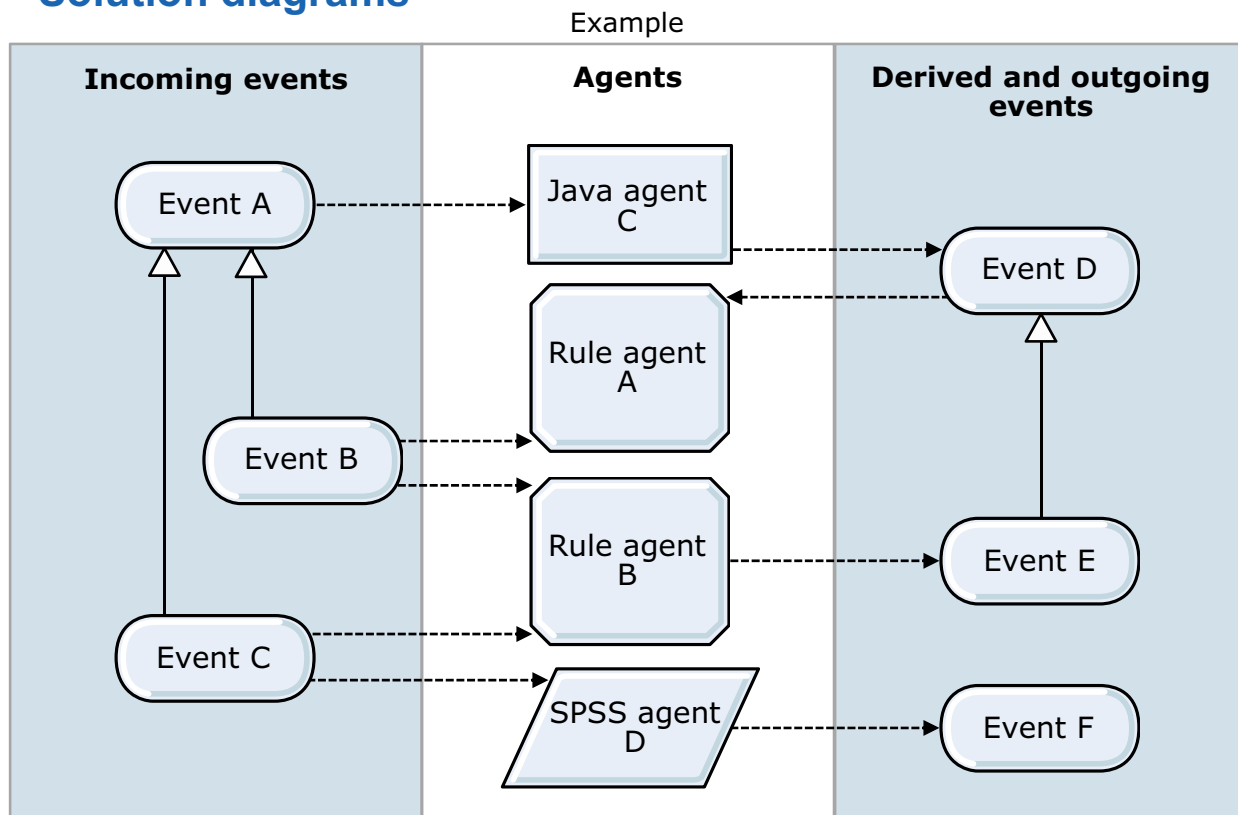
Notes:

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 and which events and event patterns are interesting.

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.

Solution diagrams



© Copyright IBM Corporation 2015

Figure 2-5. Solution diagrams

WB3942.0

Notes:

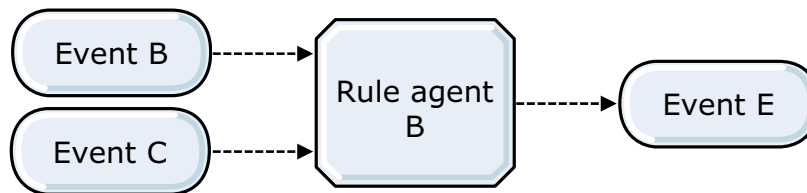
Solution Diagrams are composed of the following elements:

- A heading. The first line of the heading is the name of the solution in 16 point font. The second line of the heading is Version plus the version number in 12 point font.
- Three vertical columns, which are divided by vertical separators:
 - Column 1: Incoming Events: Incoming events (received from outside world)
 - Column 2: Agents: Agents that are defined by the solution
Suggested practice is for the name of the Agent to indicate the bound entity of the agent.
Example: Customer Agent would be bound to the Customer entity.
 - Column 3: Derived and Outgoing Events: Derived events and events that are emitted to outside world
- Rounded rectangles: Events
- Solid lines with arrows: Inheritance relationships between events
- Dashed lines with arrows: Agent event subscription or emission

- White chamfered rectangles: Rule agents
- Dark gray rectangles: Java agents
- Light gray parallelograms: SPSS agents

Events that are emitted by agents and optionally sent to the outside world are in the third column.

Agent diagrams



© Copyright IBM Corporation 2015

Figure 2-6. Agent diagrams

WB3942.0

Notes:

The Agent Diagram is used to focus on a single agent, including the event that it subscribes to, and the events that it emits. The diagram places the subscribed events on the left of the diagram and the emitted events on the right of the diagram. Do not indicate inheritance relationships between events on the diagram.

The diagram is typically generated by selecting elements from the solution overview diagram and pasting them into a new diagram. The events are rearranged so that all the subscribed events are on the left and the emitted events are on the right.

To ensure transactional integrity and performance, the programming model must target updates to data that is local to the bound entity.

Before you define relationships between entities, you must think of the lifecycle of each of these entities to avoid testing entities that do not exist. The value of an entity is null if it is not created or if it is destroyed. You must remove the references to the bound entity before the agent destroys it. After the entity is destroyed, it is no longer possible to remove the references to this entity.

The relationships between entities are defined in the business model. The binding from the agent to the entity is done at the agent level, in the agent descriptor.

Choosing an agent type

- 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 guideline can also be used to make your choice of agent:
 - Use a rule agent if you need event aggregation in your solution 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

© Copyright IBM Corporation 2015

Figure 2-7. Choosing an agent type

WB3942.0

Notes:

Rule agents

- Make decisions
- React to the arrival of a new event, such as a withdrawal in New York
- Apply business rules that SMEs write to recognize particular combinations of events and entity state, such as a sequence of withdrawals in different cities within the same day

Predictive agents

- Invoke self-trained SPSS models and update predictive scores

Java agents

- Typically used for pre-processing incoming events and post-processing outgoing events
- Can be used to extract keywords from incoming emails
- Represent in a given human language a message with an outgoing message with a symbolic code

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.

Planning for event delivery

- Inbound events
 - The solution receives events from external endpoints through the solution gateway
 - You define inbound binding for events from an external endpoint
 - Inbound binding identifies the format and protocol of inbound messages, and must reference an inbound endpoint
 - Inbound endpoints represent the external origin of event messages
- Outbound or generated events tell the outside world that a particular action must be performed, such as calling security or displaying a message
 - The solution can send events as a result of the agent execution
 - Outbound binding determines which outbound events are sent, and determines the message format and protocol to be used
 - Outbound binding must reference an outbound endpoint
 - Outbound endpoints represent the destination (either a JMS connection factory and destination, or a URL)

© Copyright IBM Corporation 2015

Figure 2-8. Planning for event delivery

WB3942.0

Notes:

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

- Solution gateway API
 - The solution gateway API is the mechanism through which inbound events are submitted to the Decision Server Insights runtime environment and routed to the required agents. 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.
- Connectivity
 - Inbound connectivity acts as a bridge from external messaging endpoints to the solution gateway.
 - Outbound connectivity is the mechanism by which events can be delivered from a solution. There is no programmatic equivalent of the gateway for outbound events.

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, 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

© Copyright IBM Corporation 2015

Figure 2-9. Understanding “time” in Decision Server Insights

WB3942.0

Notes:

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.



Location-based reasoning

- If your solution needs to evaluate geographical locations or measure distances, make sure that you understand spatial geometries in Decision Server Insights
- Objects that define geographic locations are represented by these spatial geometries:
 - Geometry
 - Point
 - Line string
 - Linear ring
 - Vertex
 - Polygon

© Copyright IBM Corporation 2015

Figure 2-10. Location-based reasoning

WB3942.0

Notes:

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:

- The shape of the Earth
- The computation of floating points

2.2. Planning for development and test environments

Planning for development and test environments



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

Figure 2-11. Planning for development and test environments

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 2-12. Setting up the development and test environments

WB3942.0

Notes:

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 backend 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

© Copyright IBM Corporation 2015

Figure 2-13. Design concerns

WB3942.0

Notes:

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.

Planning development and production environments

- Development and test environments should emulate your production environment



© Copyright IBM Corporation 2015

Figure 2-14. Planning development and production environments

WB3942.0

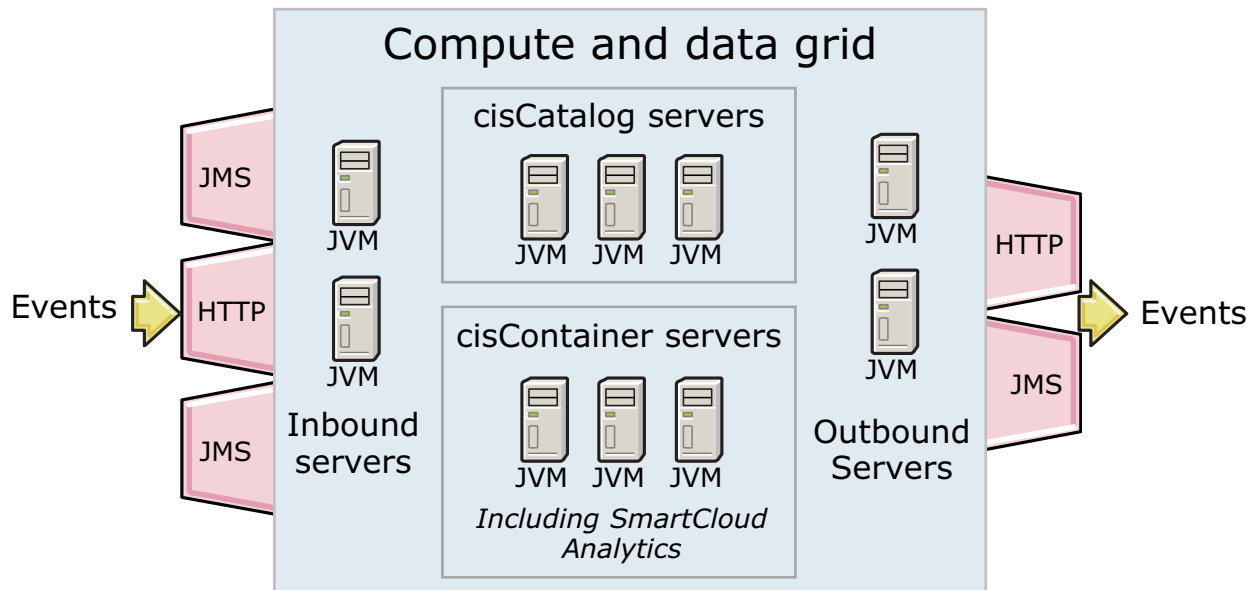
Notes:

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.

- Runtime environment and operations are centralized in a single data center.
- Inbound and outbound connectivity is hosted with the runtime container.
- HTTP features are enabled and JMS inbound and outbound features can be configured, if required. Persistence is turned off by default, and there is no disaster recovery facility.

Planning development and production environments



© Copyright IBM Corporation 2015

Figure 2-15. Planning development and production environments

WB3942.0

Notes:

Multiple server topology for production

Use this topology to meet your availability needs and to ensure minimal downtime in the event of an IT system disruption (disaster recovery). 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.

For more information about topologies, see the product documentation.

2.3. Creating solutions

Creating solutions



© Copyright IBM Corporation 2015

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

9.1

Figure 2-16. Creating solutions

WB3942.0

Notes:

Steps of building a Decision Server Insights solution

- To build your solution
 1. Create a solution project
 2. Import or create 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. Use the Java API to connect to a solution gateway, instantiate the necessary entities, and send an event payload to an Insight Server
 10. Run the application to submit events and see the agents execute

© Copyright IBM Corporation 2015

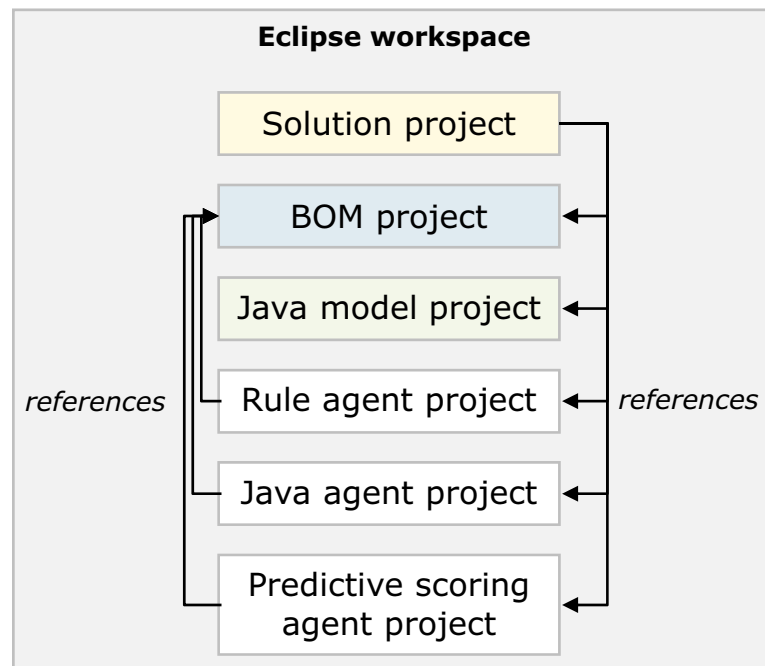
Figure 2-17. Steps of building a Decision Server Insights solution

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

Figure 2-18. Getting started on an Insights solution

WB3942.0

Notes:

The solution project is an Eclipse project that references the agent projects and the shared business object model. The solution project also contains the files to define the connectivity and the deployment configurations.

When you create a solution project, you also use a BOM project, which provides the vocabulary for your agents and rules.

After the business object model is complete, you can create the agent projects. Each agent is stored in a separate project. Rule agent projects can contain several rules, but only one agent per project. All agent projects reference the BOM project.

The Java Model project is a container for the executable object model that is generated from the business object model. The name of this project is based on the solution project name. This project is empty until the BOM is populated with entity and event definitions.

Rule agent project: The rule agent project is a container for rules that define the business logic. The rules are created in the rules folder of the agent project. The agent project also contains a descriptor that you must complete to define the subscription logic of the agent.

Java agent project: The Java agent project is a container for Java code to process events. In the Java agent, you import the package that contains the Java interfaces. The agent project also contains a descriptor that you must complete to define the subscription logic of the agent.

Predictive scoring agent project: The predictive scoring agent project agent calls an external scoring service that is available in SPSS and applies a scoring model.

The agent project also contains a descriptor that you must complete to define the subscription logic of the agent.

Decision Insight perspective

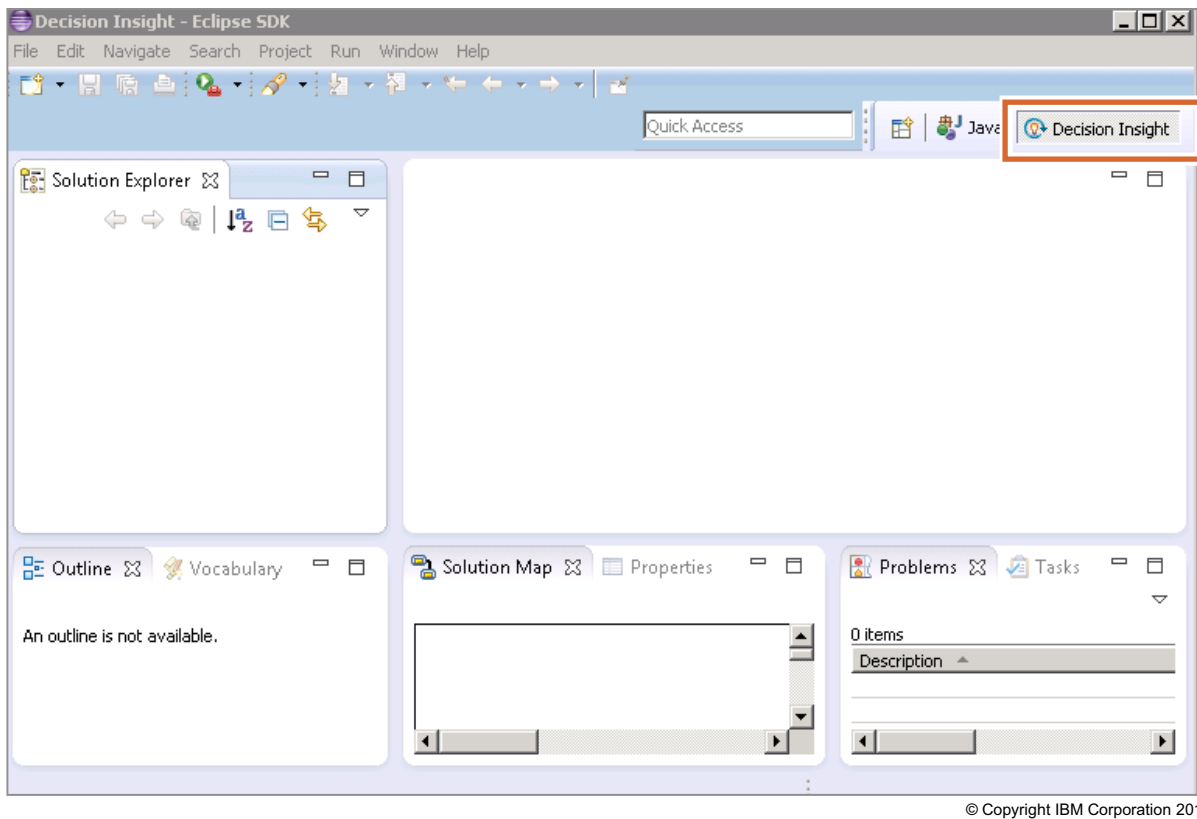


Figure 2-19. Decision Insight perspective

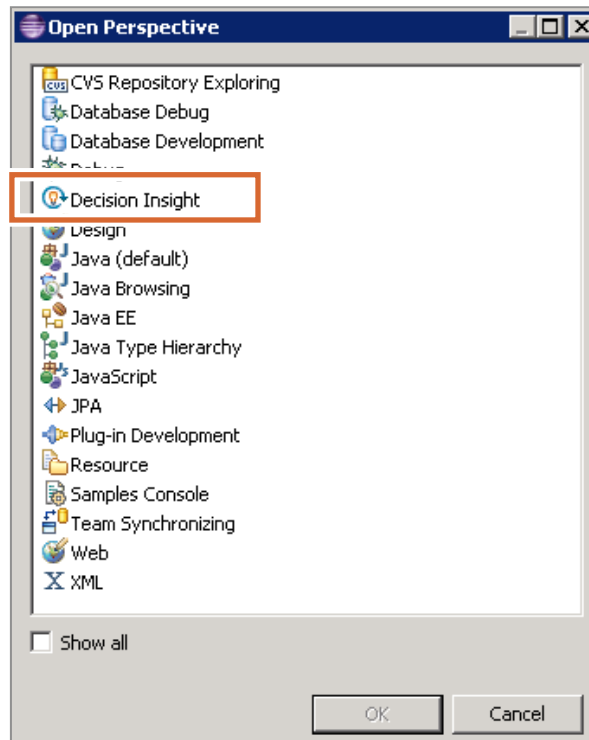
WB3942.0

Notes:

To create a Decision Server Insight solution, you use the Decision Insight perspective in Insight Designer.

Setting up the Decision Insight perspective

- Insight Designer uses the **Decision Insight** perspective



© Copyright IBM Corporation 2015

Figure 2-20. Setting up the Decision Insight perspective

WB3942.0

Notes:

By default, Insight Designer opens in the Java perspective, so you use the Open Perspective dialog box to switch.

Before you create a solution, you must set the target platform to use the Insight Server. Java agents are plug-in projects. To compile the Java code for these plug-in projects, you must set the Insight Server target platform. The target platform is a set of OSGi bundles.

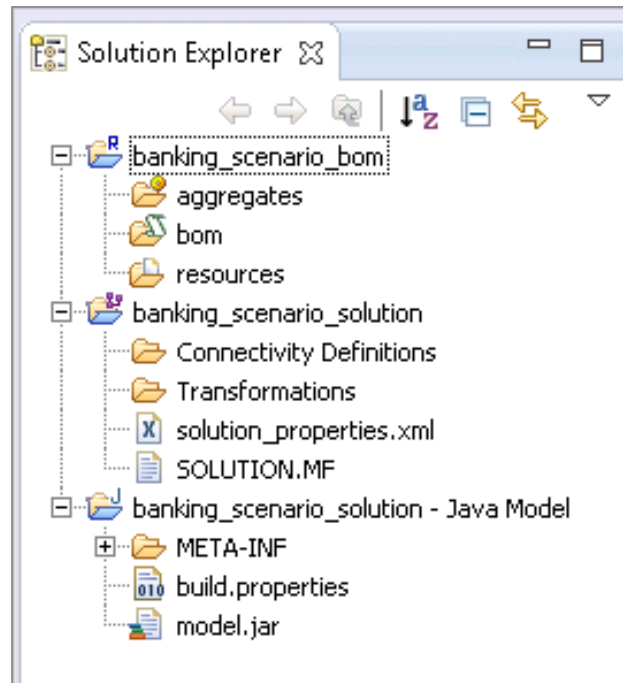


Note

You must set the target platform every time you create a workspace.

Solution project folders and files

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



© Copyright IBM Corporation 2015

Figure 2-21. Solution project folders and files

WB3942.0

Notes:

As you see in the example, when you create a solution, three projects are generated.

The BOM project 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)



Note

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

The solution project folder contains these folders and files:

- **Connectivity Definitions** folder: Stores the connectivity definition (.cdf) 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

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



© Copyright IBM Corporation 2015

Figure 2-22. Solution Map view in Decision Insight perspective

WB3942.0

Notes:

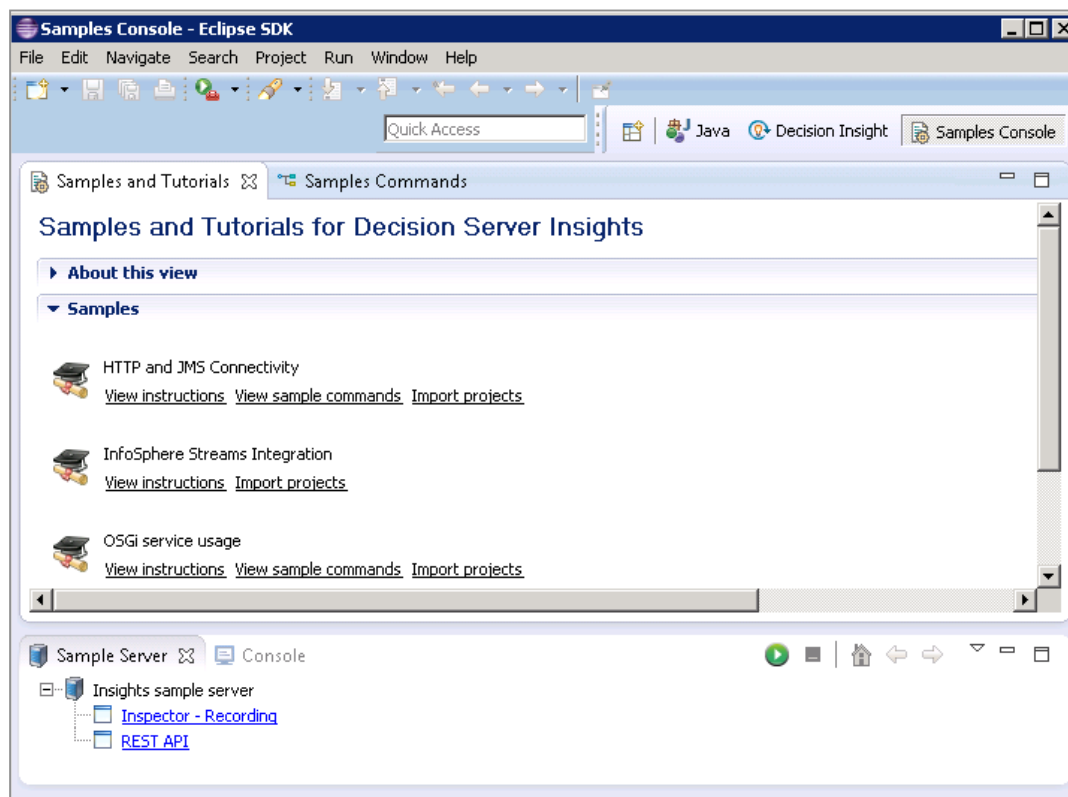
You start with modeling, which includes the business model, aggregates, and connectivity definitions. You learn more about the Model task later.

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 the solution references.

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.

Samples Console for Insights sample server



© Copyright IBM Corporation 2015

Figure 2-23. Samples Console for Insights sample server

WB3942.0

Notes:

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.

Unit summary

Having completed this unit, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 2-24. Unit summary

WB3942.0

Notes:

Checkpoint questions

1. True or false: An outbound event from one agent cannot be an inbound event for another agent.
2. 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

© Copyright IBM Corporation 2015

Figure 2-25. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

Checkpoint 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. 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).**

© Copyright IBM Corporation 2015

Figure 2-26. Checkpoint answers

WB3942.0

Notes:

Exercise 2

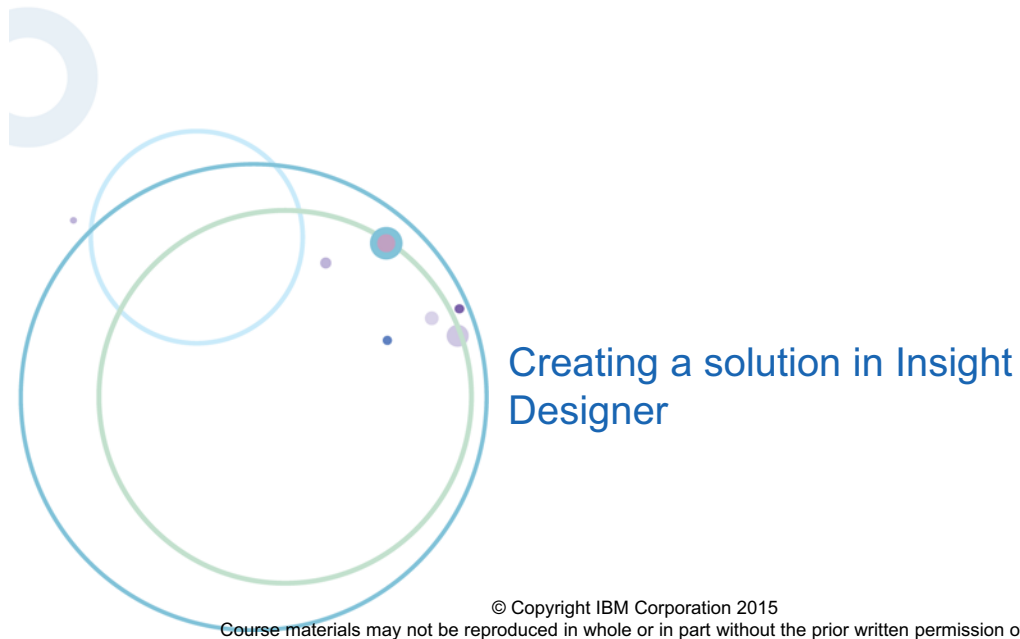


Figure 2-27. Exercise 2

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Create a solution project

© Copyright IBM Corporation 2015

Figure 2-28. Exercise objectives

WB3942.0

Notes:

Unit 3. Creating the business model

What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

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

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 3-1. Unit objectives

WB3942.0

Notes:

Topics

- Overview of the business model
- Writing the business model

© Copyright IBM Corporation 2015

Figure 3-2. Topics

WB3942.0

Notes:

3.1. Overview of the business model

Overview of the business model



© Copyright IBM Corporation 2015

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

9.1

Figure 3-3. Overview of the business model

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

BMD language is powerful, natural language, and it integrates well with source code control.

You model concepts (which aren't objects or events), such as an address.

Creating agents: the agent descriptor binds the agent to an entity so events and context are not mixed between entities.

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

© Copyright IBM Corporation 2015

Figure 3-5. Elements of a business model

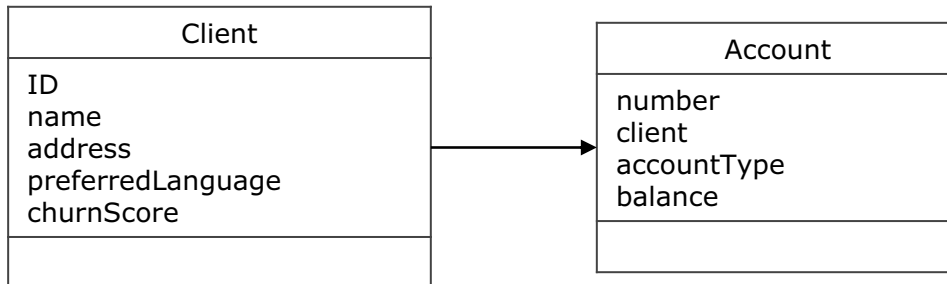
WB3942.0

Notes:

Entities have a lifecycle, which means they can live a long time in your domain. They have an identity (and an identifier). Events are related to them through the entity IDs that are carried by the events. Agents watch the entity and react when events or event patterns are detected for that entity.

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.

© Copyright IBM Corporation 2015

Figure 3-6. Example entity model

WB3942.0

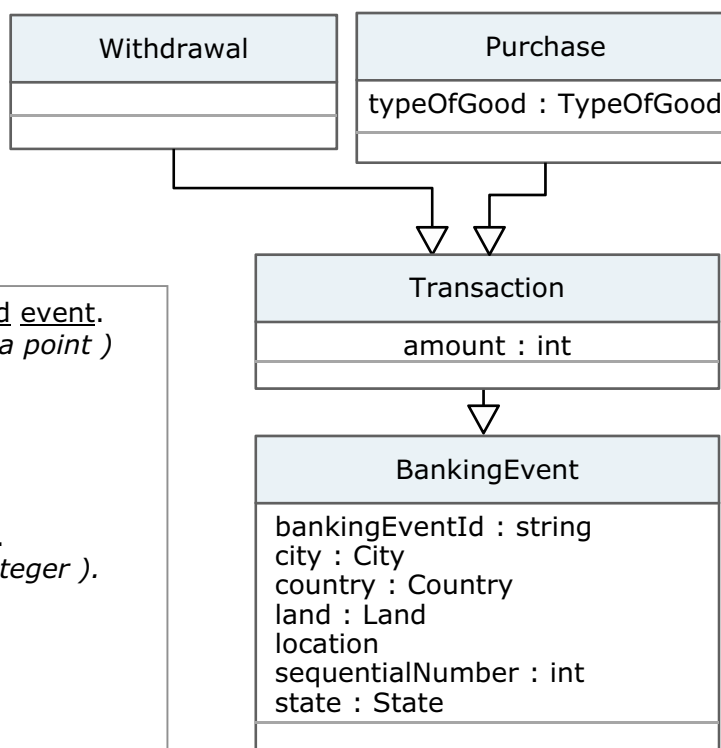
Notes:

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

In this example, the client ID attribute is the identifier. The account entity uses the number attribute as its identifier. You define the relationship between entities by using the “is related to” construct. In this example, an account is related to a client.

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 .

© Copyright IBM Corporation 2015

Figure 3-7. Example event model with relationships

WB3942.0

Notes:

An event type describes the shape of an event. An event type is 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 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 geolocations.

You can model **relationships** between object types. The type of construct that you use determines the type of relationship. You can also define simple and multiple relationships.

Example enumerations

- Enumerations define a list of possible values

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

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

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

© Copyright IBM Corporation 2015

Figure 3-8. Example enumerations

WB3942.0

Notes:

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

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.

© Copyright IBM Corporation 2015

Figure 3-9. Facets

WB3942.0

Notes:

3.2. Writing the business model

Writing the business model



© Copyright IBM Corporation 2015

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

9.1

Figure 3-10. Writing the business model

WB3942.0

Notes:

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.
```

© Copyright IBM Corporation 2015

Figure 3-11. Definition structure

WB3942.0

Notes:

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**

© Copyright IBM Corporation 2015

Figure 3-12. Defining attributes

WB3942.0

Notes:

Entities, concepts, and events can have attributes.

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.

© Copyright IBM Corporation 2015

Figure 3-13. Entity definition

WB3942.0

Notes:

Event definition

- 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.

© Copyright IBM Corporation 2015

Figure 3-14. Event definition

WB3942.0

Notes:

Business model editor (1 of 3)

- **Definitions**
tab

- Model is written with natural language in the BMD



```

a client is a business entity identified by a name.
a client has a segment.
a client has a churn score (numeric).
a client has a monthly profitability (numeric).
a client has a propensity to buy HIGH END WINE (numeric).
a client has a propensity to buy BROADWAY SHOW TICKETS (numeric).
a client has a preferred language (a language ).
a client has a locations (a multi point ) .

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

a banking event is a client related event.
a banking event has a sequential number (integer ).
a banking event has a location (a point ) used as the default geometry.
a banking event has a country.
a banking event has a Land.
a banking event has a state.
a banking event has a county.
a banking event has a city.
  
```

© Copyright IBM Corporation 2015

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

WB3942.0

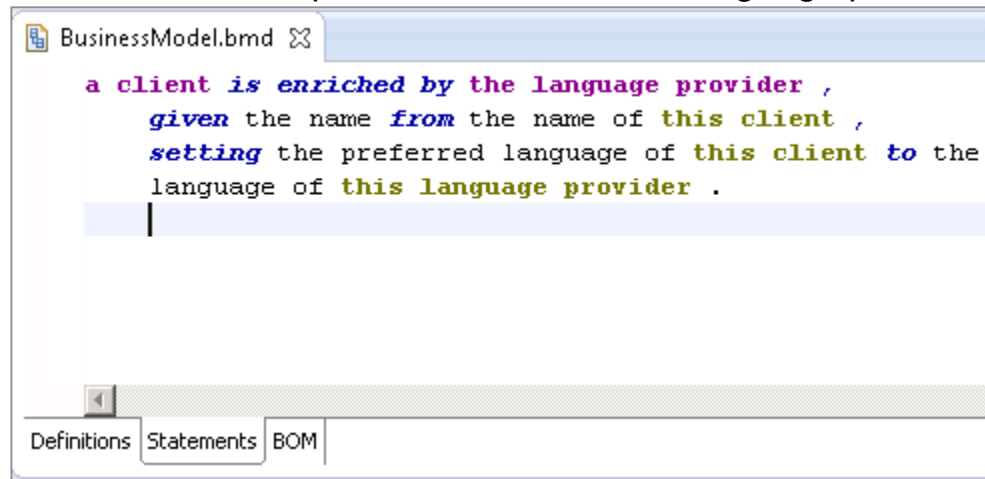
Notes:

The model is written in BMD, which is Business Model Definition language.

- Natural language
- Integrates well with source code control

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 a data provider that is called “language provider”



© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

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

Business model editor (3 of 3)

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



© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 3-18. Unit summary

WB3942.0

Notes:

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 3-19. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 3-20. Checkpoint answers

WB3942.0

Notes:

Exercise 3

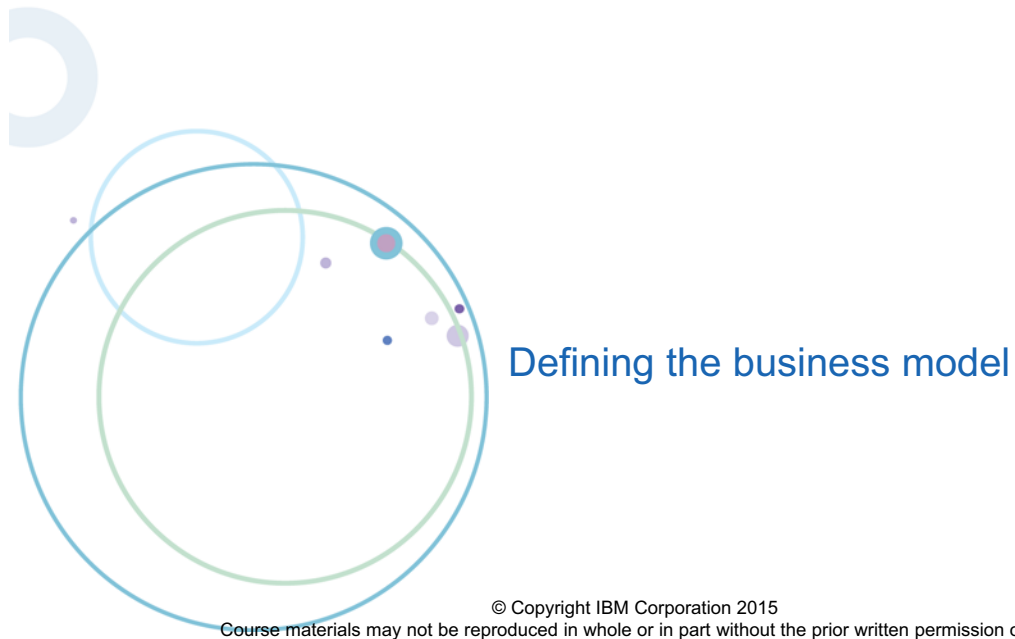


Figure 3-21. Exercise 3

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Create a business model definition file

© Copyright IBM Corporation 2015

Figure 3-22. Exercise objectives

WB3942.0

Notes:

Unit 4. Authoring the business logic

What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

- 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
- Define global aggregates

How you will check your progress

- Checkpoint
- Exercises

Unit objectives

After completing this unit, you should be able to:

- 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
- Define global aggregates

© Copyright IBM Corporation 2015

Figure 4-1. Unit objectives

WB3942.0

Notes:

Topics

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

© Copyright IBM Corporation 2015

Figure 4-2. Topics

WB3942.0

Notes:

4.1. Implementing business logic with agents

Implementing business logic with agents



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

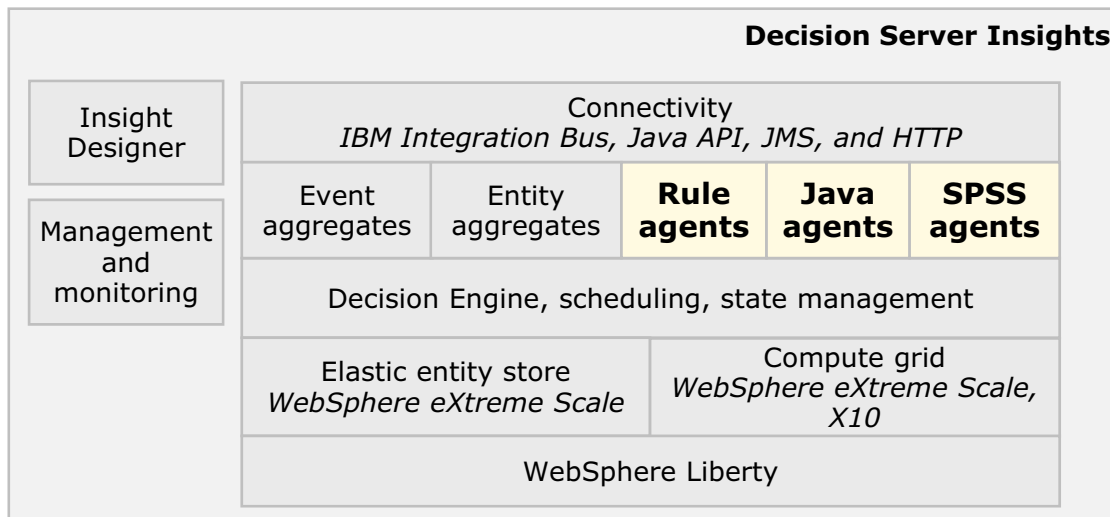
Figure 4-3. Implementing business logic with agents

WB3942.0

Notes:

Runtime architecture: Agents

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



© Copyright IBM Corporation 2015

Figure 4-4. Runtime architecture: Agents

WB3942.0

Notes:

Agents are managed in the runtime tier.

The Decision Server Insights solution routes events to agents based on the data in the incoming events (payload), binds an entity to the agent, and runs the business logic that is defined in the agent.

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

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 send new events
- Agents can either be triggered by the arrival of events or scheduled for execution

© Copyright IBM Corporation 2015

Figure 4-5. Agents

WB3942.0

Notes:

Agents relate events to entities, and detect situations such as risks or opportunities by using correlation, aggregation, rules, predictive models, and time logic.

Java agents can also operate over single events or a sequence of events independent of any entity.

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
- You can define priority for the agent to determine the runtime invocation order if an event is dispatched to several agents that target the same entity

© Copyright IBM Corporation 2015

Figure 4-6. Agents and entities

WB3942.0

Notes:

Agents can update their internal state, update the state of their bound entity, or emit new events. An agent can create, update, or remove its bound entity. An agent has read-only access to other non-bound entities.

The business logic for an agent can include the following operations, which can be combined:

- Emit a derived event that is based on the data in the incoming event.
- Emit a derived event that is based on events that are already received.
- Emit a derived event that is based on the state of the entity, or state accessible from the entity, which can include some temporal filtering of the attributes of the entity.
- Update the attributes of the bound entity.
- Read attributes of an entity that is resolved through a relationship from the bound entity or event.
- Create the bound entity of the agent from data in the incoming event.
- Remove the entity that is bound to the agent.

Agent types

- 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
- 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

© Copyright IBM Corporation 2015

Figure 4-7. Agent types

WB3942.0

Notes:

Creating rule agents

Create a rule agent to define the business logic that binds incoming events to entities.

Creating Java agents

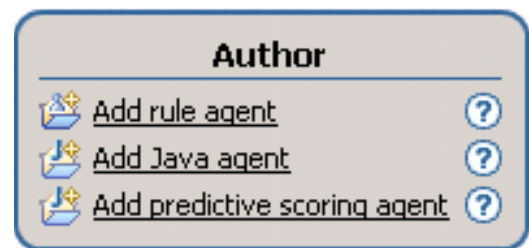
Create a Java agent to process events by using Java code. For example, you create a Java agent to calculate complex computations, to call a web service, or to write some logic that is not likely to change.

Creating predictive scoring agents

Use the Scoring Agent wizard to create a predictive scoring agent that calls a scoring configuration on an external scoring server. The agent invokes a Java class that supplies input and output values.

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



© Copyright IBM Corporation 2015

Figure 4-8. Creating rule agents

WB3942.0

Notes:

Before you create a rule agent, you must create a solution project and model the entities and events.

In the rule agent, you complete the description of the agent to specify the bound entity and the events to process.

Then, you write the rules that define the business logic of the agent.

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

© Copyright IBM Corporation 2015

Figure 4-9. Agent projects

WB3942.0

Notes:

The agent signature also defines the applicability condition. Decision Server Insights routes events to specific agents, and loads the data to execute the business logic.

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

© Copyright IBM Corporation 2015

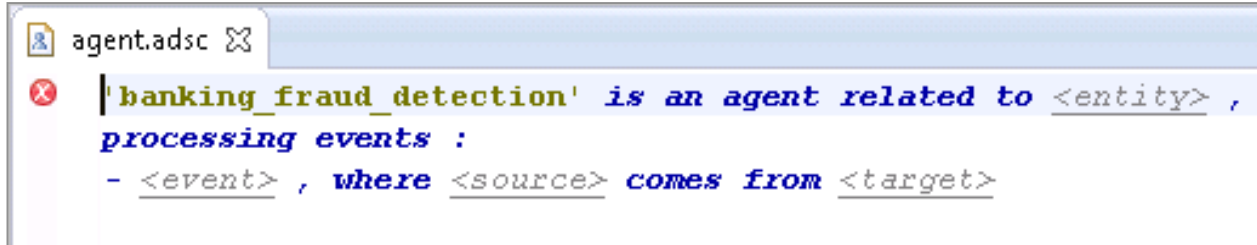
Figure 4-10. Agent descriptors

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 4-11. Writing an agent descriptor

WB3942.0

Notes:

The agent descriptor routes appropriate events to the entity. In this example, all banking events for this client entity are processed by the agent for detecting fraud patterns. The rules that are attached to the agent implement the business logic for fraud detection.

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*

© Copyright IBM Corporation 2015

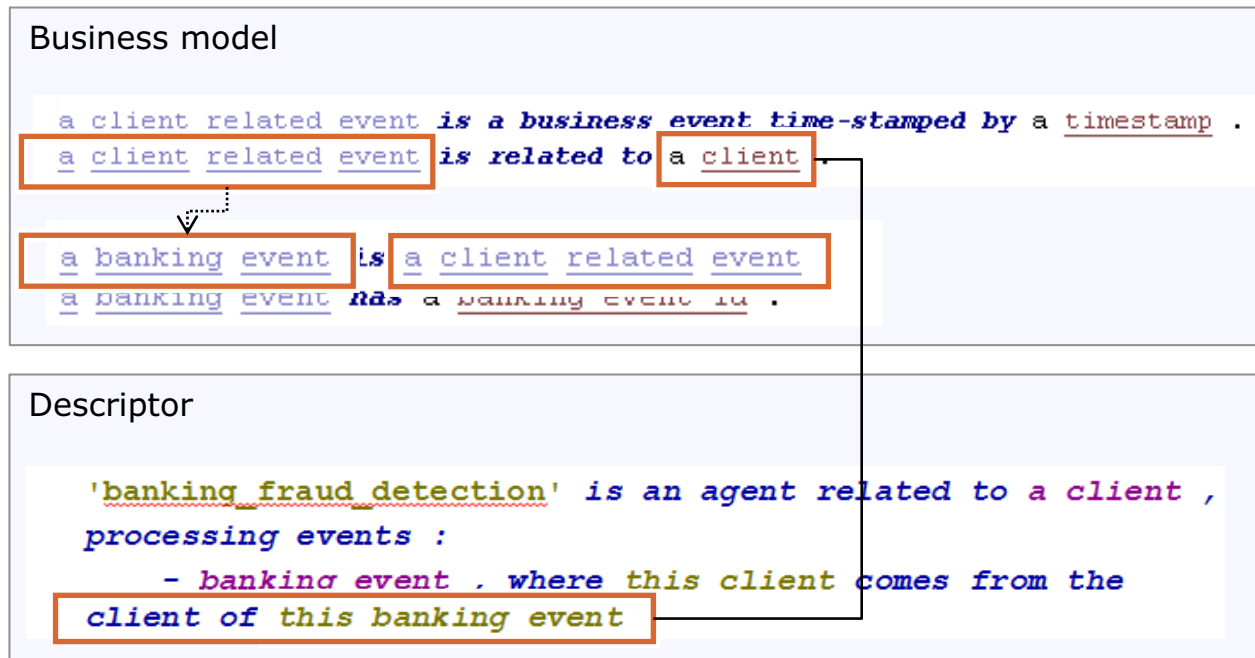
Figure 4-12. Example descriptor

WB3942.0

Notes:

Relationship between events and entities

- Relationships that are defined in the business model are used in the descriptor to determine how the entity is retrieved from the event



© Copyright IBM Corporation 2015

Figure 4-13. Relationship between events and entities

WB3942.0

Notes:

The vocabulary available in the rules comes from the elements that are defined in the business model and the agent descriptor.

4.2. Rule agents

Rule agents



9.1

Figure 4-14. Rule agents

WB3942.0

Notes:

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
- By using the rules, rule agents can emit new events

© Copyright IBM Corporation 2015

Figure 4-15. Creating rule agents

WB3942.0

Notes:

Structure of an action rule

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

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

© Copyright IBM Corporation 2015

Figure 4-16. Structure of an action rule

WB3942.0

Notes:

A rule defines the specific actions to take when certain conditions are met. A rule is composed of several parts that define the conditions and actions. The structure of the rule depends on the type of rule that you want to write.

A basic rule associates a condition with an action. The rule states what action to perform when a condition is true.

The vocabulary available in the rules comes from the elements that are defined in the business model and the agent descriptor. You write a rule in a formal rule language. A rule can be expressed by using business terms, variables, operators, and values.

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 only applicable 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`

© Copyright IBM Corporation 2015

Figure 4-17. Defining the “when” part of a rule

WB3942.0

Notes:

The *when* part is used for rules that process events.

Rules with a *when* part are executed upon the reception of the event that is specified in the *when* part, or at a specified time after the reception of this event.

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

© Copyright IBM Corporation 2015

Figure 4-18. Using “when” to postpone event processing

WB3942.0

Notes:

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 ;
```

© Copyright IBM Corporation 2015

Figure 4-19. Example: Testing for the absence of an event

WB3942.0

Notes:

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 ;

© Copyright IBM Corporation 2015

Figure 4-20. Defining rule variables

WB3942.0

Notes:

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 ;
```

© Copyright IBM Corporation 2015

Figure 4-21. Calling global aggregate variables in rules

WB3942.0

Notes:

You can use your global event aggregates and global entity aggregates similar to the way you use other rule variables. For example, you can use global aggregates in condition statements and calculations to determine a course of action or whether an event should be emitted.

However, because global aggregates are defined and calculated by a different process than rule variables, you cannot create them in a rule.

For event aggregates, the value of the variable is calculated and updated locally after an event arrives. Because updates occur on all nodes in the system, the value becomes consistent only after the aggregated values are received at every node. At one time, agents on different servers might have different aggregate values.

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:`

© Copyright IBM Corporation 2015

Figure 4-22. Defining conditions

WB3942.0

Notes:

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 and 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;
```

© Copyright IBM Corporation 2015

Figure 4-23. Defining actions

WB3942.0

Notes:

Time-related tests

- 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 12/1/2014
then
    print "A banking event occurred after 12/1/2014"
```

© Copyright IBM Corporation 2015

Figure 4-24. Time-related tests

WB3942.0

Notes:

Now is set from the time stamp of the event, which is truncated to the second. If events arrive out of order, **now** remains set at the most recent time. For example, if an event arrives with a time stamp of today at 11:00:00 AM, followed by another event with a time stamp of today at 10:55:00 AM, **now** is set to 11:00:00 AM when both events are processed.

- Events might arrive with a delay in the agent, and in an unexpected order.
- Events have a time stamp that is defined by the event source.
- In most cases, there is 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 before `now`.

Location-related tests

- You can write rules to reason over location-aware entities or events and perform the following operations:
 - Detect if 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

© Copyright IBM Corporation 2015

Figure 4-25. Location-related tests

WB3942.0

Notes:

4.3. Java agents and predictive scoring agents

Java agents and predictive scoring agents



© Copyright IBM Corporation 2015

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

9.1

Figure 4-26. Java agents and predictive scoring agents

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 4-27. Setting the target platform in Insight Designer

WB3942.0

Notes:

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.

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 )
```

© Copyright IBM Corporation 2015

Figure 4-28. Java agents

WB3942.0

Notes:

Use Java agents to calculate complex computations to call a web service or to write some logic that is not likely to change.

For details on developing Java agents, see:

https://www.ibm.com/support/knowledgecenter/SSQP76_8.7.0/com.ibm.odm.itoa.develop/topics/tsk_create_java_agent_intro.html?lang=en

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

© Copyright IBM Corporation 2015

Figure 4-29. Predictive agents

WB3942.0

Notes:

Decision Server Insights provides a dedicated wizard to create a predictive scoring agent that links to the scoring server.

- An SPSS model is created in SPSS Modeler.
- A scoring server is exposed in SPSS Runtime Infrastructure.

For details on developing predictive scoring agents, see:

https://www.ibm.com/support/knowledgecenter/SSQP76_8.7.0/com.ibm.odm.itoa.develop/topics/tsk_dev_predictive_agent.html?lang=en

4.4. Global aggregates

Global aggregates



9.1

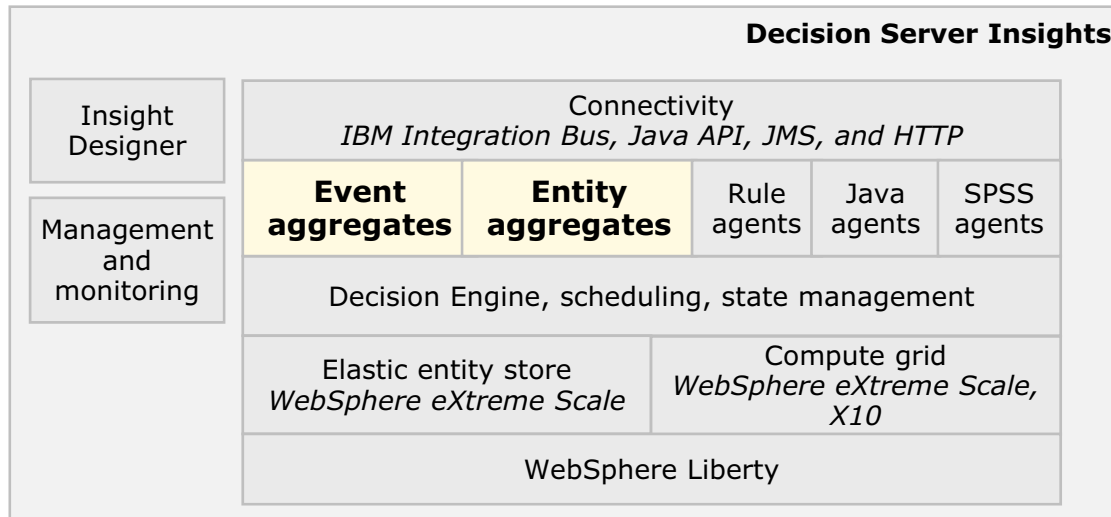
Figure 4-30. Global aggregates

WB3942.0

Notes:

Runtime architecture: Global aggregates

- Two types:
 - Event aggregates
 - Entity aggregates



© Copyright IBM Corporation 2015

Figure 4-31. Runtime architecture: Global aggregates

WB3942.0

Notes:

Decision Server Insights includes both a **local** and **global** programming model for entities and events. The global programming model is supported by X10, which provides a globally distributed, asynchronous programming model.

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

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
- Scalar values
 - What is the average age of all customers?
 - How many accounts were opened last year?
- Local event aggregates accumulate information over all events that are delivered to a particular entity (as opposed to all entities)
 - Local aggregate: 30-day average transactions of the account
 - Global aggregate: 30-day average transactions of all accounts

© Copyright IBM Corporation 2015

Figure 4-32. What is a global aggregate

WB3942.0

Notes:

A global event aggregate can find the total, minimum, maximum, or average value of a collection of events, or the number of events. For example, if your events are flight departures, you can determine the average number of delayed flight departures over the past month or another time period.

Types of global aggregate

- Event aggregates
- Entity aggregates
- Local event aggregates accumulate information over all events that are delivered to a particular entity (as opposed to all entities)
 - Local aggregate example: 30-day average transactions of **the account**
 - Global aggregate example: 30-day average transactions of **all accounts**

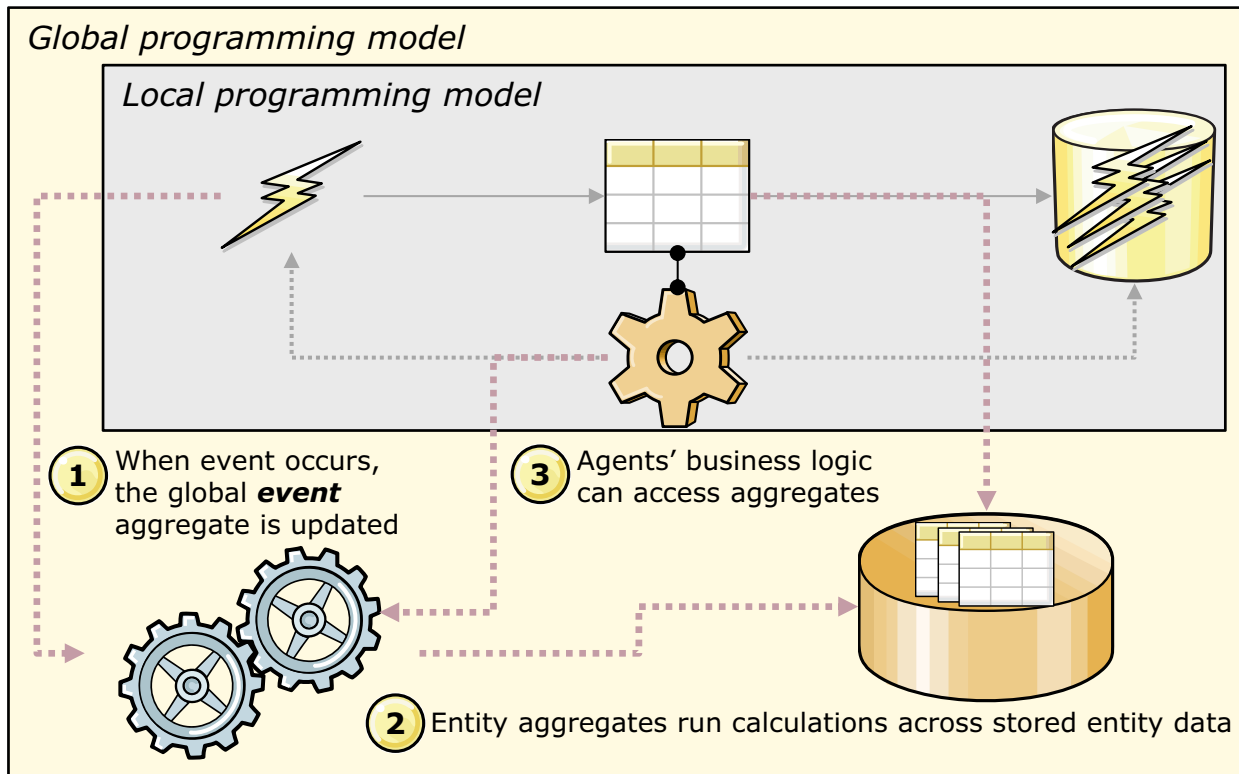
© Copyright IBM Corporation 2015

Figure 4-33. Types of global aggregate

WB3942.0

Notes:

Aggregates in the programming model



© Copyright IBM Corporation 2015

Figure 4-34. Aggregates in the programming model

WB3942.0

Notes:

Global event aggregates run continuously as part of a deployed solution. A global event aggregate is calculated every time one of the events occurs. You can use event aggregates to calculate values from specific types of events. Every time an event occurs, the aggregate updates its stored value.

Global entity aggregates calculate values across all the entities. The global aggregate is one of the sources for the decision context within a local rule.

Aggregate values are accessible to rule agents. Predictive agents can also access global aggregates.

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
```

© Copyright IBM Corporation 2015

Figure 4-35. Global entity aggregate

WB3942.0

Notes:

A global entity aggregate can find the total, minimum, maximum, or average value of a collection of entities and their attributes. For example, if your model has aircraft entities, you can determine the average load of all the aircraft and evaluate that average daily.

The calculations are launched according to the schedule and are run as batch jobs that occur in each of the nodes of your solution. Therefore, the results from the individual partitions are not initially consistent. However, over time a feedback mechanism ensures that the results become consistent.

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`

© Copyright IBM Corporation 2015

Figure 4-36. Global event aggregates

WB3942.0

Notes:

A global event aggregate can find the total, minimum, maximum, or average value of a collection of events, or the number of events. For example, if your events are flight departures, you can determine the average number of delayed flight departures over the past month or another time period.

Global event aggregates run continuously as part of a deployed solution. A global event aggregate is calculated every time one of the events occurs.

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

© Copyright IBM Corporation 2015

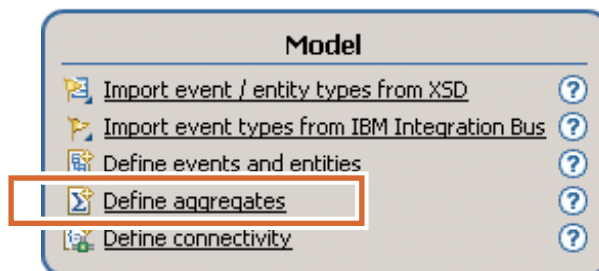
Figure 4-37. Global aggregate functions

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

Figure 4-38. Defining aggregates

WB3942.0

Notes:

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.

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>
```

© Copyright IBM Corporation 2015

Figure 4-39. Retrieving global aggregate value

WB3942.0

Notes:

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

Aggregates can be called from rules, Java code, or REST.

Unit summary

Having completed this unit, you should be able to:

- 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
- Define global aggregates

© Copyright IBM Corporation 2015

Figure 4-40. Unit summary

WB3942.0

Notes:

Checkpoint 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
3. True or false: In Java agents, there is a separate API to access event and entity aggregates.
4. True or false: A global entity aggregate can perform calculations across entire populations of entities.

© Copyright IBM Corporation 2015

Figure 4-41. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

4.

Checkpoint 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.

3. In Java agents, there is a separate API to access event and entity aggregates

Answer: False. You use the same API to access both event and entity aggregates.

4. A global entity aggregate can perform calculations across entire populations of entities

Answer: True.

© Copyright IBM Corporation 2015

Figure 4-42. Checkpoint answers

WB3942.0

Notes:

Exercise 4

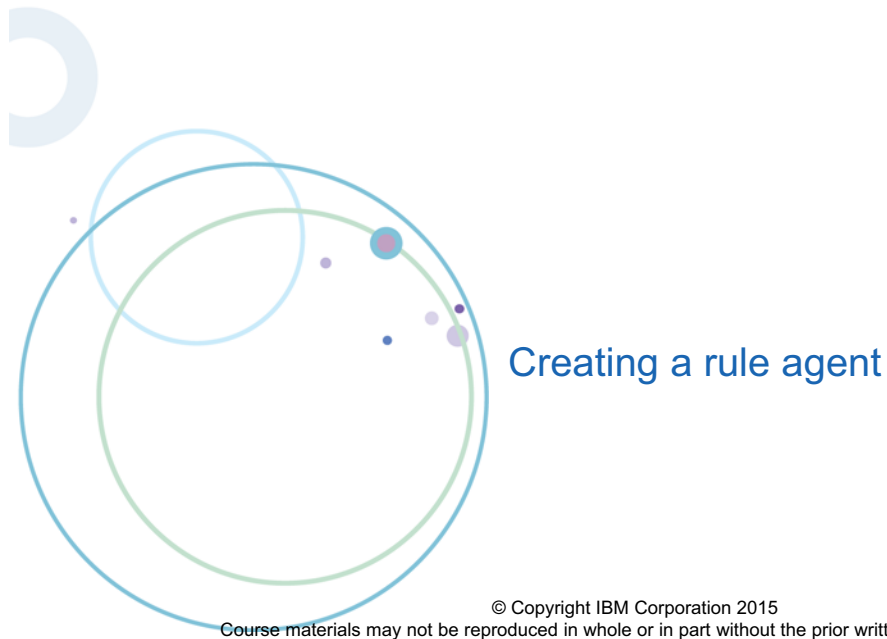


Figure 4-43. Exercise 4

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

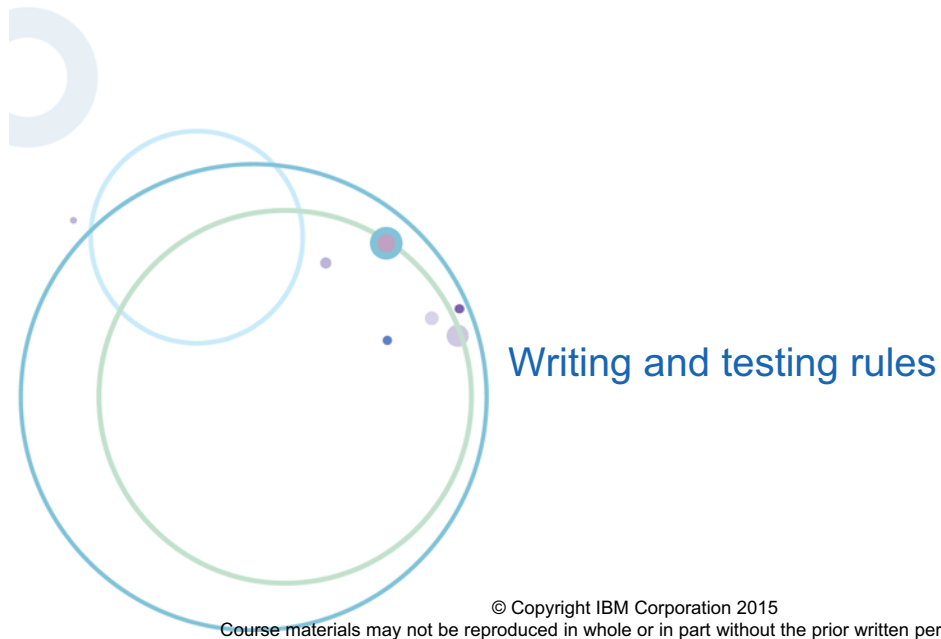
© Copyright IBM Corporation 2015

Figure 4-44. Exercise objectives

WB3942.0

Notes:

Exercise 5



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

Figure 4-45. Exercise 5

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 4-46. Exercise objectives

WB3942.0

Notes:

Exercise 6

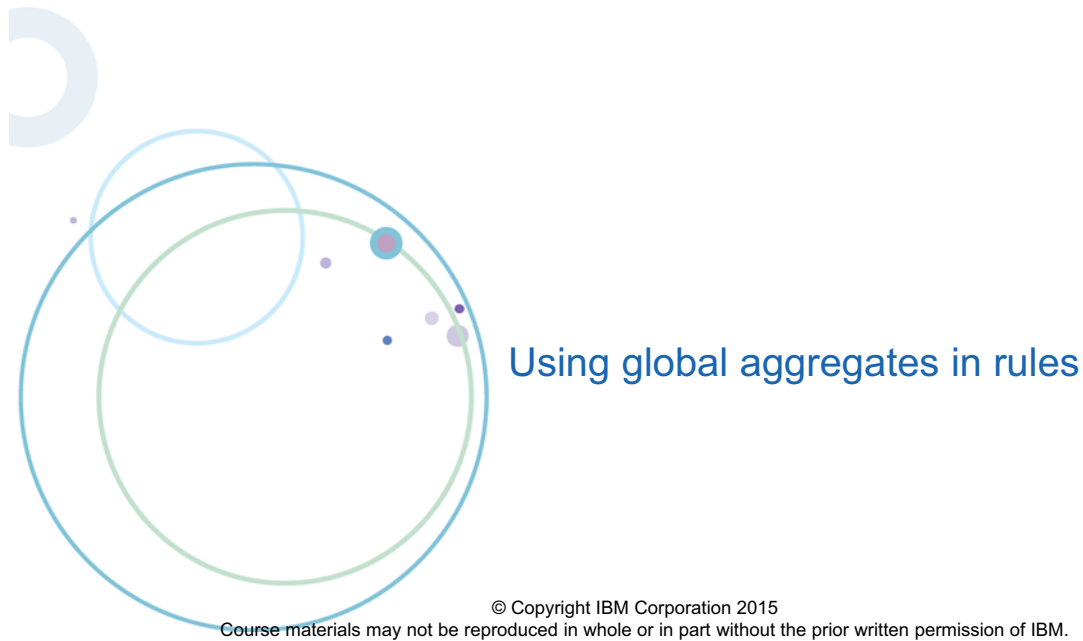


Figure 4-47. Exercise 6

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 4-48. Exercise objectives

WB3942.0

Notes:

Exercise 7

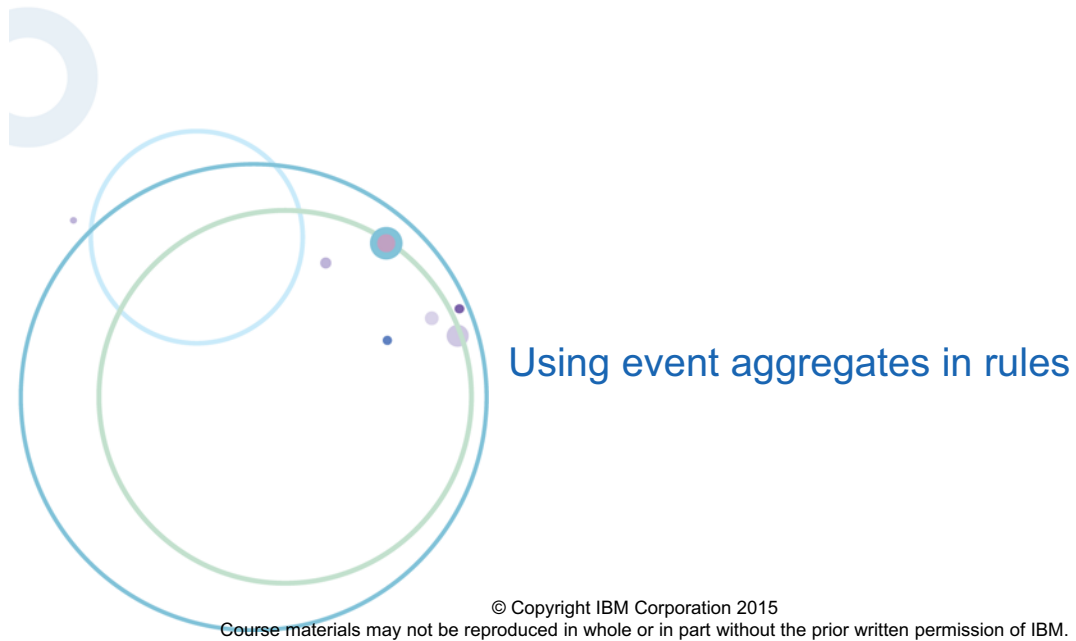


Figure 4-49. Exercise 7

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Use event aggregates and aggregate functions in rules

© Copyright IBM Corporation 2015

Figure 4-50. Exercise objectives

WB3942.0

Notes:

Exercise 8

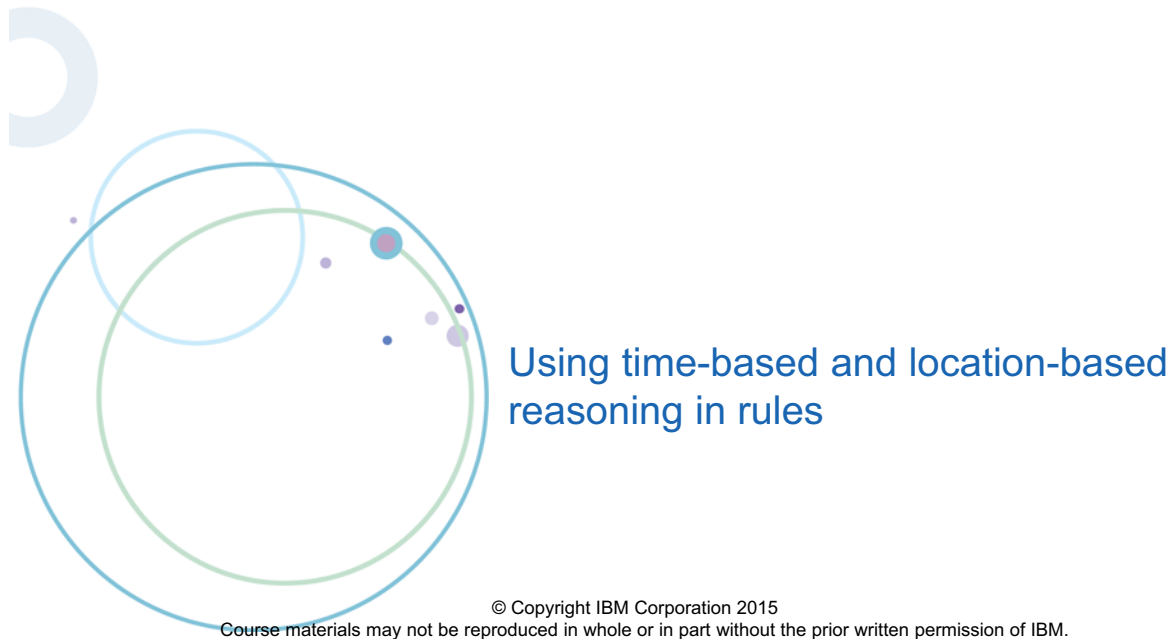


Figure 4-51. Exercise 8

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 4-52. Exercise objectives

WB3942.0

Notes:

Exercise 9

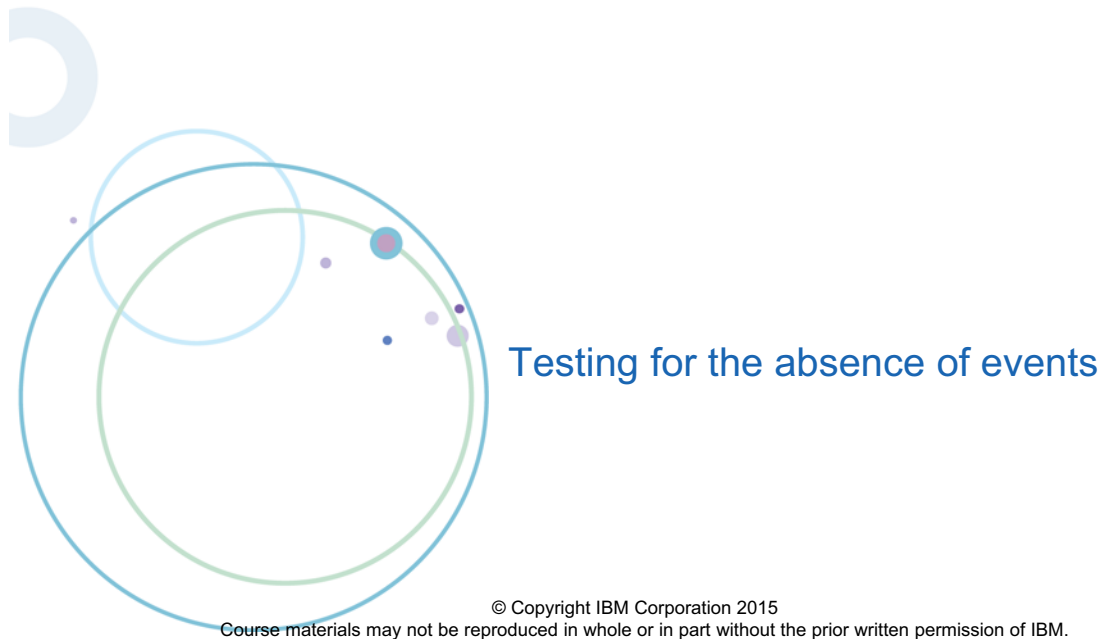


Figure 4-53. Exercise 9

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Test for the absence of events

© Copyright IBM Corporation 2015

Figure 4-54. Exercise objectives

WB3942.0

Notes:

Unit 5. Deploying solutions

What this unit is about

This unit teaches you how to manage solution deployment.

What you should be able to do

After completing this unit, you should be able to:

- Explain how to export and deploy solutions
- Describe how to manage solutions through the solutionManager script

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Explain how to export and deploy solutions
- Describe how to manage solutions through the solutionManager script

© Copyright IBM Corporation 2015

Figure 5-1. Unit objectives

WB3942.0

Notes:



Topics

- Exporting and deploying

© Copyright IBM Corporation 2015

Figure 5-2. Topics

WB3942.0

Notes:

5.1. Exporting and deploying

Exporting and deploying



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

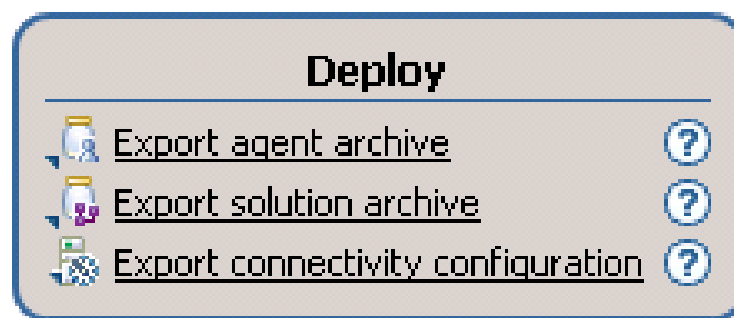
Figure 5-3. Exporting and deploying

WB3942.0

Notes:

Deploying from Insight Designer

- Export solution archive
- Deploy to Insight Server
- Use links in the Deploy goal of the Solution Map to guide you through exporting and deploying



© Copyright IBM Corporation 2015

Figure 5-4. Deploying from Insight Designer

WB3942.0

Notes:

After you develop a solution and want to test it, you must first export the solution as an archive, and then deploy the archive file to an Insight Server.

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`

© Copyright IBM Corporation 2015

Figure 5-5. Exporting solutions and agents

WB3942.0

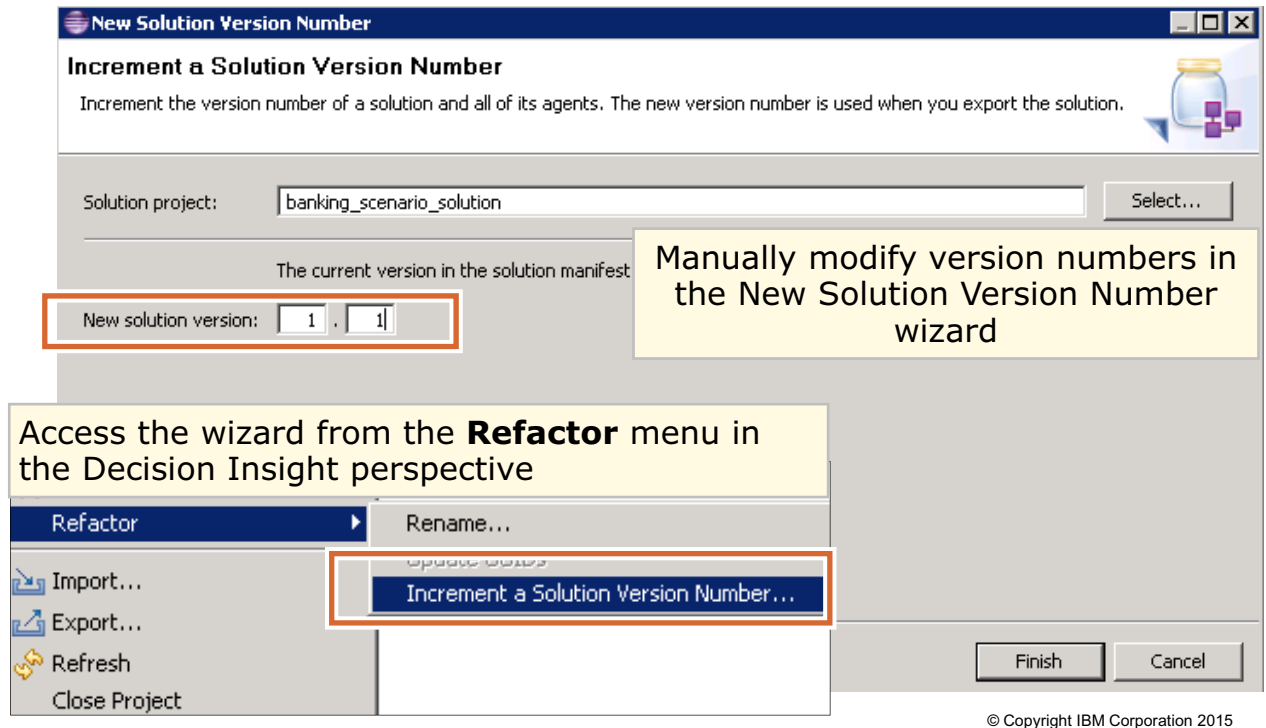
Notes:

You can export a solution and its agents to an archive that you then deploy 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.

Version policy

- By default, solutions are deployed with the version 1.0



© Copyright IBM Corporation 2015

Figure 5-6. Version policy

WB3942.0

Notes:

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 is 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.

Deploying solution archives

- To deploy a solution from Insight Designer, you can either:
 - Run the solution manager script
 - Use JMX MBeans
- The deployment script installs the solution files on the server and updates the `server.xml` file to activate the solution on the Liberty server
- Run the script on Windows from the `<InstallDir>/runtime/ia/bin` directory
- You can run the deployment script locally or remotely
 - Example: Local deployment

```
solutionManager deploy local C:\solution.esa --server=cisDev
```

© Copyright IBM Corporation 2015

Figure 5-7. Deploying solution archives

WB3942.0

Notes:

You can deploy a solution across a cluster. All servers in the cluster must have the same solutions installed.

The remote deployment script includes parameters for the remote host. Example:

- Remote:

```
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 solution manager scripts are run from the `InstallDir/runtime/ia/bin` directory.

Deployed solutions files

- After you deploy, the solution is deployed to the **<InstallDir>/runtime/solutions/lib** directory
- The solution manifest file is copied to a product extension directory named **<InstallDir>/runtime/solutions/lib/features**
- Use the REST API to verify deployment by typing this URL in a browser:
`http://localhost:9080/ibm/ia/rest/solutions`

© Copyright IBM Corporation 2015

Figure 5-8. Deployed solutions files

WB3942.0

Notes:

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.

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 without stopping it, you get errors

© Copyright IBM Corporation 2015

Figure 5-9. Undeploying (1 of 2)

WB3942.0

Notes:

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.

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>\solutions\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

© Copyright IBM Corporation 2015

Figure 5-10. Undeploying (2 of 2)

WB3942.0

Notes:

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 solution directories `InstallDir/ia/solutions/lib` and `InstallDir/ia/solutions/lib/features`.

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
```

© Copyright IBM Corporation 2015

Figure 5-11. Deleting solution files

WB3942.0

Notes:

To delete these files, open the `InstallDir\solutions\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 the server as a background process
- `stop`: Stops a catalog or connectivity server

Unit summary

Having completed this unit, you should be able to:

- Explain how to export and deploy solutions
- Describe how to manage solutions through the solutionManager script

© Copyright IBM Corporation 2015

Figure 5-12. Unit summary

WB3942.0

Notes:

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 5-13. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 5-14. Checkpoint answers

WB3942.0

Notes:

Exercise 10

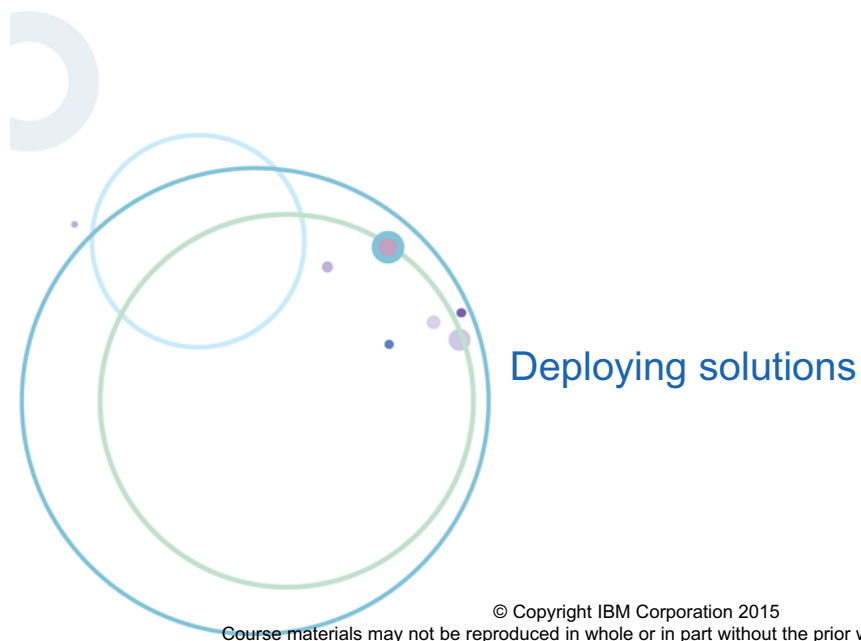


Figure 5-15. Exercise 10

WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 5-16. Exercise objectives

WB3942.0

Notes:

Unit 6. Testing solutions

What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

- Test solutions with the TestDriver API
- Troubleshoot with the REST API
- Troubleshoot with log analysis
- Use Insight Inspector
- Create testing exercises with Generic Client

How you will check your progress

- Checkpoint

Unit objectives

After completing this unit, you should be able to:

- Test solutions with the TestDriver API
- Troubleshoot with the REST API
- Troubleshoot with log analysis
- Use Insight Inspector
- Create testing exercises with Generic Client

© Copyright IBM Corporation 2015

Figure 6-1. Unit objectives

WB3942.0

Notes:



Topics

- TestDriver API
- REST API
- Log analysis
- Insight Inspector
- Generic Client

© Copyright IBM Corporation 2015

Figure 6-2. Topics

WB3942.0

Notes:

6.1. TestDriver API

TestDriver API



© Copyright IBM Corporation 2015

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

9.1

Figure 6-3. TestDriver API

WB3942.0

Notes:

TestDriver API

- A TestDriver object provides all methods that are required to test a solution
- TestDriver helps you test and debug solutions, manage entities, determine the system and solution status, and create and submit events to Insight Server
- The TestDriver class is the main entry point for testing a solution
- TestDriver provides model factory methods, such as getConceptFactory and getEventFactory to create events, entities, and concepts
- The TestDriver API captures events for debugging and problem determination

© Copyright IBM Corporation 2015

Figure 6-4. TestDriver API

WB3942.0

Notes:

TestDriver properties

- TestDriver is configured through several properties that are contained within a Java Properties object that contains property/value pairs for the various properties
- To ensure that a test driver instance can connect to a server, create a `testdriver.properties` file and add properties and to modify the behavior of the test driver

© Copyright IBM Corporation 2015

Figure 6-5. TestDriver properties

WB3942.0

Notes:

Example test driver 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
```

© Copyright IBM Corporation 2015

Figure 6-6. Example test driver properties

WB3942.0

Notes:

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);
}
```

© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

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);
```

© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

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();
```

© Copyright IBM Corporation 2015

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

WB3942.0

Notes:

6.2. REST API

REST API



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

Figure 6-11. REST API

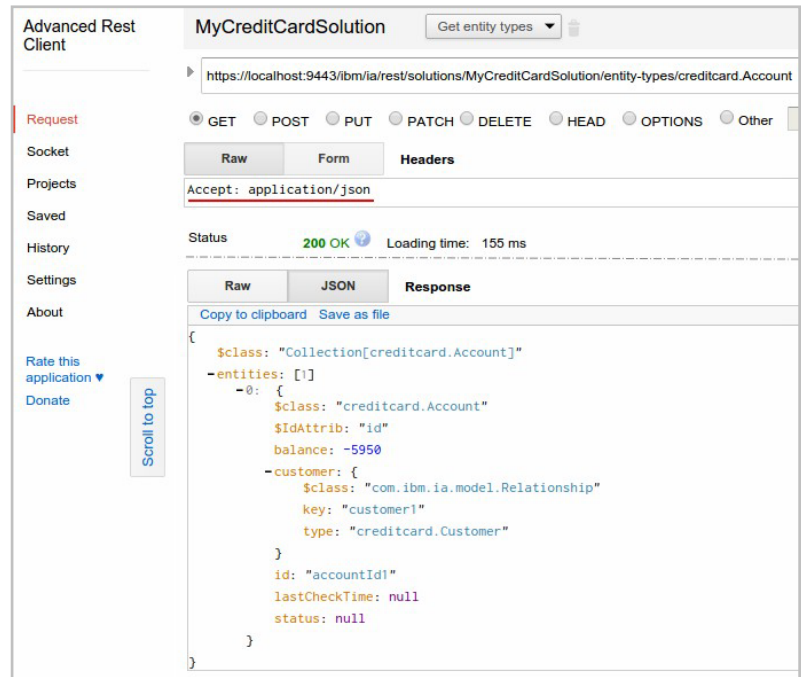
WB3942.0

Notes:

REST methods

- Use a REST client for introspection of the deployed solutions, with types of entities, entity instances, and global aggregate values

- To get **JSON responses**
 - Set the "Accept:" header in the request to "get"
 - More readable than XML for debugging



© Copyright IBM Corporation 2015

Figure 6-12. REST methods

WB3942.0

Notes:

You can use REST to view deployed solutions, types of entities, entity instances, and global aggregate values. However, REST does not provide visibility on the events that are stored in the working memory of agents.

REST resources

- To list all deployed solutions:
`http://localhost:9080/ibm/ia/rest/solutions`
- To list all entity types that are managed by your solution:
`http://localhost:9080/ibm/ia/rest/solutions/
MySolution/entity-types`
- To list all aggregates in your solution:
`http://localhost:9080/ibm/ia/rest/solutions/
MySolution/aggregate`

© Copyright IBM Corporation 2015

Figure 6-13. REST resources

WB3942.0

Notes:

6.3. Log analysis



© Copyright IBM Corporation 2015

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

9.1

Figure 6-14. Log analysis

WB3942.0

Notes:

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 Started 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 render rules silent

© Copyright IBM Corporation 2015

Figure 6-15. Why are log files important?

WB3942.0

Notes:

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 cluster
 - Only one server for simple configurations: `cisDev`

© Copyright IBM Corporation 2015

Figure 6-16. Where and what to gather

WB3942.0

Notes:

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 caught and makes the condition false

© Copyright IBM Corporation 2015

Figure 6-17. Log analysis

WB3942.0

Notes:

6.4. Insight Inspector

Insight Inspector



© Copyright IBM Corporation 2015

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

9.1

Figure 6-18. Insight Inspector

WB3942.0

Notes:

A debugging tool

- The Insight Inspector is a debugging tool for solution developers
- Solution developers need to:
 - Verify the solution
 - Analyze the data in events and entities
 - Troubleshoot failures and errors
 - For example: no emitted events, rules are not fired, unexpected results
- The Insight Inspector is a component of the Insights Runtime feature
 - For non-production server, cisDev

© Copyright IBM Corporation 2015

Figure 6-19. A debugging tool

WB3942.0

Notes:

Web UI

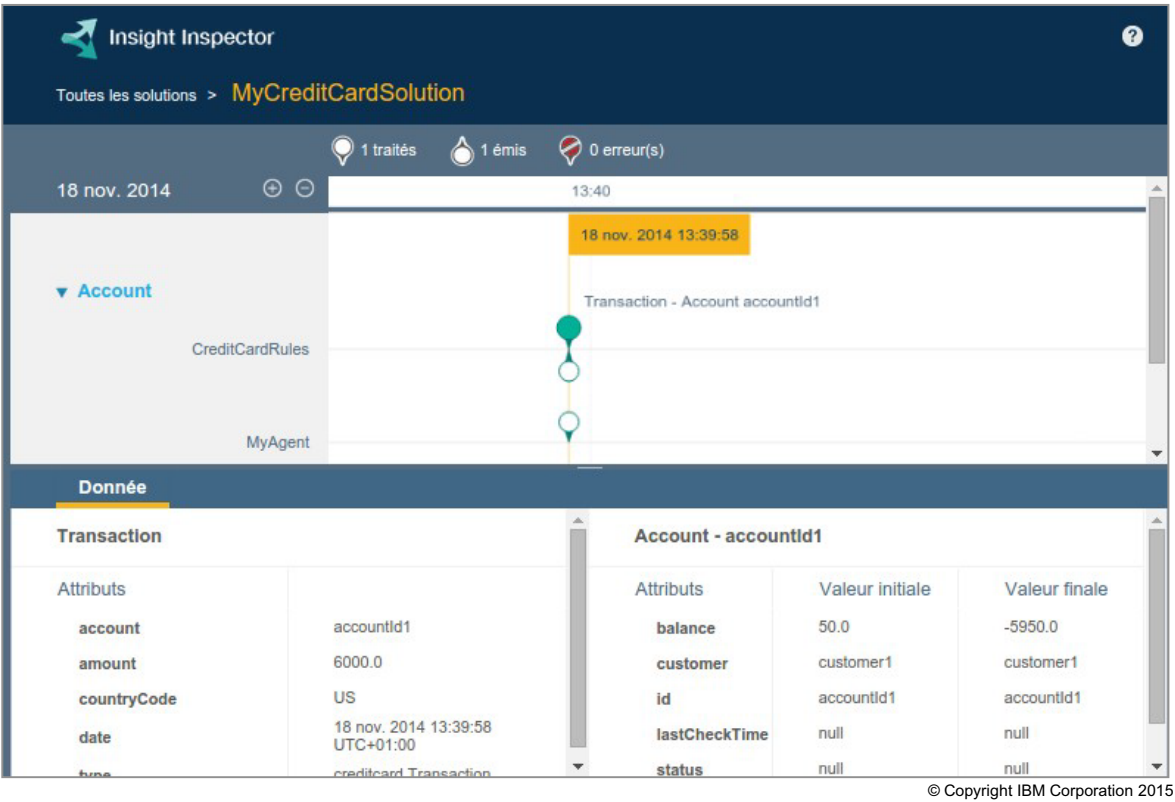


Figure 6-20. Web UI WB3942.0

Notes:

Capabilities

Before:

- Developer accomplishes debugging by using:
 - Java unit testing tool
 - REST API to query entity data
 - DebugReceiver
 - Log files

Now:

- The Insight Inspector provides a visual means to do the same debugging
- The first version of Insight Inspector is limited in its features
 - Inspect processed and emitted events on a timeline, which are grouped by entity or agent
 - Inspect the initialized data of the event
 - Inspect the initial and final values of the entity

© Copyright IBM Corporation 2015

Figure 6-21. Capabilities

WB3942.0

Notes:

How it works

- Start the recording using `TestDriver.startRecording()` or the RESTful API:
`http://[hostname]:[port]/ibm/insights/rest/recording/start/MySolution`
- Create entities and emit events outside Insight Inspector
- Stop the recording using `TestDriver.startRecording()` or the RESTful 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`

© Copyright IBM Corporation 2015

Figure 6-22. How it works

WB3942.0

Notes:

6.5. Generic Client

Generic Client



© Copyright IBM Corporation 2015

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

9.1

Figure 6-23. Generic Client

WB3942.0

Notes:

Highlights

- This generic client provides a user interface that you can use to interact with any existing Decision Server Insights solution for:
 - Demonstrations
 - Learning
 - Debugging
- It uses introspection to discover the entity model and the event model, so you do not have to work too much to use it with your existing solution
- Generic Client was written in Java and uses the TestDriver API
- Generic Client is not part of the product

© Copyright IBM Corporation 2015

Figure 6-24. Highlights

WB3942.0

Notes:

Plugging the generic client to your own solution (1 of 3)

- Assume that your solution project is called:
`alarm_management_solution`
- To plug the generic client to this solution, you must perform the following tasks:
 - Create a Java project that is called `alarm_management_client`
 - Java Build Path:
 - Projects: `generic_client` and `alarm_management_solution - Java Interfaces`
 - Libraries: All JAR files in `<InstallDir>/runtime/ia/gateway` and all JAR files in `<InstallDir>/runtime/wlp/clients`
 - Copy `testdriver.properties` from `banking_scenario_client` to `alarm_management_client`
 - Check regarding `testdriver.properties`
 - Create a main class that is called `AlarmManagementDemo`, following `BankingScenarioClient` as a template

© Copyright IBM Corporation 2015

Figure 6-25. Plugging the generic client to your own solution (1 of 3)

WB3942.0

Notes:

Plugging the generic client to your own solution (2 of 3)

- Create `AlarmManagementEventCatalog` to extend the abstract class `EventAdapter`

It defines:

- The event types that you want to send
- For each event type that is sent:
 - What fields are calculated (read only) as opposed to editable
 - What fields should be displayed
 - What processing you want to apply to an event before it is sent
 - Typically, you want to calculate calculated fields from editable fields
- How to display the events that the solution is emitting: What color and what text

© Copyright IBM Corporation 2015

Figure 6-26. Plugging the generic client to your own solution (2 of 3)

WB3942.0

Notes:

Plugging the generic client to your own solution (3 of 3)

- Create `AlarmManagementScenario` to extend the abstract class `EntityAdapter`

It defines:

- `getRootEntityCatalog`
 - The catalog of root entities (the top-level folders in the upper pane, and the root entities each top-level folder contains)
 - Frequently, you want each root folder to correspond to an entity type, and to contain all the entities of this type
 - In this case, follow `BankingScenario`
 - However, you can decide to organize the top-level folders differently, for example one per use case
- `createEntities`
 - The entities that you want to create programmatically every time you launch the client
- It is a good practice to split the scenario class in two levels:
 - `AlarmManagementScenario` implementing `getRootEntityCatalog` and factory methods such as `createClient` in `BankingScenario`
 - One or more subclasses, such as `AlarmManagementScenario1` that implement different versions of `createEntities`

© Copyright IBM Corporation 2015

Figure 6-27. Plugging the generic client to your own solution (3 of 3)

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Test solutions with the TestDriver API
- Troubleshoot with the REST API
- Troubleshoot with log analysis
- Use Insight Inspector
- Create testing exercises with Generic Client

© Copyright IBM Corporation 2015

Figure 6-28. Unit summary

WB3942.0

Notes:

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.

© Copyright IBM Corporation 2015

Figure 6-29. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

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.

© Copyright IBM Corporation 2015

Figure 6-30. Checkpoint answers

WB3942.0

Notes:

Unit 7. Modeling and defining connectivity

What this unit is about

This unit teaches you how to model and define connectivity for your solution.

What you should be able to do

After completing this unit, you should be able to:

- Describe inbound and outbound connectivity for a solution

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Describe inbound and outbound connectivity for a solution

© Copyright IBM Corporation 2015

Figure 7-1. Unit objectives

WB3942.0

Notes:

Topics

- Event delivery
- Modeling connectivity in your solution
- Defining connectivity for your solution
- Deploying connectivity in your solution

© Copyright IBM Corporation 2015

Figure 7-2. Topics

WB3942.0

Notes:

7.1. Event delivery

Event delivery



© Copyright IBM Corporation 2015

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

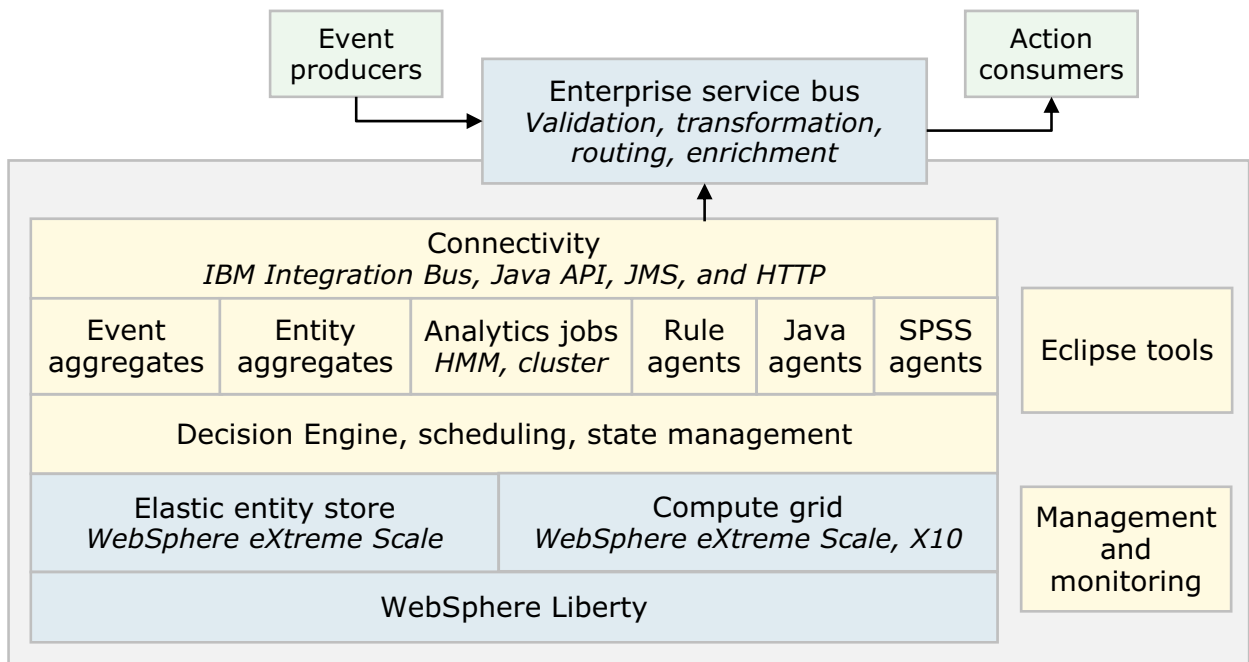
9.1

Figure 7-3. Event delivery

WB3942.0

Notes:

High-level event delivery architecture



© Copyright IBM Corporation 2015

Figure 7-4. High-level event delivery architecture

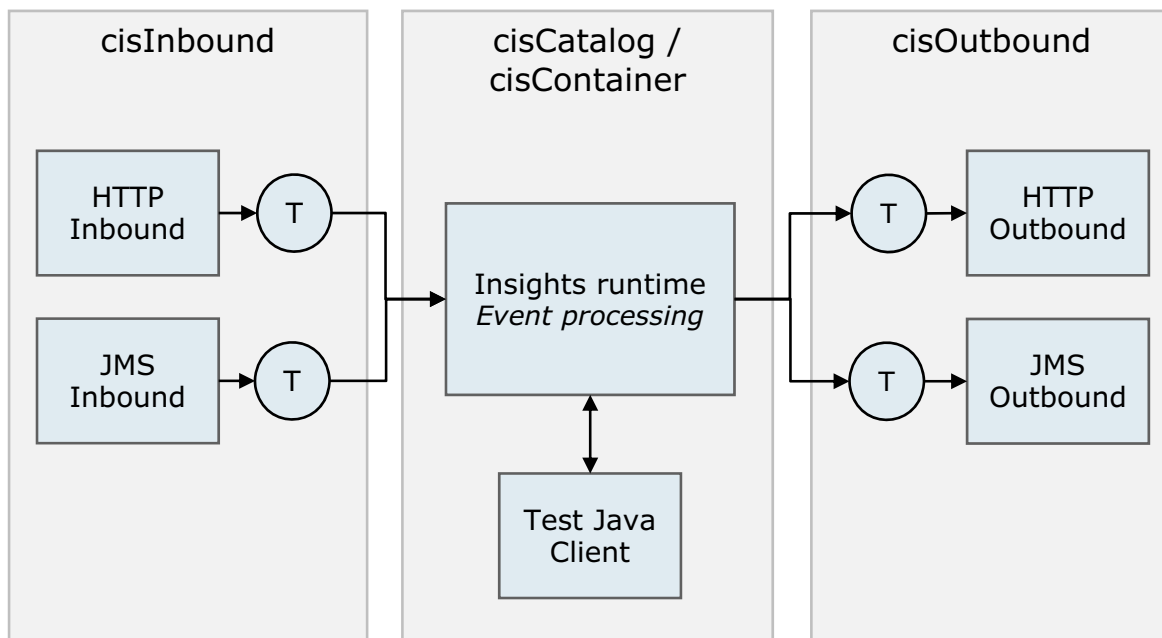
WB3942.0

Notes:

Decision Server Insights uses bindings and endpoints to define how a solution receives (inbound) and sends (outbound) events.

A binding describes the event types that are sent or received, and how they are represented in a message.

Connectivity components



© Copyright IBM Corporation 2015

Figure 7-5. Connectivity components

WB3942.0

Notes:

7.2. Modeling connectivity in your solution

Modeling connectivity in your solution



© Copyright IBM Corporation 2015

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

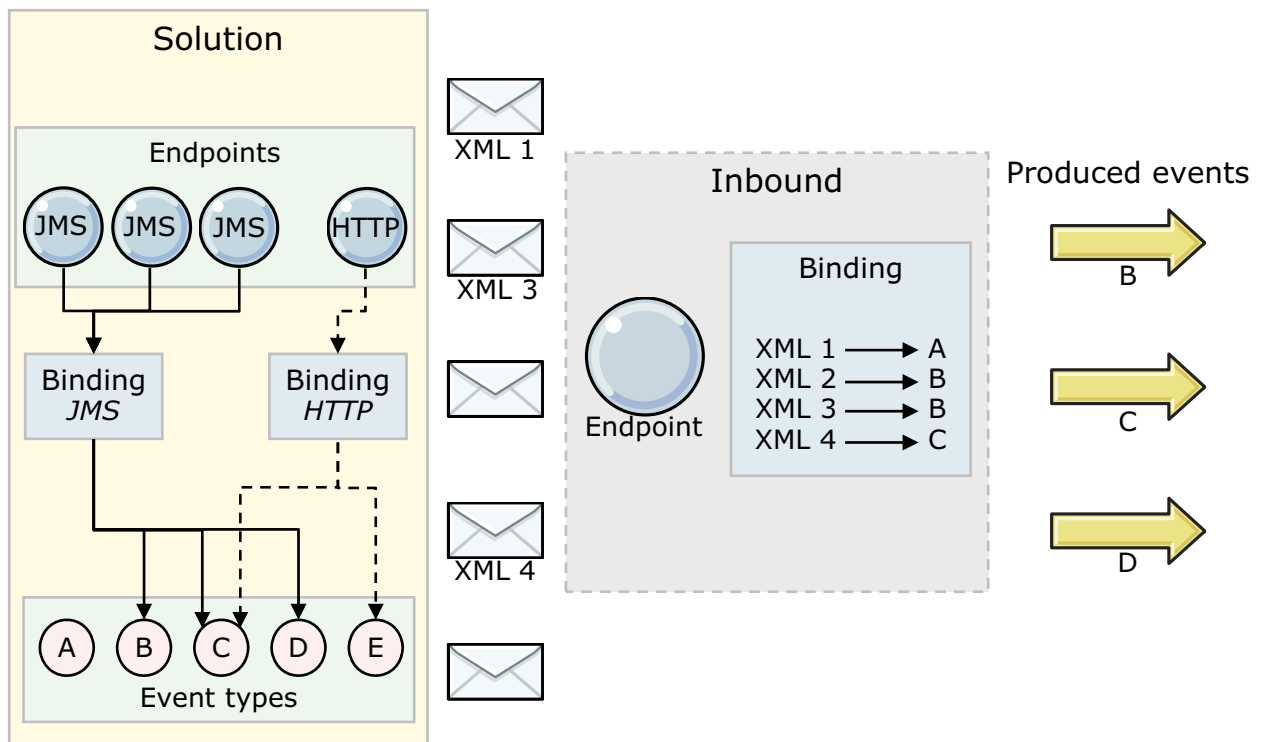
9.1

Figure 7-6. Modeling connectivity in your solution

WB3942.0

Notes:

Event delivery: Inbound connectivity



© Copyright IBM Corporation 2015

Figure 7-7. Event delivery: Inbound connectivity

WB3942.0

Notes:

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.

The endpoint represents the origin of the messages.

This diagram shows four inbound endpoints and the bindings that they reference.

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

© Copyright IBM Corporation 2015

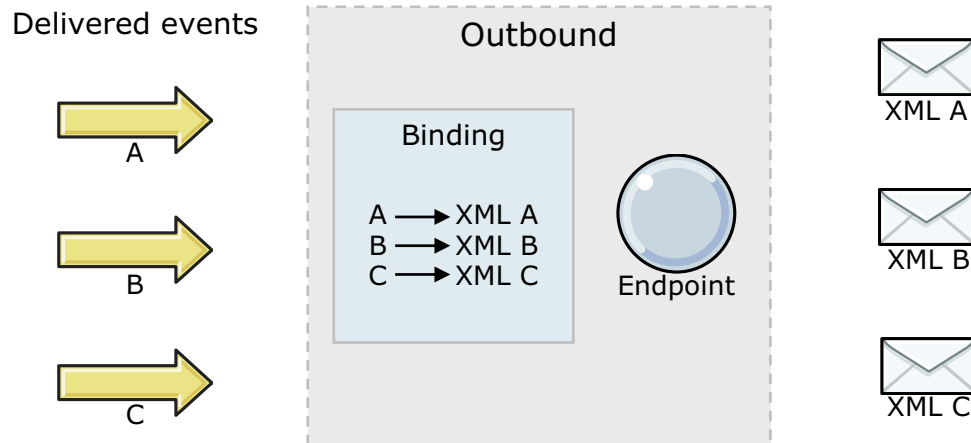
Figure 7-8. Solution gateway for inbound connectivity

WB3942.0

Notes:

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.

Event delivery: Outbound connectivity



© Copyright IBM Corporation 2015

Figure 7-9. Event delivery: Outbound connectivity

WB3942.0

Notes:

Outbound connectivity is the mechanism by which events can be delivered from a solution to the outside world. There is no 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 and destination, or a URL.

7.3. Defining connectivity for your solution

Defining connectivity for your solution



© Copyright IBM Corporation 2015

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

9.1

Figure 7-10. Defining connectivity for your solution

WB3942.0

Notes:

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 by using the wizard
 2. Export and deploy the solution, which now includes connectivity definitions, to the runtime
 3. Export and deploy the connectivity configuration to the server
 4. The solution with the connectivity definition is deployed to the Insights runtime

© Copyright IBM Corporation 2015

Figure 7-11. Defining connectivity in the solution project

WB3942.0

Notes:

When you create a solution in Insight Designer, it uses the solution gateway API for inbound connectivity. The solution can be deployed to an Insights runtime and tested by using a test Java client that implements the solution gateway API, with no requirement for HTTP or JMS binding.

To use inbound or outbound connectivity, you create a connectivity definition file as part of the solution. The definition defines a binding with the type of transport to use, either HTTP or JMS. The binding also defines the 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. The configuration file defines the actual connection details, which can be changed after deployment.

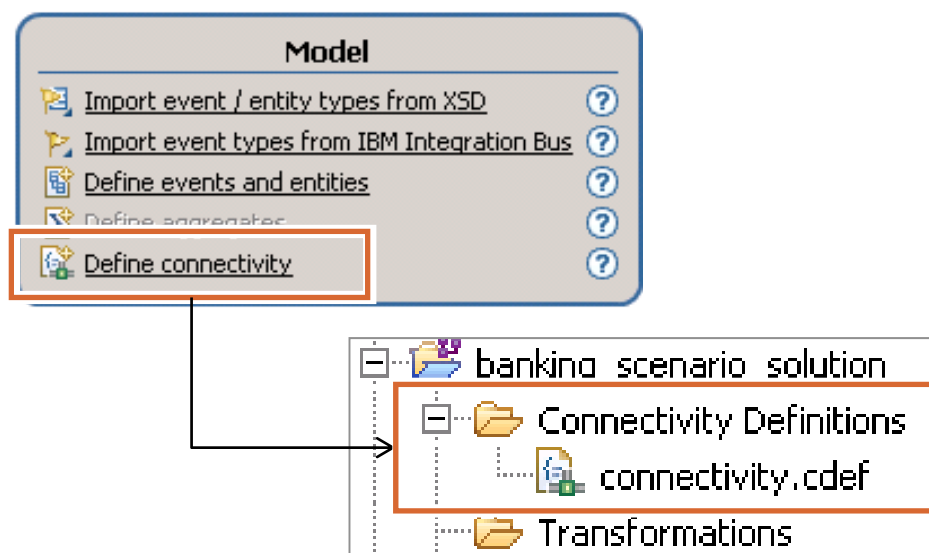
After you define connectivity for the solution, you deploy the solution to the runtime. You use the Export Solution Connectivity Server Configuration wizard to generate a connectivity configuration XML file.

To deploy the configuration, you use the `connectivityManager` command-line script to deploy 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 runtime.

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



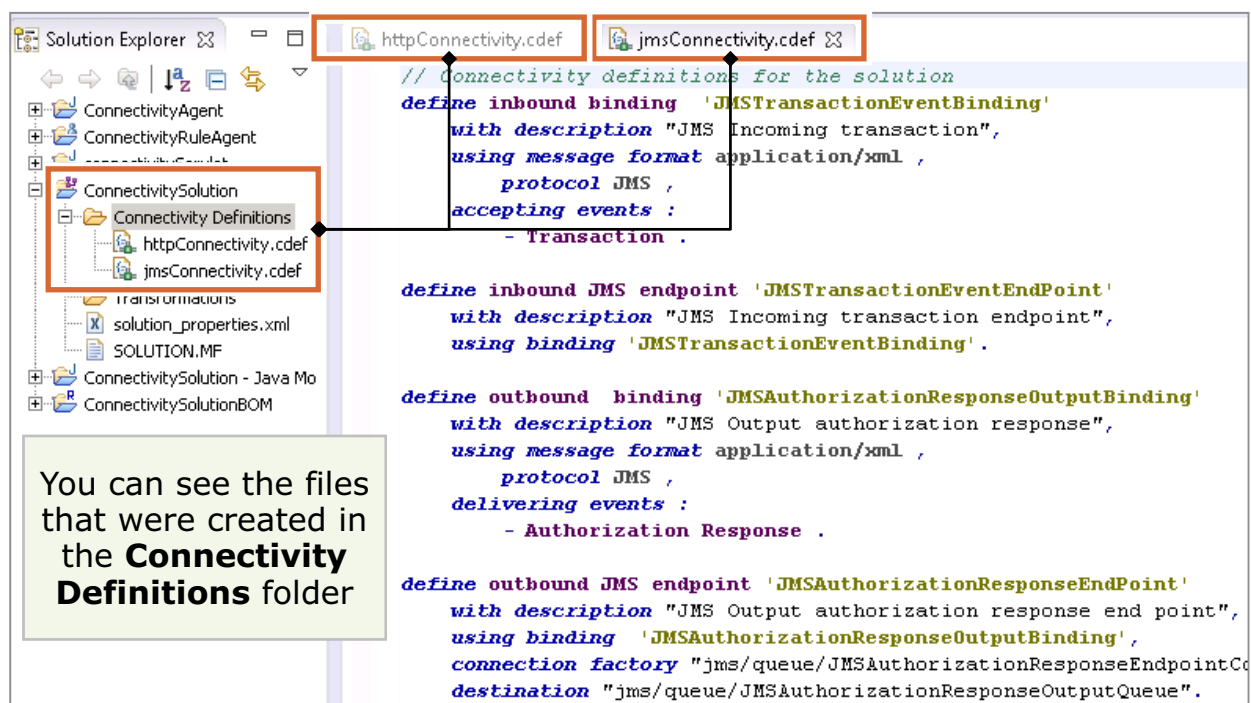
© Copyright IBM Corporation 2015

Figure 7-12. Define connectivity in your solution

WB3942.0

Notes:

Export solution connectivity server configuration



© Copyright IBM Corporation 2015

Figure 7-13. Export solution connectivity server configuration

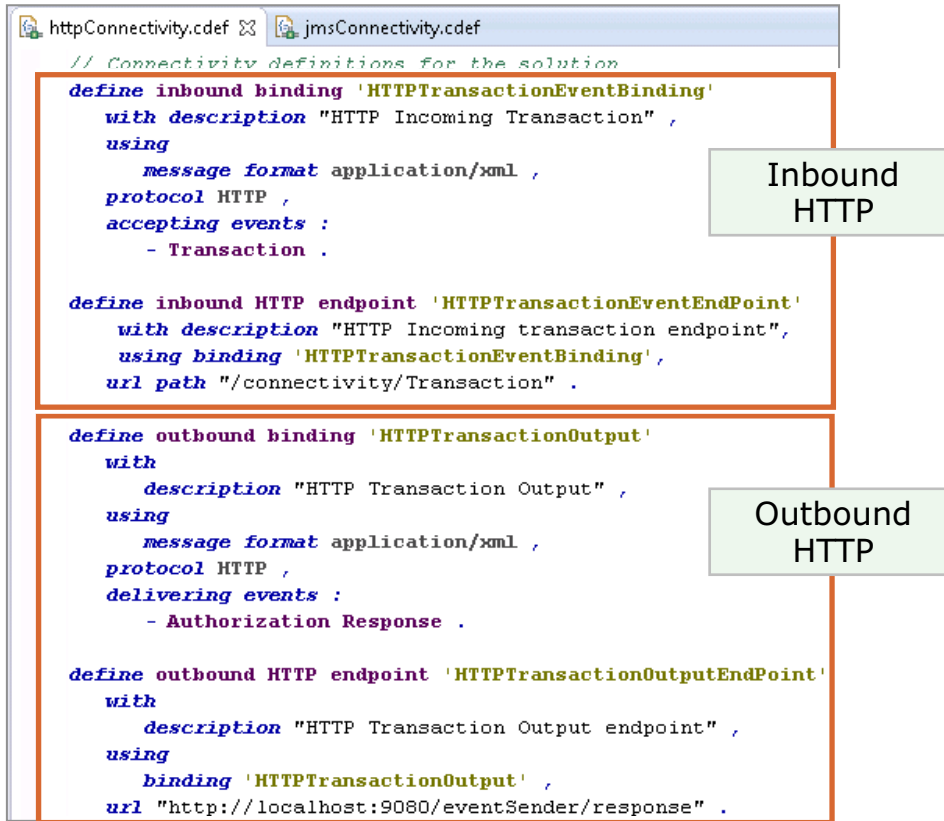
WB3942.0

Notes:

For this example, two files are created, one for HTTP and one for JMS binding.

In this example, the inbound and outbound definitions are combined into one file. If you have separate inbound and outbound servers, you must create two server configuration files from these definition files.

Create the connectivity definitions



© Copyright IBM Corporation 2015

Figure 7-14. Create the connectivity definitions

WB3942.0

Notes:

To define the file, the editor can prompt you for each piece of information that is required by pressing Ctrl+Space while typing.

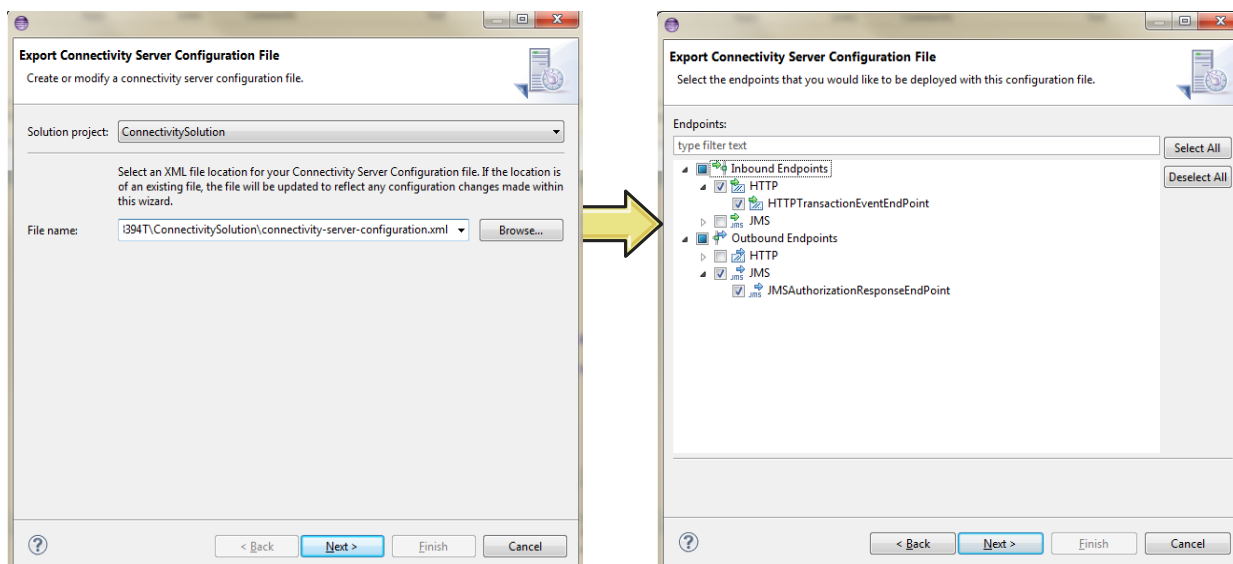
You must specify the following information:

- 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)

If you are defining an HTTP inbound endpoint, you must provide a URL path on which this inbound endpoint receives messages and that contains at least two levels. A URL path is not required if you are defining a JMS inbound endpoint.

In this example, the inbound connectivity is bound to the “Transaction” event, with a description: “HTTP Incoming Transaction”. The message format is `application/xml` and uses the HTTP protocol. The URL for the inbound endpoint uses path `/connectivity/Transaction`.

Export solution connectivity server configuration



© Copyright IBM Corporation 2015

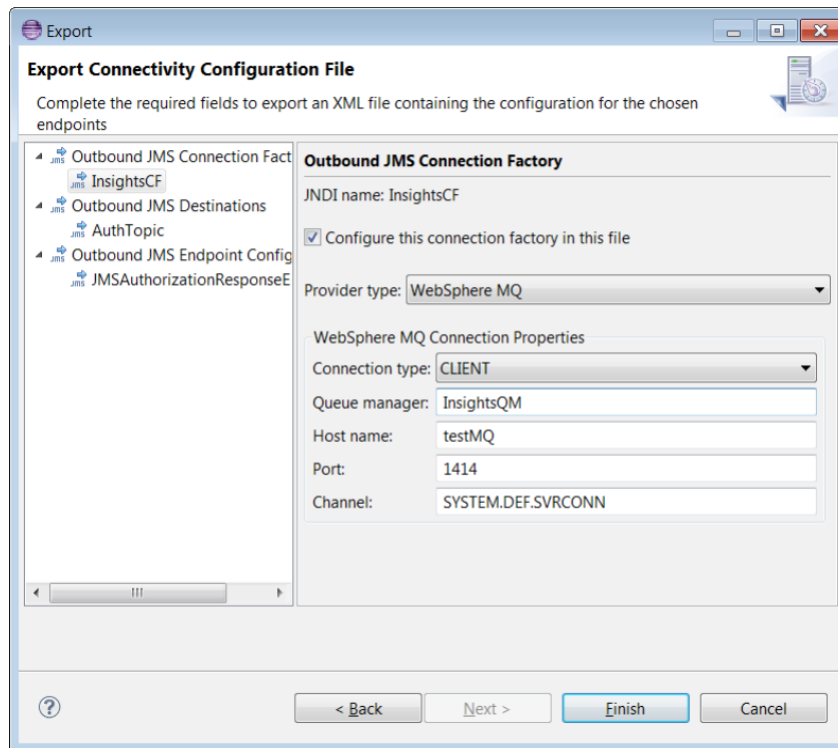
Figure 7-15. Export solution connectivity server configuration

WB3942.0

Notes:

You are prompted to give a file name to save the configuration. Next, you see a list of the inbound and outbound endpoints. You select the endpoints that you want to include.

Create the connectivity definitions



© Copyright IBM Corporation 2015

Figure 7-16. Create the connectivity definitions

WB3942.0

Notes:

The wizard provides a page for you to enter the configuration details. Most of the information is related to JMS, but it can also be used to override the URL for an outbound HTTP endpoint.

After the configuration file is exported, and the solution is deployed, you can deploy the connectivity configuration.

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
 - For example:

```
<variable name="wmqJmsClient.rar.location"
value="C:/wmq/wmq.jmsra.rar"/>
```
 - Note: The `server.xml` file has commented-out sections that describe what needs to be enabled for the two JMS providers

© Copyright IBM Corporation 2015

Figure 7-17. Providers for JMS

WB3942.0

Notes:

7.4. Deploying connectivity in your solution

Deploying connectivity in your solution



© Copyright IBM Corporation 2015

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

9.1

Figure 7-18. Deploying connectivity in your solution

WB3942.0

Notes:

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 <path>/solutionFeature  
<path>/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
 - server.xml in the root directory is updated to include the config file

© Copyright IBM Corporation 2015

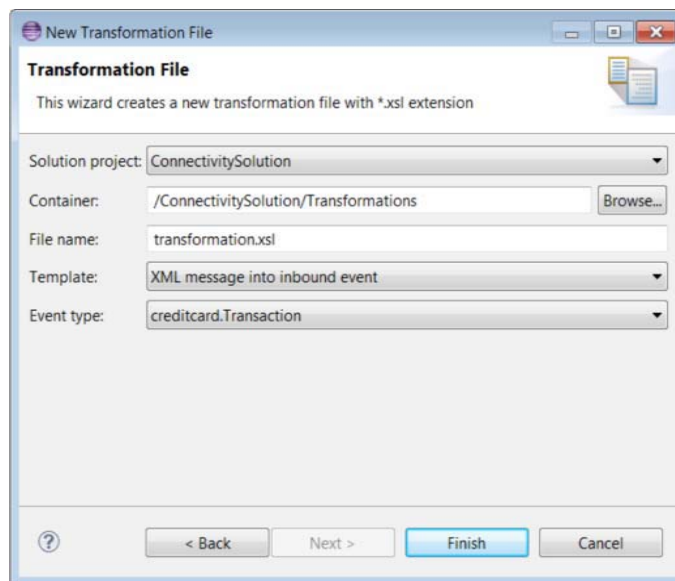
Figure 7-19. Deploying a connectivity configuration

WB3942.0

Notes:

Transformations

- Use transformations to send an arbitrary XML message to the channel
 - The message can then be translated into an XML message that can be understood by Decision Server Insights
- You can also transform an outbound event into a different XML file to be sent to the receiver



© Copyright IBM Corporation 2015

Figure 7-20. Transformations

WB3942.0

Notes:

To create the transformation file, use the **File > New Transformation File** menu.

You enter the details and select whether you want to transform inbound or outbound events. A separate file is required for each transformation.

The wizard generates a template file that is modified to do the XSLT transformation.

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 .
```

© Copyright IBM Corporation 2015

Figure 7-21. Classifying inbound messages for transformation

WB3942.0

Notes:

Troubleshooting

- 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
- Outbound Buffer Manager
 - Use the Outbound Buffer Manager to look at whether you have pending outbound events that cannot be sent for some reason
 - You can also use the Outbound Buffer Manager to clear the events for an endpoint if required

© Copyright IBM Corporation 2015

Figure 7-22. Troubleshooting

WB3942.0

Notes:

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 send into Insights on the Inbound Channel creates events that are to be sent through the Outbound Channels
- To check that the events are being processed correctly by the server, it is possible to use a test client
 - Using a test client removes the channels from the path and determines whether events are being handled correctly
 - The IBM Knowledge Center for this product has more information on developing a test client

© Copyright IBM Corporation 2015

Figure 7-23. Common issues and resources

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Describe inbound and outbound connectivity for a solution

© Copyright IBM Corporation 2015

Figure 7-24. Unit summary

WB3942.0

Notes:

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: Decision Server Insights can classify and transform inbound messages through transformation.

© Copyright IBM Corporation 2015

Figure 7-25. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

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: Decision Server Insights can classify and transform inbound messages through transformation

Answer: True.

© Copyright IBM Corporation 2015

Figure 7-26. Checkpoint answers

WB3942.0

Notes:

Exercise 11

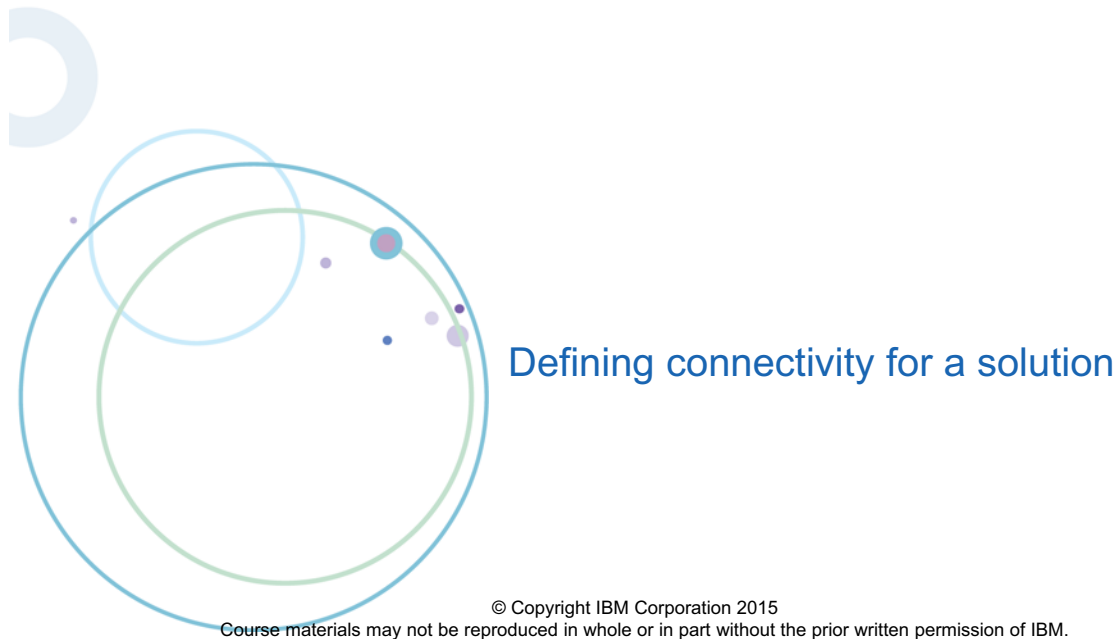


Figure 7-27. Exercise 11

9.1
WB3942.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

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

© Copyright IBM Corporation 2015

Figure 7-28. Exercise objectives

WB3942.0

Notes:

Unit 8. Integrating Decision Server Insights

What this unit is about

This unit explores the integration capabilities of Decision Server Insights.

What you should be able to do

After completing this unit, you should be able to:

- 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 IIB monitoring events
- Create a predictive scoring agent

How you will check your progress

- Checkpoint

Unit objectives

After completing this unit, you should be able to:

- 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 IIB monitoring events
- Create a predictive scoring agent

© Copyright IBM Corporation 2015

Figure 8-1. Unit objectives

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-2. Topics

WB3942.0

Notes:

8.1. Overview of integration requirements

Overview of integration requirements



© Copyright IBM Corporation 2015

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

9.1

Figure 8-3. Overview of integration requirements

WB3942.0

Notes:

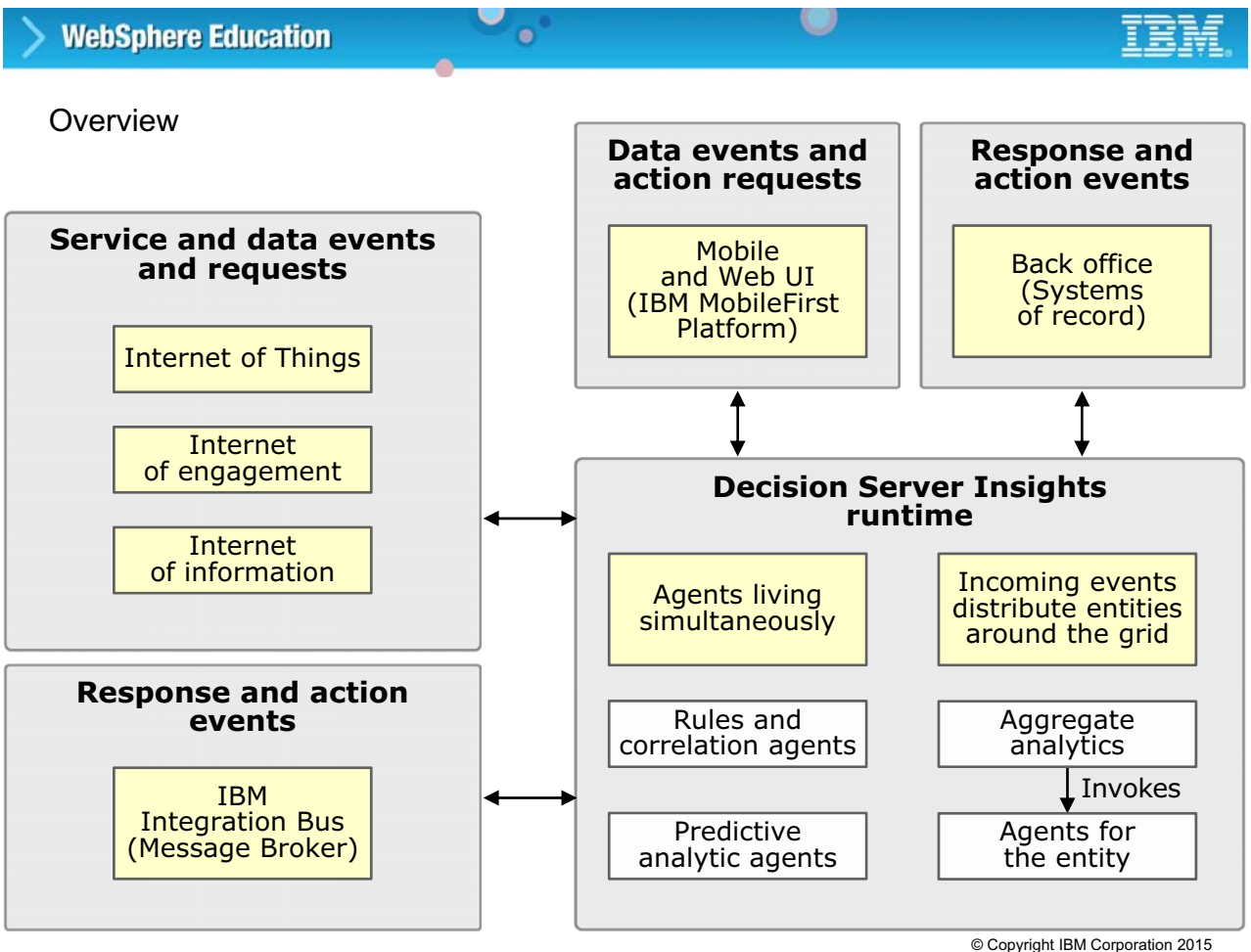


Figure 8-4. No title

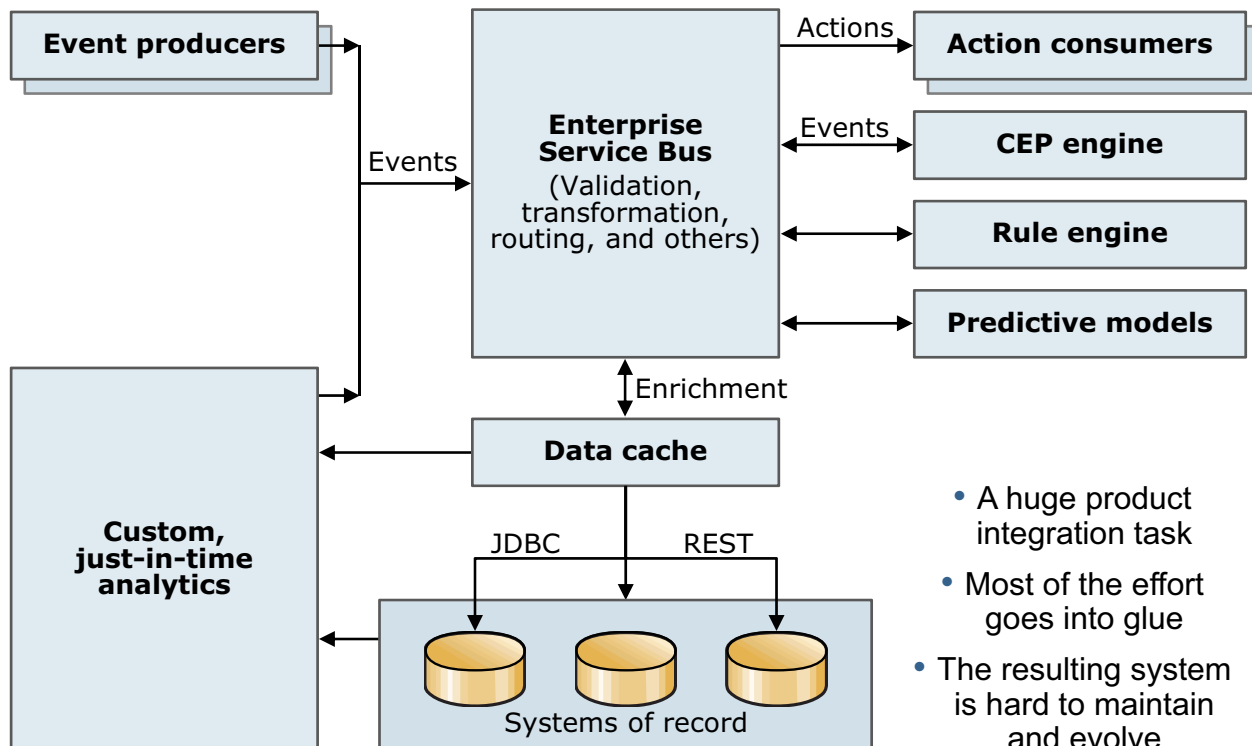
WB3942.0

Notes:

The overview of Decision Server Insights integration capabilities:

- Listens to anything and everything
- Brings information, requests, and action logic to the data
- Maintains wide-angle view of primary entity transactions
- Alerts systems to risk and opportunity
- Continuously updates Decision Server Insights at all scales
- Replicates data to permanent storage
- Maintains continuous availability
 - Hardware and software failures
 - Solution changes

Rolling out a Decision Server Insights solution today



- A huge product integration task
- Most of the effort goes into glue
- The resulting system is hard to maintain and evolve

© Copyright IBM Corporation 2015

Figure 8-5. Rolling out a Decision Server Insights solution today

WB3942.0

Notes:

8.2. Submitting events from an Integration Bus message flow

Submitting events from an Integration Bus message flow



© Copyright IBM Corporation 2015

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

9.1

Figure 8-6. Submitting events from an Integration Bus message flow

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-7. Submitting events from an Integration Bus message flow

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

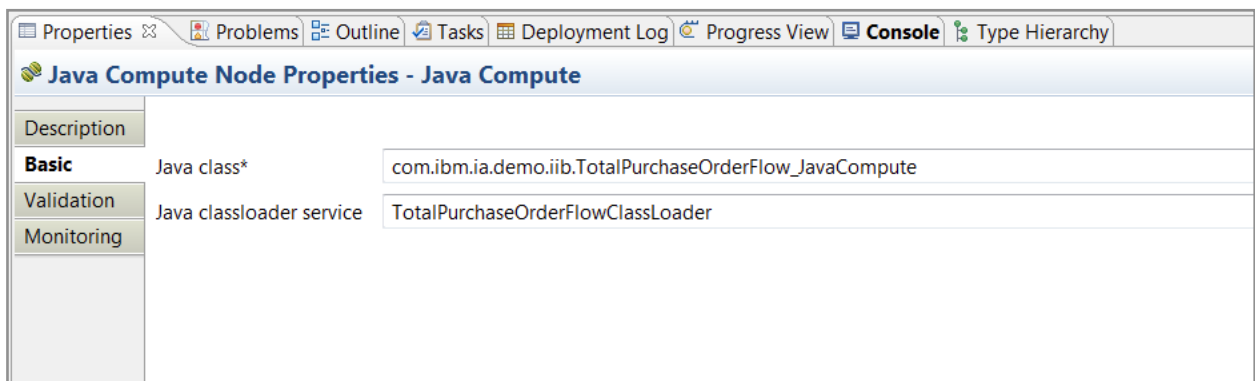
Figure 8-8. Events model

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

Figure 8-9. Java Compute Node

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-10. Deploying IBM Integration Bus message flows

WB3942.0

Notes:

8.3. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights



© Copyright IBM Corporation 2015

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

9.1

Figure 8-11. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

WB3942.0

Notes:

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 can be understood by the Insights runtime
 - Configures a JMS binding connectivity definition, containing all the necessary classifiers to process incoming events into the solution

© Copyright IBM Corporation 2015

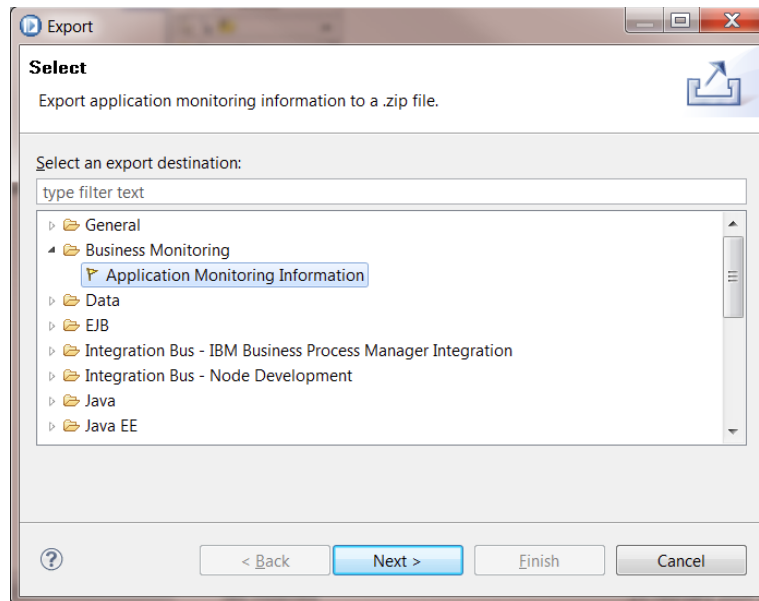
Figure 8-12. Consuming WebSphere Message Broker or IBM Integration Bus monitoring events within Decision Server Insights

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

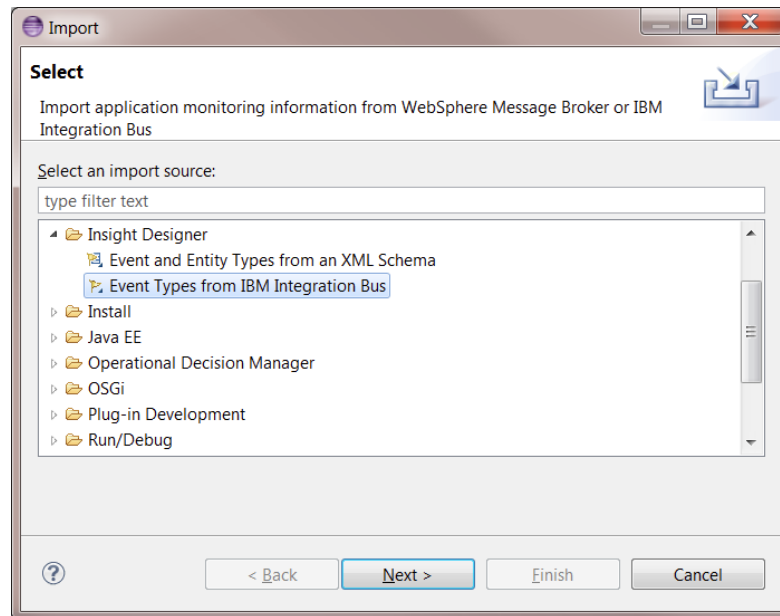
Figure 8-13. Step 1

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

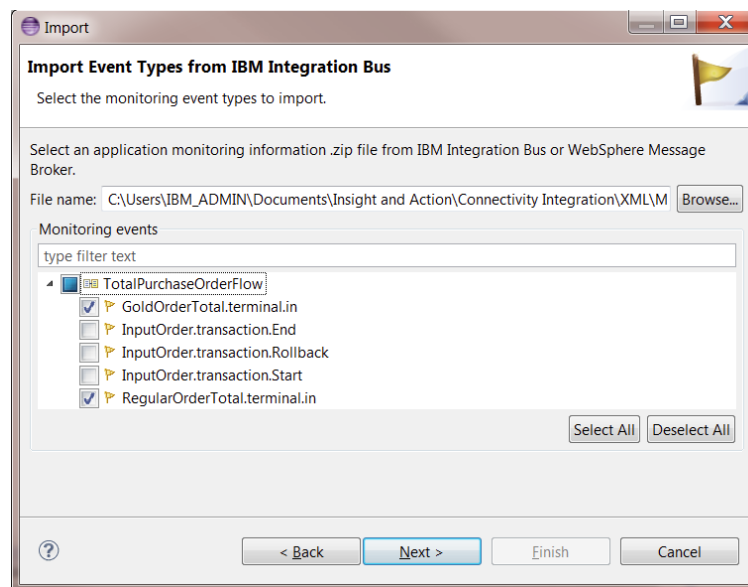
Figure 8-14. Step 2

WB3942.0

Notes:

Step 3

- Select the **Application Monitoring Information** file, which was exported from the IBM Integration Bus or WebSphere Message Broker
- The wizard then lists the events that it finds within the file, so that the ones that should be imported can be selected



© Copyright IBM Corporation 2015

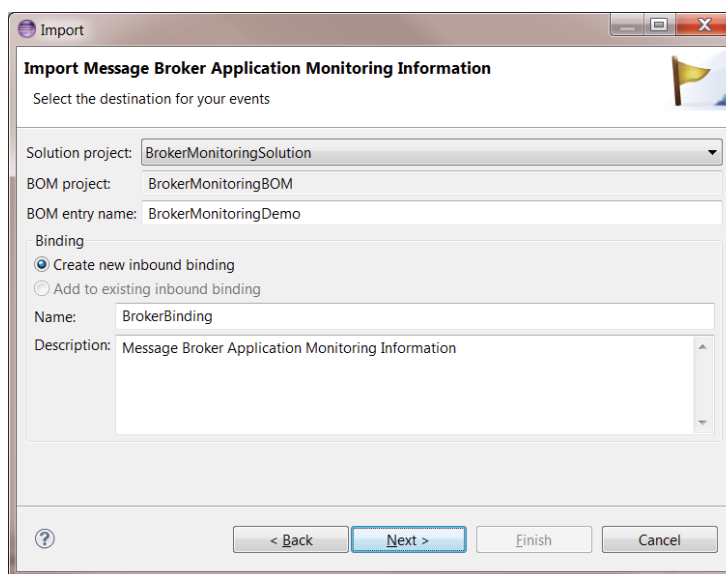
Figure 8-15. Step 3

WB3942.0

Notes:

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



© Copyright IBM Corporation 2015

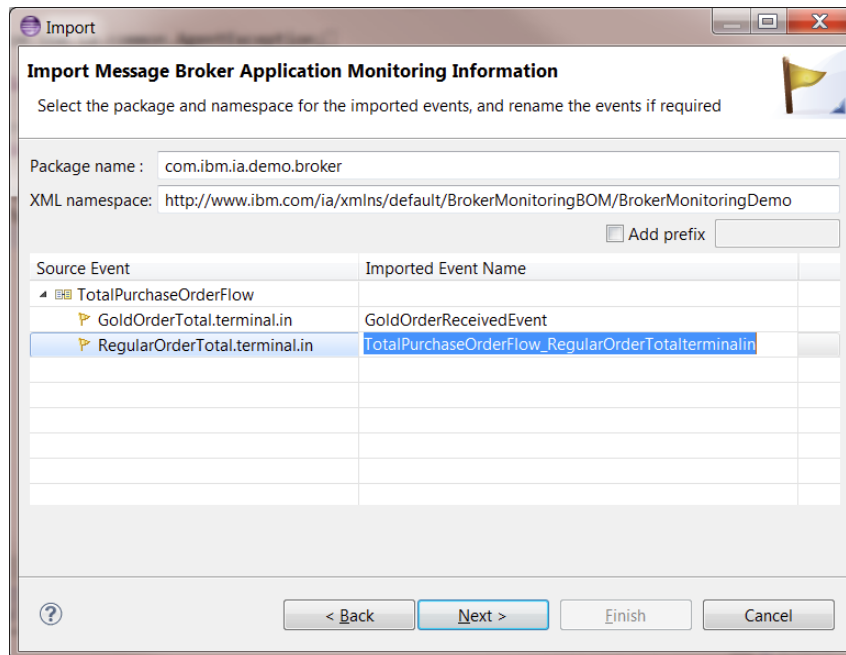
Figure 8-16. Step 4

WB3942.0

Notes:

Step 5

- On the third page, you can specify the package for the events and the namespace for the transformed events



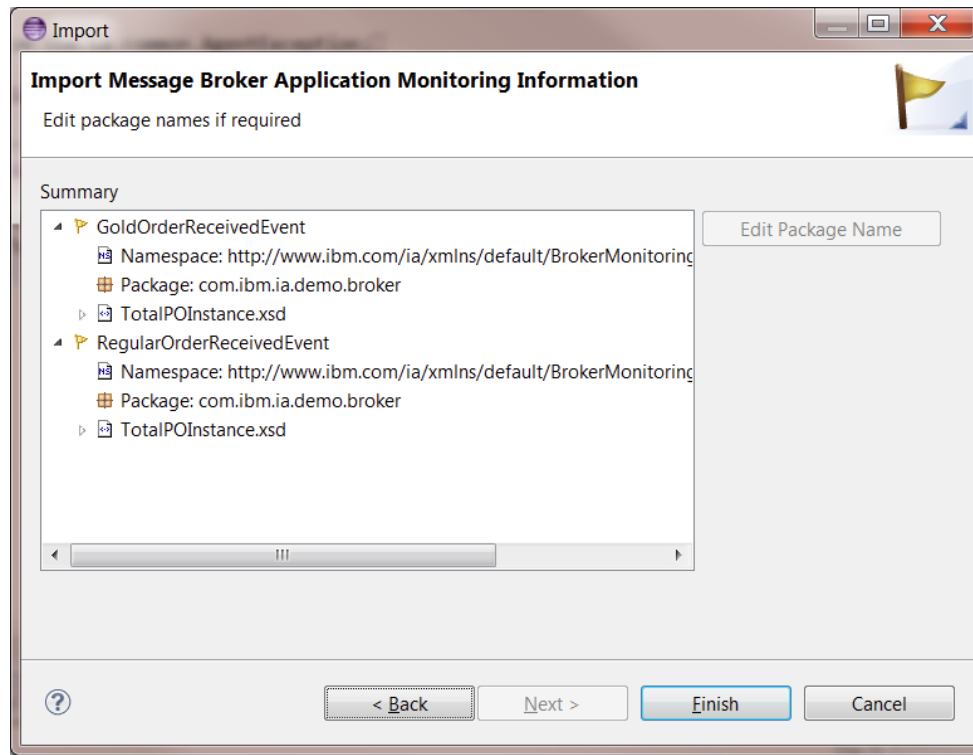
© Copyright IBM Corporation 2015

Figure 8-17. Step 5

WB3942.0

Notes:

Summary of the events to be imported



© Copyright IBM Corporation 2015

Figure 8-18. Summary of the events to be imported

WB3942.0

Notes:

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 (`.cddef`) 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

© Copyright IBM Corporation 2015

Figure 8-19. The result

WB3942.0

Notes:

8.4. Integration with SPSS

Integration with SPSS



© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9.1

Figure 8-20. Integration with SPSS

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-21. SPSS

WB3942.0

Notes:

Predictive analytics

- Integration is supported by 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

© Copyright IBM Corporation 2015

Figure 8-22. Predictive analytics

WB3942.0

Notes:

SPSS integration is supported through Decision Server Insights predictive scoring agents. A predictive scoring agent is a Java agent that can retrieve scores from predictive models on an external SPSS scoring server.

- Invoke scoring on a remote SPSS Scoring Server.
- Contain generated Java code to simplify user task of writing the code.
- Use the results from predictive model to make better decisions.

Decision Server Insights has built-in features that support connections between predictive scoring agents and SPSS Scoring servers without additional boot-strapping.

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

© Copyright IBM Corporation 2015

Figure 8-23. How it works

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-24. Creating a predictive scoring agent

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-25. Troubleshooting

WB3942.0

Notes:

Limitations: SPSS is included separately.

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 WDT 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
 - Storing the scores in the bound entity and making sure that the predictive scoring agent has a higher priority than the subsequent agents

© Copyright IBM Corporation 2015

Figure 8-26. Suggested practices

WB3942.0

Notes:

Integration with ODM Decision Server Insights supports SPSS Collaboration and Deployment Services versions 5 and 6.

8.5. OSGi services

OSGi services



© Copyright IBM Corporation 2015

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

9.1

Figure 8-27. OSGi services

WB3942.0

Notes:

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

© Copyright IBM Corporation 2015

Figure 8-28. OSGi Services

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- 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 IIB monitoring events
- Create a predictive scoring agent

© Copyright IBM Corporation 2015

Figure 8-29. Unit summary

WB3942.0

Notes:

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 8-30. Checkpoint questions

WB3942.0

Notes:

Write your answers here:

1.

2.

3.

Checkpoint 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.

© Copyright IBM Corporation 2015

Figure 8-31. Checkpoint answers

WB3942.0

Notes:

Unit 9. Course summary

What this unit is about

This unit summarizes the course and provides information for future study.

What you should be able to do

After completing this unit, you should be able to:

- 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

Unit objectives

After completing this unit, you should be able to:

- 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

© Copyright IBM Corporation 2015

Figure 9-1. Unit objectives

WB3942.0

Notes:

Course learning objectives

After completing this course, you should be able to:

- 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

© Copyright IBM Corporation 2015

Figure 9-2. Course learning objectives

WB3942.0

Notes:

To learn more on the subject (1 of 2)

- For more information about Decision Server Insights and decision management
 - www.ibm.com/support/knowledgecenter/SSQP76_8.7.0/kc_welcome_odmV.html
 - Kolban's Book on IBM Decision Server Insights
<http://neilkolban.com/ibm/decision-server-insights>
- For more information about decision management:
 - www.ibm.com/software/decision-management
 - www.ibm.com/support/knowledgecenter/SSQP76_8.7.0/com.ibm.odm.distrib/kc_welcome_odm_distrib.html

© Copyright IBM Corporation 2015

Figure 9-3. To learn more on the subject (1 of 2)

WB3942.0

Notes:

To learn more on the subject (2 of 2)

- IBM Training website:
`www.ibm.com/training`
- Other IBM websites:
 - `www.ibm.com/software/websphere`
 - `www.redbooks.ibm.com`
 - `www.ibm.com/developerworks`

© Copyright IBM Corporation 2015

Figure 9-4. To learn more on the subject (2 of 2)

WB3942.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- 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

© Copyright IBM Corporation 2015

Figure 9-5. Unit summary

WB3942.0

Notes:

Appendix A. List of abbreviations

ABRD	agile business rule development
API	application programming interface
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
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	Hyper Text Markup Language
HTDS	hosted transparent decision service
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
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
XU	Execution Unit
z/OS	Z Series Operating System

Appendix B. Resource guide

Completing this WebSphere Education course is a great first step in building your WebSphere, CICS, and SOA skills. Beyond this course, IBM offers several resources to keep your WebSphere 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**
 - You can demonstrate to your employer or clients your new WebSphere, CICS, or SOA mastery through achieving IBM Professional Certification. WebSphere 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 WebSphere products and roles, including developers and administrators.
 - For more information, see: <http://www.ibm.com/services/learning/sites.wss/us/en?pageType=page&c=a0003096>

Social media links

You can keep in sync with WebSphere Education, including new courses and certifications, course previews, and special offers, by visiting any of the following social media websites.

- **Twitter**
 - Receive short and concise updates from WebSphere Education a few times each week.
 - Follow WebSphere Education at: twitter.com/websphere_edu

- **Facebook:**
 - Become a fan of 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:**
 - Visit the IBM Training YouTube channel to learn about IBM training programs and courses.
 - Find IBM Training at: youtube.com/IBMTraining

Support

- **WebSphere Support portal**
 - The WebSphere Support website provides access to a portfolio of support tools. From the WebSphere Support website, you can access several downloads, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARs). To collaboratively solve issues, the support website is a clearing house of links to online WebSphere communities and forums. The IBM support website is now customizable so you can add and delete portlets to the information most important to the WebSphere products 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>
- **WebSphere 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 WebSphere Education course.
 - For more information, see:
<http://www.ibm.com/software/info/education/assistant/>

WebSphere 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 WebSphere 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**

- Information centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The information 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 WebSphere products. The WebSphere information center and library are located conveniently in the left navigation on WebSphere 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 WebSphere zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
- For more information, see: <http://www.ibm.com/developerworks>

WebSphere Services

- IBM Software Services for WebSphere 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 WebSphere 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 WebSphere 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.ibm.com/developerworks/websphere/services/>

