

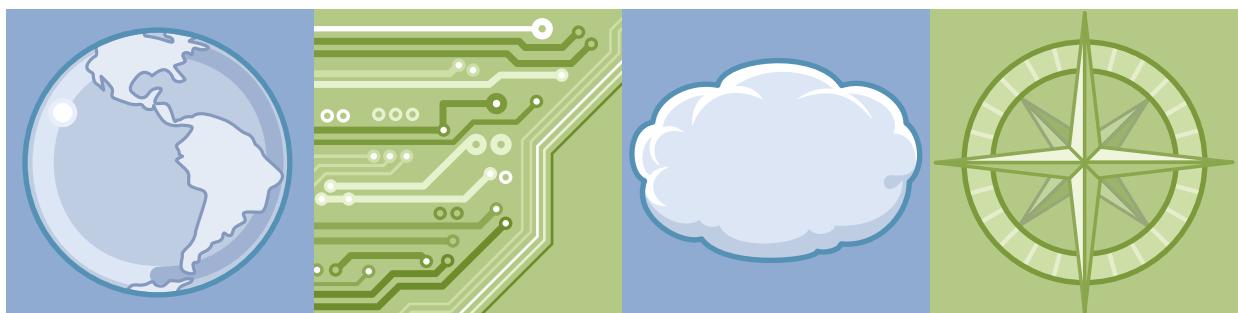


IBM Training

Student Notebook

Administering IBM Operational Decision Manager V8.7.1

Course code WB393 / ZB393 ERC 1.0



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®	DB2®	developerWorks®
Express®	IBM SmartCloud®	ILOG®
Insight™	Insights™	Orchestrate®
PartnerWorld®	PureApplication®	Redbooks®
SPSS®	Tivoli®	WebSphere®
z/OS®		

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

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

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

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

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

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

SoftLayer® is a trademark or registered trademark of SoftLayer, Inc., an IBM Company.

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

June 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.

Contents

Trademarks	xv
Course description.....	xvii
Agenda	xix
Unit 1. Introducing Operational Decision Manager for administrators	1-1
Unit objectives	1-2
Topics	1-3
1.1. Operational Decision Manager offerings and components	1-5
Operational Decision Manager offerings and components	1-6
Operational Decision Manager editions	1-7
Operational Decision Manager cloud-based offerings	1-9
Decision Server Rules	1-10
Decision Server Events	1-11
Decision Center	1-12
Decision Server Insights	1-13
1.2. Interaction between components.....	1-15
Interaction between components	1-16
Interaction between components (1 of 2)	1-17
Interaction between components (2 of 2)	1-19
Operational Decision Manager user roles	1-20
Interaction between roles	1-21
Administrator tasks	1-22
Roles and activities	1-23
1.3. Installation overview	1-25
Installation overview	1-26
Operational Decision Manager installation files (1 of 2)	1-27
Operational Decision Manager installation files (2 of 2)	1-28
What you can install: Launchpad or Installation Manager	1-29
Troubleshooting the installation	1-31
1.4. IBM ODM on Cloud	1-33
IBM ODM on Cloud	1-34
Operational Decision Manager deployment targets	1-35
IBM ODM on Cloud overview	1-36
IBM ODM on Cloud: Three runtime environments	1-37
Unit summary	1-38
Checkpoint questions	1-39
Checkpoint answers	1-40
Exercise 1	1-41
Exercise objectives	1-42
Lab environment	1-43
Unit 2. Configuring Operational Decision Manager	2-1
Unit objectives	2-2
Topics	2-3
2.1. Configuring ODM servers	2-5
Configuring ODM servers	2-6

Configuring Operational Decision Manager Java EE and Java SE servers	2-7
2.2. Configuring Rule Execution Server	2-9
Configuring Rule Execution Server	2-10
Configuring Rule Execution Server	2-11
Configuring Rule Execution Server on WebSphere Application Server (1 of 4)	2-12
Configuring Rule Execution Server on WebSphere Application Server (2 of 4)	2-13
Configuring Rule Execution Server on WebSphere Application Server (3 of 4)	2-14
Configuring Rule Execution Server on WebSphere Application Server (4 of 4)	2-15
2.3. Configuring Decision Center	2-17
Configuring Decision Center	2-18
Configuring Decision Center	2-19
Configuring Decision Center on WebSphere Application Server (1 of 3)	2-20
Configuring Decision Center on WebSphere Application Server (2 of 3)	2-21
Configuring Decision Center on WebSphere Application Server (3 of 3)	2-22
2.4. Completing the installation.....	2-23
Completing the installation	2-24
Installation Settings wizard	2-25
Completing the Rule Execution Server installation (1 of 2)	2-26
Completing the Rule Execution Server installation (2 of 2)	2-27
Completing the Decision Center installation (1 of 2)	2-28
Completing the Decision Center installation (2 of 2)	2-29
2.5. Profile templates for WebSphere Application Server.....	2-31
Profile templates for WebSphere Application Server	2-32
Using profile templates in the Profile Management tool	2-33
Decision Server Rules profile template files	2-34
Decision Center profile template files	2-35
Cluster scripts	2-36
Running the Decision Server script	2-37
Running the Decision Center script	2-38
2.6. Configuring Rule Execution Server for production.....	2-39
Configuring Rule Execution Server for production	2-40
Configuring Rule Execution Server in different environments (1 of 2)	2-41
Configuring Rule Execution Server in different environments (2 of 2)	2-42
Setting up Rule Execution Server for production	2-43
2.7. Clustering ODM servers for high availability	2-45
Clustering ODM servers for high availability	2-46
Clustering on WebSphere Application Server	2-47
WebSphere Application Server clustering and the ideal topology	2-48
ODM ideal topology	2-49
Operational Decision Manager cell	2-50
Decision Server ideal topology cell	2-52
Shared environment	2-53
Staged environment	2-55
2.8. IBM ODM Pattern V8.7.1.....	2-57
IBM ODM Pattern V8.7.1	2-58
IBM ODM Pattern	2-59
IBM ODM Pattern 8.7.1 artifacts	2-60
IBM ODM Pattern V8.7.1 contents	2-63
Virtual system patterns	2-64
Example: Predefined system pattern for IBM ODM with DB2 for HADR	2-65
Virtual system instances	2-66

Unit summary	2-67
Checkpoint questions	2-68
Checkpoint answers	2-69
Exercise 2	2-70
Exercise objectives	2-71
Exercise 3	2-72
Exercise objectives	2-73
Exercise 4	2-74
Exercise objectives	2-75
Unit 3. Managing user roles and permissions.....	3-1
Unit objectives	3-2
Topics	3-3
Approaching ODM as an administrator	3-4
Recall: Roles and activities	3-5
3.1. Application server security.....	3-7
Application server security	3-8
Secure mode on the application server	3-9
Security layers in WebSphere Application Server	3-10
Configuring secure mode on the application server	3-12
3.2. Configuring security for ODM modules on WebSphere Application Server	3-13
Configuring security for ODM modules on WebSphere Application Server	3-14
Application security roles and user-defined groups	3-15
Mandatory role definitions to access Rule Execution Server through the console	3-16
Rule Execution Server role definitions	3-17
Mandatory role definitions to access Decision Center through the console	3-18
Mandatory role definitions to access Decision Server Events	3-19
3.3. Configuring single sign-on (SSO)	3-21
Configuring single sign-on (SSO)	3-22
Single sign-on	3-23
3.4. Defining custom groups for Decision Center.....	3-25
Defining custom groups for Decision Center	3-26
Defining custom groups for Decision Center (1 of 2)	3-27
Defining custom groups for Decision Center (2 of 2)	3-29
3.5. Administering security in Decision Center.....	3-31
Administering security in Decision Center	3-32
Adding custom groups to Decision Center database	3-33
Enforcing security on a project	3-34
Fine-grained permissions	3-35
Permission definitions	3-36
Specifying fine-grained permissions	3-37
Permissions in action	3-38
Permissions in action: Example	3-39
3.6. Troubleshooting security in Decision Center.....	3-41
Troubleshooting security in Decision Center	3-42
Issue: Authorization failed	3-43
Issue: Incorrect permissions	3-44
Troubleshooting support	3-45
Unit summary	3-46
Checkpoint questions	3-47
Checkpoint answers	3-48

Exercise 5	3-49
Exercise objectives	3-50
Unit 4. Synchronizing across environments	4-1
Unit objectives	4-2
Topics	4-3
4.1. Collaboration between business and technical users	4-5
Collaboration between business and technical users	4-6
Simultaneous development across environments	4-7
Synchronization between users	4-8
Synchronization workflow	4-10
Synchronizing between Decision Center and Rule Solutions for Office	4-12
4.2. Choosing the master source	4-15
Choosing the master source	4-16
Choosing the master source	4-17
Dedicated synchronization manager	4-18
4.3. Understanding the synchronization mechanism	4-19
Understanding the synchronization mechanism	4-20
Synchronization tools	4-21
Synchronization architecture	4-22
Connection entries	4-23
Side-by-side comparison	4-24
Choosing how to synchronize	4-25
UUIDs	4-26
Synchronization commands	4-27
4.4. What to synchronize	4-29
What to synchronize	4-30
Rule project items (1 of 2)	4-31
Rule project items (2 of 2)	4-32
4.5. Importing and exporting projects with Decision Center	4-35
Importing and exporting projects with Decision Center	4-36
Export and import tools	4-37
Exporting projects	4-38
Importing projects into Decision Center	4-39
Importing projects into Designer	4-41
Unit summary	4-42
Checkpoint questions	4-43
Checkpoint answers	4-44
Exercise 6	4-45
Exercise objectives	4-46
Unit 5. Managing deployment	5-1
Unit objectives	5-2
Topics	5-3
Recall: Administrator tasks	5-4
5.1. Deployment principles	5-5
Deployment principles	5-6
Recall: ODM interaction	5-7
Preparing for deployment	5-8
Ruleset archive	5-9
RuleApps and RuleApp archives	5-10

RuleApp archives	5-12
Ruleset path	5-13
5.2. Deployment options.	5-15
Deployment options	5-16
Deployment options	5-17
Deployment from Rule Designer	5-18
RuleApp projects in Rule Designer (1 of 2)	5-19
RuleApp projects in Rule Designer (2 of 2)	5-20
Managing rulesets in the RuleApp	5-21
Using a Rule Execution Server configuration (1 of 2)	5-22
Using a Rule Execution Server configuration (2 of 2)	5-24
Deployment from Decision Center	5-25
Deployment from Rule Execution Server console	5-26
Deployment with Ant scripts	5-27
5.3. RuleApp and ruleset properties.	5-29
RuleApp and ruleset properties	5-30
Properties	5-31
Deployed RuleApp and ruleset names	5-33
5.4. Version policies for deployment.	5-35
Version policies for deployment	5-36
Version policy options (1 of 3)	5-37
Version policy options (2 of 3)	5-38
Version policy options (3 of 3)	5-39
5.5. Managing XOMs	5-41
Managing XOMs	5-42
Introduction to Java XOM management	5-43
Managed Java XOM elements	5-44
Link from a ruleset to a managed Java XOM element	5-45
Java XOM deployment (1 of 2)	5-46
Java XOM deployment (2 of 2)	5-47
Managed XOM and Decision Center	5-48
Managed XOM, Rule Designer, and Decision Center	5-49
Managed XOM and Decision Validation Services	5-51
Managed XOM storage	5-52
Unit summary	5-53
Checkpoint questions	5-54
Checkpoint answers	5-55
Exercise 7	5-56
Exercise objectives	5-57
Unit 6. Administering Rule Execution Server	6-1
Unit objectives	6-2
Topics	6-3
6.1. Setting up Rule Execution Server	6-5
Setting up Rule Execution Server	6-6
Preparing for deployment to Rule Execution Server	6-7
Administer production enterprise application	6-9
6.2. Rule Execution Server architecture.	6-11
Rule Execution Server architecture	6-12
Rule Execution Server architecture	6-13
Key functions of Rule Execution Server	6-14

Execution stack: Execution components	6-15
Execution Unit	6-17
JMX management and execution model	6-18
Management and monitoring stack: Console (1 of 2)	6-20
Management and monitoring stack: Console (2 of 2)	6-21
Persistence layer	6-22
Ant tasks	6-24
6.3. Managed execution in action	6-25
Managed execution in action	6-26
Elements of a rule-based solution	6-27
Ruleset parsing	6-28
Ruleset execution	6-29
6.4. Platforms for Rule Execution Server	6-31
Platforms for Rule Execution Server	6-32
Rule execution environment	6-33
Possible choices: Introduction	6-34
Decision services: SOA approach	6-36
Hosted transparent decision service	6-37
Monitored transparent decision service	6-38
6.5. Managing and monitoring through the Rule Execution Server console	6-39
Managing and monitoring through the Rule Execution Server console	6-40
Rule Execution Server console features	6-41
System administration through the console	6-42
Rule Execution Server administration artifacts: RuleApps	6-43
Rule Execution Server administration artifacts: RuleApp archives	6-44
Rule Execution Server administration artifacts: Rulesets	6-46
Back up and restore	6-48
6.6. Managing resources with the REST API	6-49
Managing resources with the REST API	6-50
REST API for ruleset execution	6-51
Endpoint URLs	6-52
HTTP methods	6-54
Request and response schema	6-55
WADL representation	6-57
REST API management services	6-58
Unit summary	6-59
Checkpoint questions	6-60
Checkpoint answers	6-61
Exercise 8	6-62
Exercise objectives	6-63
Unit 7. Using Decision Warehouse to audit ruleset execution	7-1
Unit objectives	7-2
Topics	7-3
7.1. Traces	7-5
Traces	7-6
Recall: ODM administrator	7-7
Traces	7-8
7.2. Auditing ruleset execution through Decision Warehouse	7-11
Auditing ruleset execution through Decision Warehouse	7-12
Auditing ruleset execution	7-13

Rule Execution Server console: Decision Warehouse	7-14
Decision Warehouse (1 of 2)	7-15
Decision Warehouse (2 of 2)	7-16
Step 1: Default use of Decision Warehouse	7-17
Step 2: Enable rule execution monitoring	7-18
Step 3: Execute the ruleset	7-20
Step 4a: View stored decision traces	7-21
Step 4b: View decision details and fired rules	7-22
7.3. Optimizing Decision Warehouse	7-23
Optimizing Decision Warehouse	7-24
Ruleset monitoring options	7-25
Filtering trace data (1 of 2)	7-26
Filtering trace data (2 of 2)	7-28
How filters work	7-29
Removing BOM serialization	7-30
Removing BOM serialization	7-31
Reducing performance impact during BOM serialization	7-32
7.4. Customizing Decision Warehouse	7-33
Customizing Decision Warehouse	7-34
Customizing Decision Warehouse	7-35
Persistence properties	7-36
7.5. Troubleshooting	7-37
Troubleshooting	7-38
Decision Warehouse does not show stored trace data after ruleset execution	7-39
Reducing stored execution traces in Decision Warehouse	7-40
Unit summary	7-41
Checkpoint questions	7-42
Checkpoint answers	7-43
Exercise 9	7-44
Exercise objectives	7-45
Unit 8. Monitoring execution and performance.....	8-1
Unit objectives	8-2
Topics	8-3
8.1. Monitoring ruleset execution	8-5
Monitoring ruleset execution	8-6
Monitoring rulesets in Rule Execution Server console	8-7
Monitoring ruleset execution statistics (1 of 2)	8-8
Monitoring ruleset execution statistics (2 of 2)	8-9
Testing ruleset execution	8-10
Logged events on execution units (1 of 2)	8-12
Logged events on execution units (2 of 2)	8-13
8.2. Monitoring the Execution Unit (XU) on WebSphere Application Server	8-15
Monitoring the Execution Unit (XU) on WebSphere Application Server	8-16
Modifying XU configuration properties (1 of 4)	8-17
Modifying XU configuration properties (2 of 4)	8-18
Modifying XU configuration properties (3 of 4)	8-19
Modifying XU configuration properties (4 of 4)	8-20
Specifying XU trace levels	8-21
8.3. Tuning performance	8-23
Tuning performance	8-24

Common Rule Execution Server performance issues	8-25
Rule Execution Server configuration changes to improve performance	8-26
Improving performance: XU connection pool	8-27
Improving performance: Java 2 Security	8-29
Tuning performance: Decision traces	8-30
Tuning performance: Long ruleset execution times	8-31
Tuning performance: High memory use	8-33
Tuning performance: High processor use	8-35
Tuning performance: Hosted transparent decision services	8-36
Tuning performance: Planning for growth	8-38
8.4. Troubleshooting Rule Execution Server	8-39
Troubleshooting Rule Execution Server	8-40
Troubleshooting the Rule Execution Server installation	8-41
Troubleshooting Rule Execution Server performance	8-42
Retrieving the XU dump	8-43
Rule Execution Server database driver issues	8-44
Providing more memory for applications	8-45
Unit summary	8-46
Checkpoint questions	8-47
Checkpoint answers	8-48
Exercise 10	8-49
Exercise objectives	8-50
Performance tuning overview	8-51
Unit 9. Managing baselines and multiple releases	9-1
Unit objectives	9-2
Topics	9-3
Recall: ODM administrator	9-4
9.1. Rule management features	9-5
Rule management features	9-6
Rule management features	9-7
9.2. Smart folders	9-9
Smart folders	9-10
Smart folders for business rule projects	9-11
Smart folders for event projects	9-12
9.3. Version control and history	9-13
Version control and history	9-14
Version control in Decision Center	9-15
Accessing versions in the History menu	9-16
History of changes	9-17
9.4. Baselines	9-19
Baselines	9-20
Project baselines	9-21
Deployment baselines	9-22
Baselines versus deployment baselines	9-23
Restoring baselines	9-24
9.5. Multiple release management	9-25
Multiple release management	9-26
Branches	9-27
Managing multiple versions of a rule project	9-28
Copying rules between branches	9-29

Multiple release management visual support	9-30
Merging branches	9-31
Using Diff and Merge	9-33
Branches and RuleApp deployment	9-34
Branches and permissions	9-35
9.6. Managing decision services in the Enterprise console	9-37
Managing decision services in the Enterprise console	9-38
Decision services and the decision governance framework	9-39
Decision service releases, activities, and snapshots	9-40
Managing decision service changes in the Enterprise console	9-42
Unit summary	9-43
Checkpoint questions	9-44
Checkpoint answers	9-45
Exercise 11	9-46
Exercise objectives	9-47
Unit 10. Introducing Decision Server Insights for administrators	10-1
Unit objectives	10-2
Topics	10-3
10.1. Programming model	10-5
Programming model	10-6
Sense, Build, Decide, Act	10-7
Core building blocks	10-9
Programming model	10-11
Aggregates in the programming model	10-13
10.2. Decision Server Insights architecture	10-15
Decision Server Insights architecture	10-16
Decision Server Insights high-level architecture	10-17
Insight Server	10-19
10.3. Installing Decision Server Insights	10-21
Installing Decision Server Insights	10-22
Installing with Installation Manager	10-23
Installing with Installation Manager console mode	10-24
Installing: Defining the installation path	10-25
Installing: Choosing the product features	10-26
Installing: Choosing the installation type	10-27
Installing: Choosing the installation type	10-28
Installing Decision Server Insights on multiple servers	10-29
Verifying your installation	10-30
Unit summary	10-31
Checkpoint questions	10-32
Checkpoint answers	10-33
Exercise 12	10-34
Exercise objectives	10-35
Exercise environment (1 of 3)	10-36
Exercise environment (2 of 3)	10-37
Exercise environment (3 of 3)	10-38
Unit 11. Configuring Insight Server	11-1
Unit objectives	11-2
Topics	11-3

11.1. WebSphere eXtreme Scale basics	11-5
WebSphere eXtreme Scale basics	11-6
What is IBM WebSphere eXtreme Scale	11-7
Understanding a grid	11-8
Catalog service	11-10
WebSphere eXtreme Scale terminology (1 of 3)	11-11
WebSphere eXtreme Scale terminology (2 of 3)	11-12
WebSphere eXtreme Scale terminology (3 of 3)	11-13
11.2. Topologies	11-15
Topologies	11-16
Designing a topology	11-17
Reference topology goals	11-18
Catalog service	11-19
Containers and partitions (1 of 2)	11-20
Containers and partitions (2 of 2)	11-22
Inbound and outbound connectivity	11-24
Data persistence	11-26
Decision Server Insights reference topology	11-27
11.3. Sizing	11-29
Sizing	11-30
Capacity planning for containers in the grid	11-31
Calculating the required memory for the data in the grid	11-32
Determining number of partitions	11-33
Example: Calculating memory storage	11-34
Example: Determining heap size	11-36
Example: Planning for failover	11-38
Example: Number of partitions	11-39
Example: Sizing results	11-40
Example: Resulting grid	11-41
11.4. Configuring a production topology	11-43
Configuring a production topology	11-44
Steps for creating and configuring a production server topology	11-45
Server templates	11-46
Customizing servers	11-47
Configuring security (1 of 2)	11-48
Configuring security (2 of 2)	11-49
Configuring heap size for container servers	11-50
Starting the servers	11-51
Unit summary	11-52
Checkpoint questions	11-53
Checkpoint answers	11-54
Exercise 13	11-55
Exercise objectives	11-56
Course topology: 4 VMware images	11-57
Unit 12. Managing deployment for Decision Server Insights	12-1
Unit objectives	12-2
Topics	12-3
12.1. Managing deployment	12-5
Managing deployment	12-6
Deploy a solution	12-7

Local deployment	12-8
Remote deployment	12-9
Using connection properties files	12-10
Deploying to multiple servers	12-11
Deploying a solution across a grid	12-12
Deployed solutions files	12-13
Undeploying (1 of 2)	12-14
Undeploying (2 of 2)	12-15
Deleting solution files	12-16
12.2. Managing connectivity	12-17
Managing connectivity	12-18
Connectivity integration	12-19
Connectivity architecture	12-20
Connectivity architecture: Inbound	12-21
Connectivity architecture: Outbound	12-22
Connectivity definitions	12-23
Providers for JMS	12-25
Connectivity deployment steps	12-26
Configuration and deployment	12-27
Creating a solution connectivity configuration	12-28
Editing the connectivity configuration	12-29
Deploying a connectivity configuration	12-30
Troubleshooting	12-31
Common issues and resources	12-32
Unit summary	12-33
Checkpoint questions	12-34
Checkpoint answers	12-35
Exercise 14	12-36
Exercise objectives	12-37
Exercise overview: Deploying a solution and connectivity	12-38
Exercise overview: Testing connectivity	12-39
Unit 13. Administering Decision Server Insights	13-1
Unit objectives	13-2
Topics	13-3
13.1. Administration tools	13-5
Administration tools	13-6
Liberty profile monitoring	13-7
Server administration scripts	13-8
Managing servers	13-9
Managing server properties	13-10
13.2. Logging	13-11
Logging	13-12
Log and trace files	13-13
Server log settings	13-14
Server log settings	13-16
Changing log and trace settings	13-17
Finding troubleshooting information in log and trace files	13-18
Server administration properties	13-19
13.3. Monitoring WebSphere eXtreme Scale	13-21
Monitoring WebSphere eXtreme Scale	13-22

WebSphere eXtreme Scale: xscmd	13-23
Monitoring the status of quorum (1 of 2)	13-24
Monitoring the status of quorum (2 of 2)	13-25
Monitoring that all catalogs are online	13-26
13.4. Monitoring WebSphere MQ	13-27
Monitoring WebSphere MQ	13-28
Tracing the WebSphere MQ resource adapter (1 of 2)	13-29
Tracing the WebSphere MQ resource adapter (1 of 2)	13-30
Unit summary	13-31
Checkpoint questions	13-32
Checkpoint answers	13-33
Exercise 15	13-34
Exercise objectives	13-35
General administration utilities	13-36
Unit 14. Course summary	14-1
Unit objectives	14-2
Course learning objectives	14-3
Course learning objectives	14-4
To learn more on the subject (1 of 2)	14-5
To learn more on the subject (2 of 2)	14-6
Unit summary	14-7
Appendix A. List of abbreviations	A-1
Appendix B. Appendix: IBM Operational Decision Manager on Cloud	B-1
Appendix C. Resource guide	C-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®	DB2®	developerWorks®
Express®	IBM SmartCloud®	ILOG®
Insight™	Insights™	Orchestrate®
PartnerWorld®	PureApplication®	Redbooks®
SPSS®	Tivoli®	WebSphere®
z/OS®		

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

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

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

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

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

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

SoftLayer® is a trademark or registered trademark of SoftLayer, Inc., an IBM Company.

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

Course description

Administering IBM Operational Decision Manager V8.7.1

Duration: 4 days

Purpose

This course introduces administrators to IBM Operational Decision Manager (ODM) V8.7.1. You learn the concepts and skills that are necessary to install, configure, and manage Decision Server Rules, Decision Center, and Decision Server Insights.

Audience

This course is designed for administrators who install, configure, and administer Operational Decision Manager.

Prerequisites

Before taking this course, you should have:

- Basic knowledge of database management
- Familiarity with source code control (SCC) mechanisms
- Basic knowledge of Ant and the Java programming language
- Knowledge of WebSphere Application Server
- Knowledge of Java Platform, Enterprise Edition (Java EE)
- Familiarity with the Representational State Transfer (REST) architectural style
- Familiarity with the WebSphere eXtreme Scale

Objectives

After completing this course, you should be able to:

- Describe the IBM Operational Decision Manager Advanced technical architecture
- Explain topology options for a clustered environment
- Complete postinstallation configuration of Decision Server Rules, Decision Center, and Decision Server Insights
- Manage security for ODM Advanced components
- Synchronize projects across business and development environments

- Manage deployment for Decision Server Rules and Decision Server Insights
- Monitor rule execution and troubleshoot performance issues
- Manage versions, baselines, and multiple releases in Decision Center
- Install and configure a Decision Server Insights reference topology
- Configure inbound and outbound servers to manage connectivity

Agenda

Day 1

- Course introduction
- Unit 1. Introducing Operational Decision Manager for administrators
- Exercise 1. Exploring the Operational Decision Manager installation
- Unit 2. Configuring Operational Decision Manager
- Exercise 2. Configuring Rule Execution Server on WebSphere Application Server
- Exercise 3. Configuring Decision Center on WebSphere Application Server
- Exercise 4. Using a profile template to configure Rule Execution Server
- Unit 3. Managing user roles and permissions

Day 2

- Exercise 5. Customizing user access and enforcing security in Decision Center
- Unit 4. Synchronizing across environments
- Exercise 6. Synchronizing across business and development environments
- Unit 5. Managing deployment
- Exercise 7. Managing deployment
- Unit 6. Administering Rule Execution Server
- Exercise 8. Exploring the Rule Execution Server console

Day 3

- Unit 7. Using Decision Warehouse to audit ruleset execution
- Exercise 9. Auditing ruleset execution through Decision Warehouse
- Unit 8. Monitoring execution and performance
- Exercise 10. Monitoring rule execution and performance
- Unit 9. Managing baselines and multiple releases
- Exercise 11. Managing baselines and multiple releases

Day 4

- Unit 10. Introducing Decision Server Insights for administrators
- Exercise 12. Installing Decision Server Insights
- Unit 11. Configuring Insight Server
- Exercise 13. Configuring Decision Server Insights
- Unit 12. Managing deployment for Decision Server Insights
- Exercise 14. Managing deployment and connectivity
- Unit 13. Administering Decision Server Insights
- Exercise 15. Administering Decision Server Insights
- Unit 14. Course summary

Unit 1. Introducing Operational Decision Manager for administrators

What this unit is about

This unit describes the Operational Decision Manager architecture, installation, and roles.

What you should be able to do

After completing this unit, you should be able to:

- Describe the architecture of Operational Decision Manager
- Describe Operational Decision Manager installation
- Describe the Operational Decision Manager roles

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Describe the architecture of Operational Decision Manager
- Describe Operational Decision Manager installation
- Describe the Operational Decision Manager roles

© Copyright IBM Corporation 2015

Figure 1-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Operational Decision Manager offerings and components
- Interaction between components
- Installation overview
- IBM ODM on Cloud

© Copyright IBM Corporation 2015

Figure 1-2. Topics

WB393 / ZB3931.0

Notes:

1.1. Operational Decision Manager offerings and components

Operational Decision Manager offerings and components



© Copyright IBM Corporation 2015

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

10.1

Figure 1-3. Operational Decision Manager offerings and components

WB393 / ZB3931.0

Notes:

Operational Decision Manager editions

Advanced

- Decision Server Rules
- Designer Server Events
- Decision Server Insights
- Decision Center

Standard

- Decision Server Rules
- Decision Center

Express

- Decision Server Rules

© Copyright IBM Corporation 2015

Figure 1-4. Operational Decision Manager editions

WB393 / ZB3931.0

Notes:

Operational Decision Manager (ODM) has three editions to meet client needs:

- **Operational Decision Manager Standard**

This edition of Operational Decision Manager includes business rules only. You can use the complete Business Rules Management System (BRMS) from IBM to develop business rule applications.

- **Operational Decision Manager Express**

This edition offers the same functionality as Standard, but with licensing restrictions that make it an affordable entry point into rules-based decision management capabilities.

This edition remains the low-cost-of-ownership BRMS solution of IBM. You can get started with small- to medium-sized applications. ODM Express entitles you to use only the business rules capabilities, but not the events capabilities, within a limited configuration. In particular, you cannot use ODM Express in a cluster environment.

- **Operational Decision Manager Advanced**

This edition is the newest addition to the Operational Decision Management portfolio. ODM Advanced allows organizations to take advantage of the data available to them to enrich and improve their business decisions. With real-time, actionable insight capabilities, companies can now bring together data from multiple sources to identify meaningful patterns and trends that can be applied to operational decisions. As a result, an organization can create and shape business moments by automating decisions in context.

This edition provides a full-featured decision management platform that includes Decision Server Advanced. You can address the entire decision automation scope, which includes business rules, business events, and decision insights capabilities.

For more information about the IBM Operational Decision Manager packaging, see the product documentation.



Operational Decision Manager cloud-based offerings

ODM Pattern

- Latest level of ODM rules and events and packaged as part of ODM Advanced

ODM on Cloud

- A subscription-based operational decision management cloud service for development, test, and production
- Available exclusively on IBM Cloud infrastructure and managed by IBM

© Copyright IBM Corporation 2015

Figure 1-5. Operational Decision Manager cloud-based offerings

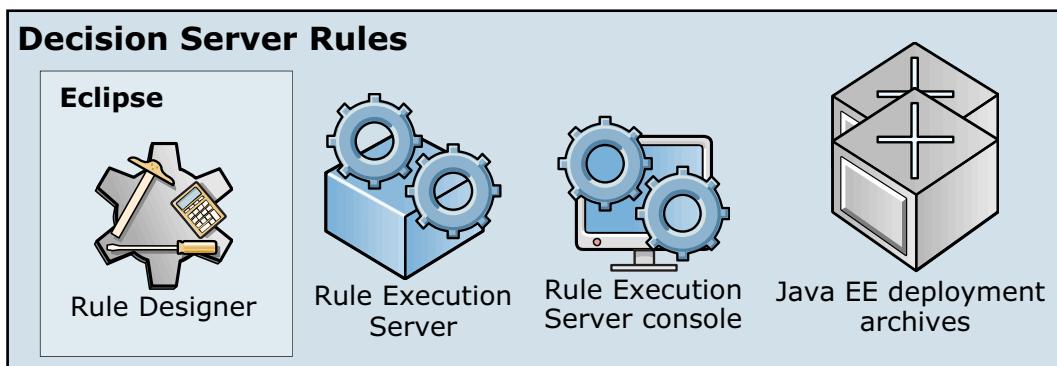
WB393 / ZB3931.0

Notes:

You learn more about ODM Pattern in the next unit. ODM on Cloud is described later in this unit.

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

Decision Server Rules



© Copyright IBM Corporation 2015

Figure 1-6. Decision Server Rules

WB393 / ZB3931.0

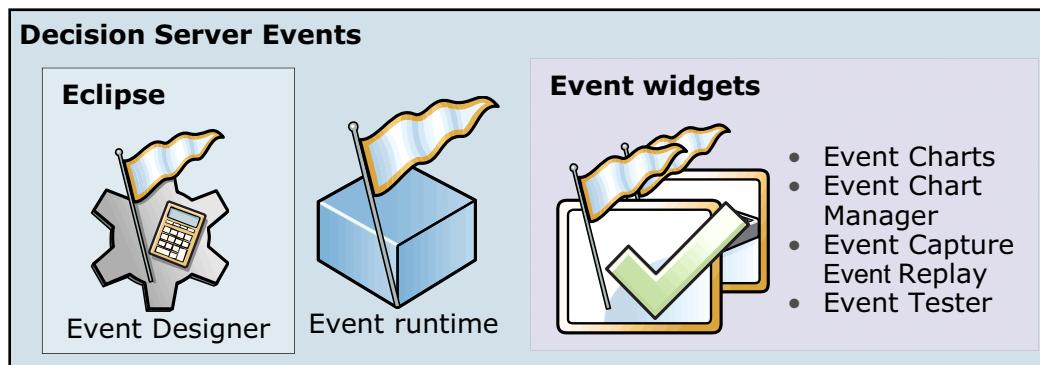
Notes:

Decision Server Rules provides the runtime and development components to automate decision logic. You can use these components to automate and govern frequently occurring, repeatable decisions that control the actions of critical business systems. Depending on your role: an architect, a developer, a QA tester, a business user, or a policy manager you are interested in different aspects of developing a business rule application and integrating it with the calling business application. It is important to understand your role and the role of others in the development of your application.

The Decision Server Rules installation includes:

- Rule Designer: Business rule application development tool for developers, modelers, and architects
- Rule Execution Server: Managed business rule execution platform that embeds the rule engine
- Rule Execution Server console
- Deployment archives for Java EE

Decision Server Events



© Copyright IBM Corporation 2015

Figure 1-7. Decision Server Events

WB393 / ZB3931.0

Notes:

In the context of business event processing (BEP), Decision Server Events serves as the event management layer in the IT infrastructure.

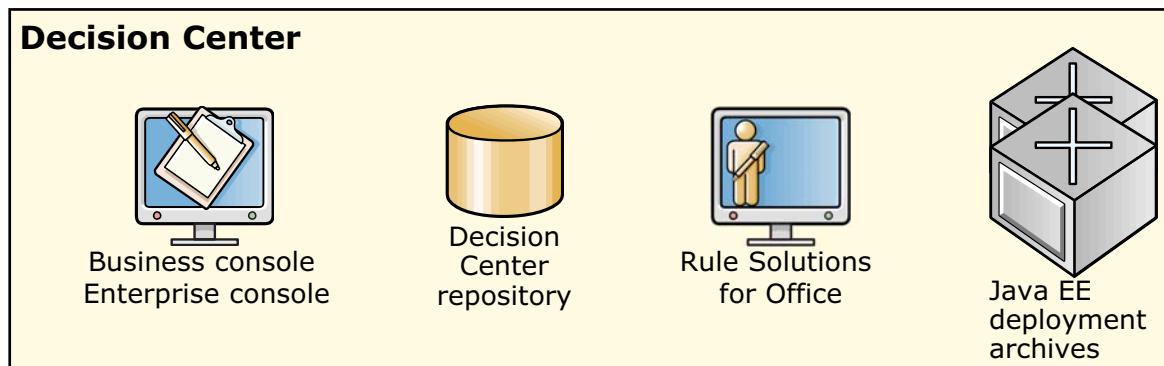
Decision Server Events is designed specifically for managing the business events that flow across systems and people with the goal of providing timely insight and response. It helps businesses detect, evaluate, and respond to events that affect their operations. It discovers event patterns and initiates actions, which are based on defined data models and business logic.

The installation of Decision Server Events includes:

- Event Designer: Used to design, develop, test, deploy, maintain, and monitor event applications.
- Event runtime: Manages the real-time business event coordination that was defined during application development. The event runtime can be deployed and configured on WebSphere Application Server.
- Event widgets: Used to capture events, replay a sequence, and run tests.

Decision Server Events is not covered in this course.

Decision Center



© Copyright IBM Corporation 2015

Figure 1-8. Decision Center

WB393 / ZB3931.0

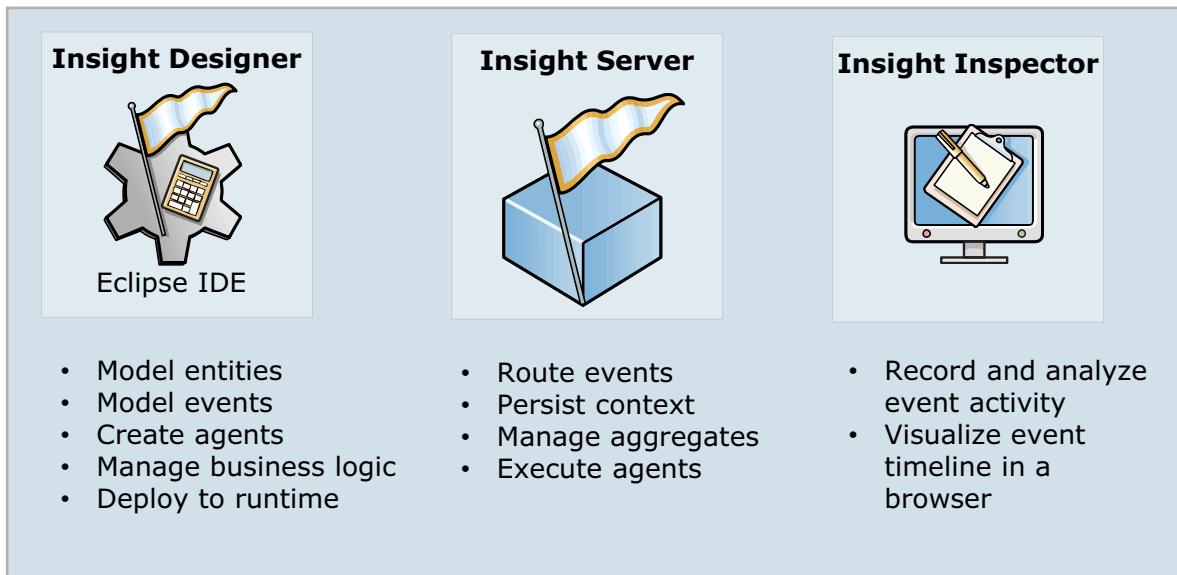
Notes:

Decision Center is the central hub that coordinates the decision lifecycle across business and IT worlds. Decision Center provides various environments and tools for business users to manage their decisions.

Decision Center installation includes these components:

- A set of business rule management tools for policy managers and business users, including:
 - Decision Center Business console
 - Decision Center Enterprise console
 - Decision Center repository
- Rule Solutions for Office add-ins for Microsoft Word and Microsoft Excel
- Deployment archives for Java EE

Decision Server Insights



© Copyright IBM Corporation 2015

Figure 1-9. Decision Server Insights

WB393 / ZB3931.0

Notes:

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

It includes:

- Insights 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

- Eclipse interface to develop rule-based, event-driven solutions
- 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

Insight Server Runtime

- 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 timeline of event activity for entities

1.2. Interaction between components

Interaction between components



© Copyright IBM Corporation 2015

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

10.1

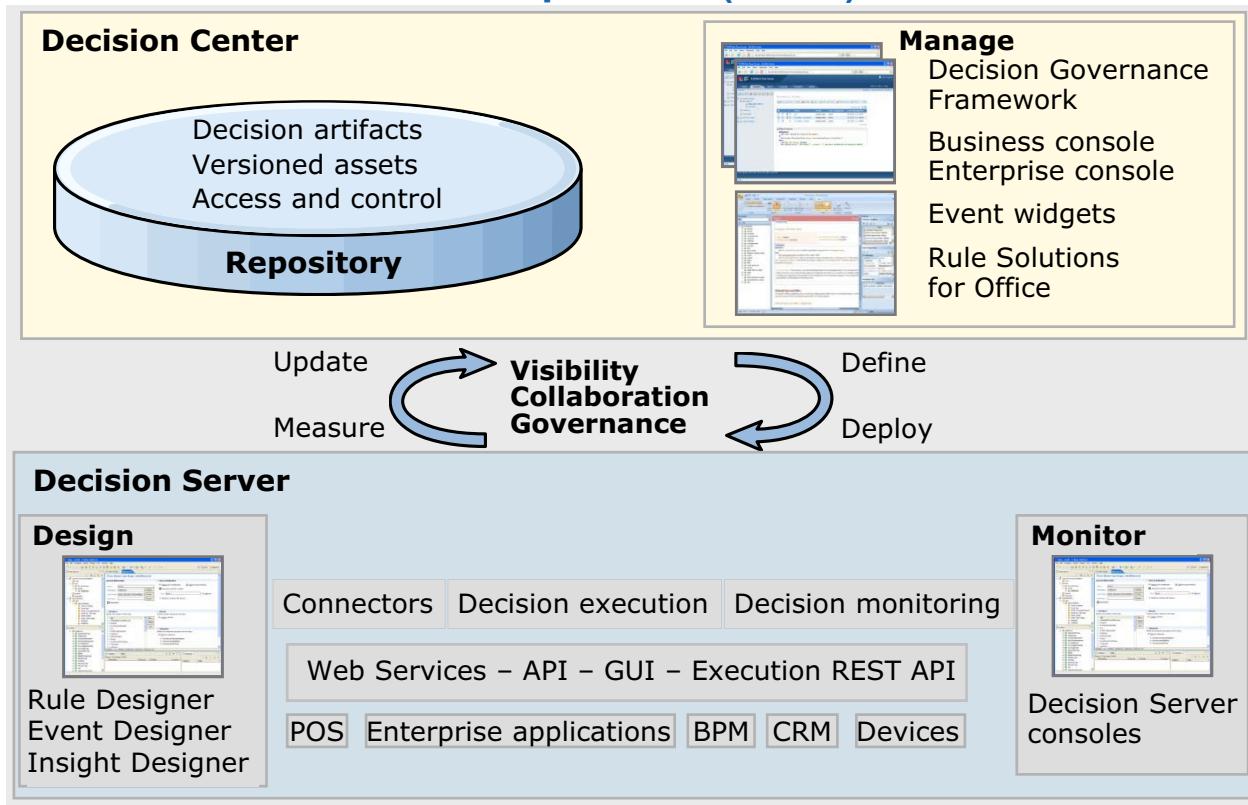
Figure 1-10. Interaction between components

WB393 / ZB3931.0

Notes:



Interaction between components (1 of 2)



© Copyright IBM Corporation 2015

Figure 1-11. Interaction between components (1 of 2)

WB393 / ZB3931.0

Notes:

Operational Decision Manager consists of a set of modules that operate in different environments, but also work together to provide a comprehensive platform for the management and execution of business rules and business events. Throughout the lifecycle of a business rule application, users who have different roles work together to perform various tasks. User roles are generally divided between technical and business user tasks. The user roles for Operational Decision Manager are covered later in this section.

Decision Center provides all the capabilities for business users to participate directly in the definition and governance of business rule and business event-based decision logic. Through the capabilities of WebSphere Decision Center, the entire organization is aligned in the implementation of automated decisions, which accelerates the maintenance lifecycle as decisions evolve based on new external and internal requirements.

- Decision artifacts are stored in a centralized repository with version control, release management, and secure access.

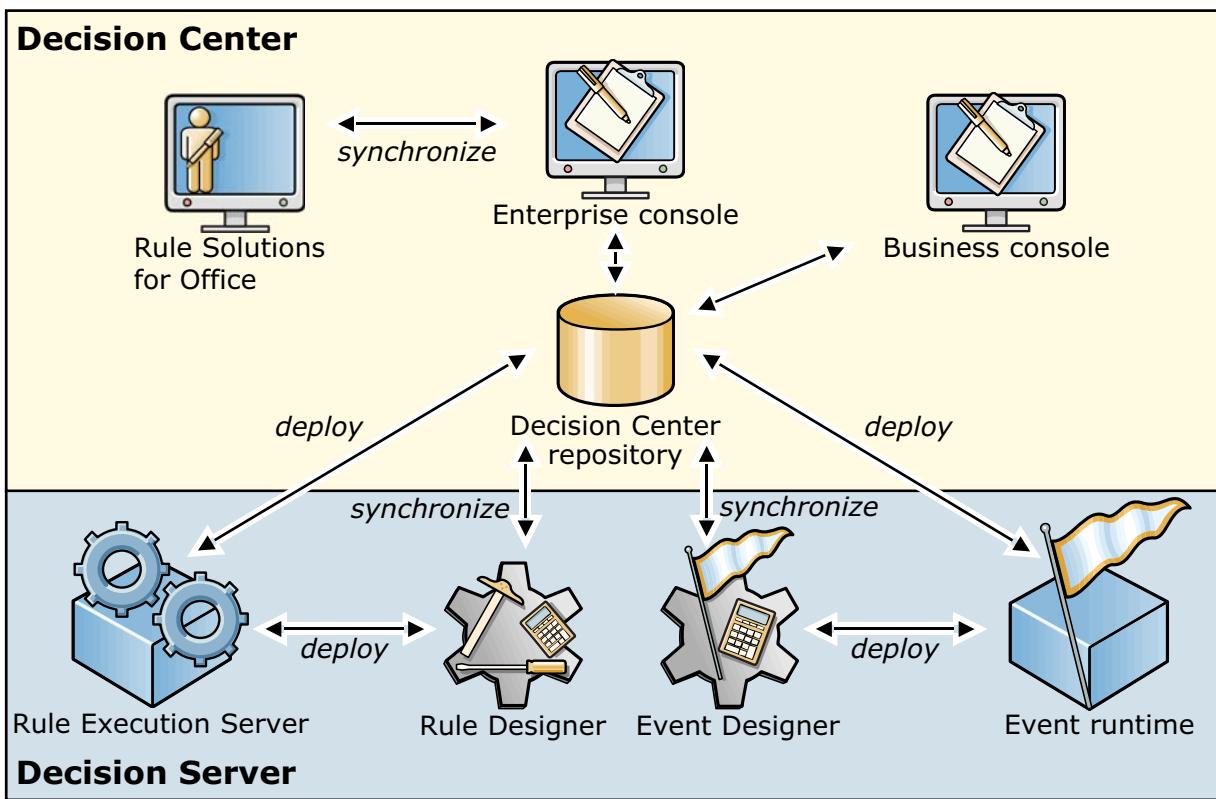
Decision Center also comes with various user interfaces to collaborate, share, and manage their business policies in a secured environment:

- Business console, which provides a social collaboration environment for rule authoring and change management
- Enterprise console, which is a web console that provides a full set of authoring and management capabilities, including testing and simulations
- Rule Solutions for Office, which provides offline authoring and review features in Microsoft Office documents

Decision Server for technical users contains all runtime components and Eclipse-based development tools.

- Rule Designer and Event Designer are integrated as Eclipse plug-ins, where developers can design, develop, and synchronize with the business environment.
- Decision Server is also the execution environment for business rules and events, and provides execution management and monitoring. Decision Server can detect event-based patterns and process this information against hundreds or even thousands of business rules to determine how to respond.

Interaction between components (2 of 2)



© Copyright IBM Corporation 2015

Figure 1-12. Interaction between components (2 of 2)

WB393 / ZB3931.0

Notes:

This figure shows an overview of how the different modules work together through synchronization and deployment.

Business rule application development starts in Designer, which is an Eclipse-based environment for technical users. Based on the business requirements, developers work in Rule Designer or Event Designer to design the project and set up the authoring environment for business users.

The synchronization mechanism in Designer allows developers to publish rules to Decision Center, where business users can author and maintain the rules in a web-based application.

Decision Center includes permissions and access control, so multiple users can work on the same project. Business users who prefer to work offline can author or review rules in Rule Solutions for Office, which is a plug-in to Microsoft Office Word and Excel. A synchronization mechanism in Decision Center allows rules to be shared and synchronized between business users who are working in Decision Center Console and users who are working in Rule Solutions for Office.

To make the rules available for execution, the rules are deployed to the Rule Execution Server, which is a managed execution environment that houses the rule engine. Rules can be deployed to Rule Execution Server from either Designer or Decision Center.

Operational Decision Manager user roles

Business users		
	Business analyst	Bridges between the business side and technical side of a business rule application
	Policy manager	Business expert and owner of business policy
	Rule author	Business domain expert who updates and reviews rules
Technical users		
	Architect	Manages overall deployment, organization of rules, and optimization of rule execution
	Developer	Develops, tests, and deploys business rule applications and event applications
	Administrator	Installs and configures rule management and execution environments

© Copyright IBM Corporation 2015

Figure 1-13. Operational Decision Manager user roles

WB393 / ZB3931.0

Notes:

During the development of a business rule application, various skills are required at different stages of the development lifecycle. In Operational Decision Manager, these skills are grouped into a set of *business* and *technical* user roles.

Business users include policy managers, rule authors, and business analysts.

Technical users include architects, developers, and administrators.



Interaction between roles

- Roles do not correspond to individuals, but to activities and responsibilities
- Tasks might not correspond to a single position in your organization
 - A business expert might be involved in the technical side of things
 - A developer might also be the person who authors and manages the rules
- Communication between the business and technical roles is vital

© Copyright IBM Corporation 2015

Figure 1-14. Interaction between roles

WB393 / ZB3931.0

Notes:

Roles refer to tasks and responsibilities rather than individuals.

Roles might not correspond to a single position in your organization, so crossover between departments might make it difficult to discern who fits into a particular role. For example, a business policy expert might be involved in the technical side of things, and a developer might also be the person who writes and manages the rules.

Particularly during the early stages of developing an application, organizations should expect extensive interaction between business roles and technical roles.

Administrator tasks

- Responsibilities:
 - Deploying and configuring the server and database for Decision Center and Rule Execution Server
 - Managing user access to Decision Center and Rule Execution Server
 - Configuring trace data sources for testing purposes
 - Deploying applications
 - Redeploying rulesets and event assets as changes are made
 - Generating detailed execution reports
 - Tracking and monitoring rule execution
 - Restoring a particular application state
- Tools: Servers for Decision Center or runtime environments



Administrator

© Copyright IBM Corporation 2015

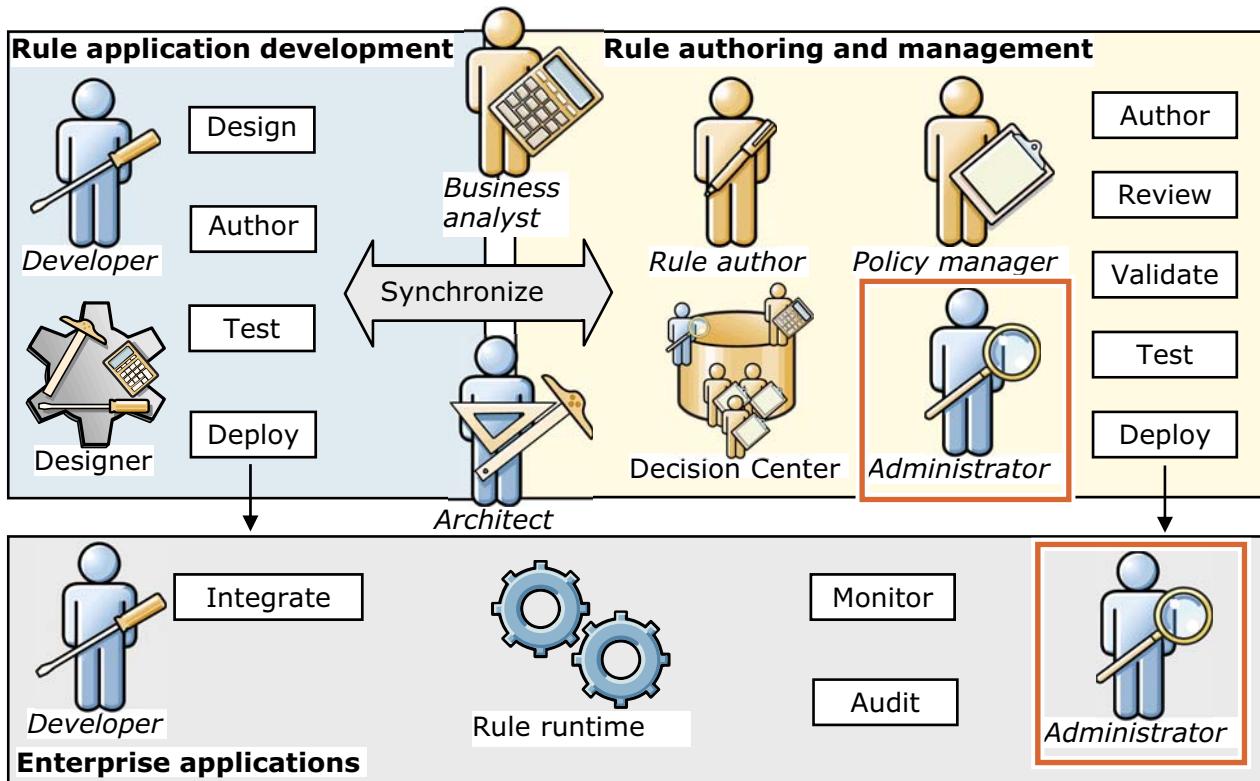
Figure 1-15. Administrator tasks

WB393 / ZB3931.0

Notes:

The administrator role involves various tasks throughout the lifecycle of a decision management project.

Roles and activities



© Copyright IBM Corporation 2015

Figure 1-16. Roles and activities

WB393 / ZB3931.0

Notes:

Here you see an overview of the various roles and activities that are involved in a decision management project.

As an administrator, you install and configure the servers and databases that are used for the Decision Server and Decision Center components. In particular, you are involved in postinstallation configuration of Rule Execution Server and Decision Center. You learn more about these tasks in the next unit, Unit 2, "Configuring Operational Decision Manager".

After the installation and configuration are complete, your role involves monitoring and management of these environments and ongoing collaboration with both business and technical teams. You also ensure that the various stakeholders have appropriate access to the product modules. You learn more about roles and permissions in Unit 3, "Managing user roles and permissions".

1.3. Installation overview

Installation overview



© Copyright IBM Corporation 2015

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

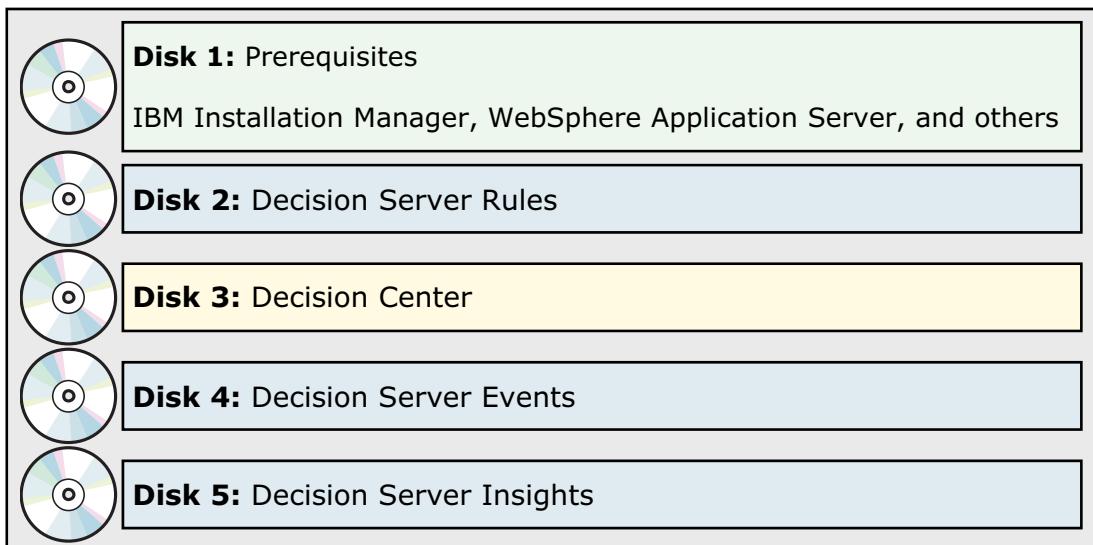
10.1

Figure 1-17. Installation overview

WB393 / ZB3931.0

Notes:

Operational Decision Manager installation files (1 of 2)



- ODM installation files are packaged as five separate disks
 - Disks 1-3 are used to install IBM Installation Manager, ODM Standard, ODM Express, and prerequisite software
 - Disks 4 and 5 are used to install ODM Advanced components

© Copyright IBM Corporation 2015

Figure 1-18. Operational Decision Manager installation files (1 of 2)

WB393 / ZB3931.0

Notes:

The Operational Decision Manager installation repositories are provided as downloadable compressed files, or on physical DVDs. You must copy or download the installation repositories onto your workstation.

If you download the repositories, which are packaged as `.tar` files, you must extract them with an archiving utility of your choice. As a result of the file extraction and depending on your licenses, you obtain all or a combination of folders that are named **disk1**, **disk2**, **disk3**, **disk4**, and **disk5**.



Operational Decision Manager installation files (2 of 2)

- In a typical launchpad installation, the launchpad uses disks 1-3 to install Decision Server Rules and Decision Center
- You can also use the Installation Manager to install Decision Server Rules and Decision Center
 - Connect the installation repositories on disks 1-3 in the Installation Manager preferences
- ODM Advanced: Use the Installation Manager to install Decision Server Events or Decision Server Insights
 - Connect the installation repositories for Decision Server Events or Decision Server Insights in the Installation Manager preferences

© Copyright IBM Corporation 2015

Figure 1-19. Operational Decision Manager installation files (2 of 2)

WB393 / ZB3931.0

Notes:

Before you install through Installation Manager, you must add the installation repositories to the Installation Manager preferences.

1. Start Installation Manager from the **Windows Start** menu or from a command line.
2. Open **File > Preferences** and specify the path to the installation repository location.

After you point to the installation directories, you can start the Installation Manager installation and select your installation repositories.



What you can install: Launchpad or Installation Manager

Launchpad	Installation Manager
<ul style="list-style-type: none"> • Typical installation <ul style="list-style-type: none"> – Prerequisite software – Component features – Default environments • Can install: <ul style="list-style-type: none"> – Decision Server Rules – Decision Center 	<ul style="list-style-type: none"> • Select components to install <ul style="list-style-type: none"> – Can also update, modify, roll back, or uninstall product packages • Can install: <ul style="list-style-type: none"> – Decision Server Rules – Decision Center – Business Rules Embedded – Decision Server Events (ODM Advanced) – Decision Server Insights (ODM Advanced)

© Copyright IBM Corporation 2015

Figure 1-20. What you can install: Launchpad or Installation Manager

WB393 / ZB3931.0

Notes:

Launchpad

You can use the product launchpad to install Operational Decision Manager along with prerequisite software.

In a typical installation, the launchpad installs the prerequisites for Operational Decision Manager. It then installs the components for Decision Center and Decision Server Rules. The installer also configures these components on a WebSphere Application Server profile.

The **Sample Server** installation creates a server profile on WebSphere Application Server to run the samples and tutorials. After you install, the sample server does not require any configuration. The profile is created the first time that you start the sample server.

Installation Manager

With the Installation Manager, you can choose which components to install. You must have an existing Installation Manager and the required software to install through the Installation Manager. You must also add the installation repositories to the Installation Manager preferences before you can install software through the Installation Manager.

With the Installation Manager, you can customize your installation for specific configurations, for example:

- Connect to other installation repositories
- Extend an existing Eclipse environment
- Install Operational Decision Manager in another compatible IBM product
- Get the archives for configuring on an application server other than WebSphere Application Server



Troubleshooting the installation

- Check the installation logs
- For a list of known issues, see the product documentation:
<http://www.ibm.com/support/docview.wss?uid=swg21683393>

© Copyright IBM Corporation 2015

Figure 1-21. Troubleshooting the installation

WB393 / ZB3931.0

Notes:

If you have problems when installing the product through Installation Manager, you can retrieve the installation logs to see details about the issues that you encountered.

Installation Manager installation logs

The IBM Installation Manager *installation logs* contain information on potential installation issues that you encounter. You can find the installation logs in the following locations.

- Windows 2000, XP, and 2003:
C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager\logs
- Windows Vista, Windows 7, and 2008:
C:\ProgramData\IBM\Installation Manager\logs
- Linux and UNIX:
/var/ibm/InstallationManager/logs

IBM Installation Manager .log file

The .log file is for IBM Installation Manager-specific issues that result from internal problems, for example, any exceptions that are issued. You can find the .log file in the following locations:

- Microsoft Windows 2000, XP, and 2003:

-C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager\pluginState\.metadata\.log

- Microsoft Windows Vista, 7, and 2008:

-C:\ProgramData\IBM\Installation Manager\pluginState\.metadata\.log

- Linux and UNIX

-/var/ibm/InstallationManager/pluginState\.metadata\.log



Note

Since the IBM Installation Manager is an Eclipse-based tool, it has a workspace. As shown in the directory paths for the .log file, the value *pluginState* is the name of the IBM Installation Manager workspace.



Information

You can use IBM Support Assistant to collect information about IBM Installation Manager to send to IBM Support.

For more information, see the following technote:

<http://www.ibm.com/support/docview.wss?uid=swg21288289>

1.4. IBM ODM on Cloud



IBM ODM on Cloud



© Copyright IBM Corporation 2015

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

10.1

Figure 1-22. IBM ODM on Cloud

WB393 / ZB3931.0

Notes:

This topic provides a brief introduction to IBM ODM on Cloud. For more information, see Appendix B, "Appendix: IBM Operational Decision Manager on Cloud".

Operational Decision Manager deployment targets

On-premises	Private cloud and IaaS cloud	PaaS cloud
<ul style="list-style-type: none">• IBM Operational Decision Manager	<ul style="list-style-type: none">• PureApplication System• PureApplication System on SoftLayer• SoftLayer IaaS	<ul style="list-style-type: none">• Business Rules for Bluemix

© Copyright IBM Corporation 2015

Figure 1-23. Operational Decision Manager deployment targets

WB393 / ZB3931.0

Notes:

IBM Operational Decision Manager supports a wide range of deployment targets, including on-premises deployment, private and IaaS clouds, and PaaS clouds.

- IaaS: Infrastructure-as-a-Service
- PaaS: Platform-as-a-Service



IBM ODM on Cloud overview

- ODM cloud service for development, testing, and production
- Features:
 - Cloud-based, collaborative, and role-based environment
 - Ready-to-use development, test, and production environments are available
 - Monthly subscription plans
- Available exclusively on IBM Cloud infrastructure and managed by IBM
 - Allows you to focus on application development, integration, and support
 - IBM manages maintenance, updates, backup, and other considerations
 - Capacity is provided in any IBM SoftLayer data center
- Artifacts that are created with IBM ODM on Cloud are compatible with IBM ODM on-premises product

© Copyright IBM Corporation 2015

Figure 1-24. IBM ODM on Cloud overview

WB393 / ZB3931.0

Notes:

IBM ODM on Cloud: Three runtime environments

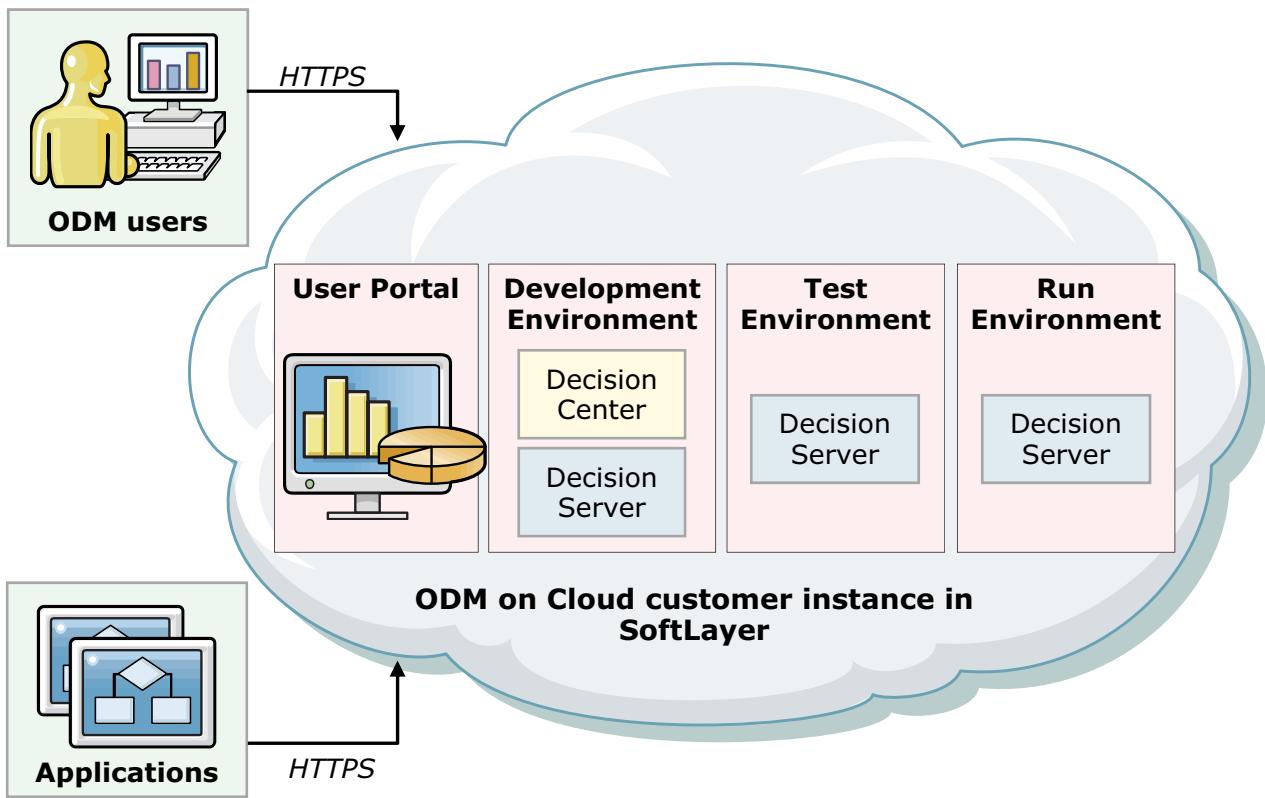


Figure 1-25. IBM ODM on Cloud: Three runtime environments

WB393 / ZB3931.0

Notes:

IBM ODM on Cloud provides three runtime environments for decision management: development, test, and run.

In this diagram:

- ODM users include developers, business analysts, business users, and rule authors who access Rule Designer, Decision Center, and the various user consoles.
- Applications are applications that call deployed decision services.



Unit summary

Having completed this unit, you should be able to:

- Describe the architecture of Operational Decision Manager
- Describe Operational Decision Manager installation
- Describe the Operational Decision Manager roles

© Copyright IBM Corporation 2015

Figure 1-26. Unit summary

WB393 / ZB3931.0

Notes:

Checkpoint questions

- 1. True or False:** You can use the launchpad to install Decision Server Insights.
- 2. True or False:** Decision Server Rules includes Rule Designer, Rule Execution Server, Rule Execution Server console, and Rule Execution Server application server archives.
- 3. True or False:** Decision Center includes only the following components: Decision Center Business console, Decision Center Enterprise console, Rule Solutions for Office add-ins, and Decision Center application server archives.
- 4. True or False:** ODM Standard installations include Decision Server Events and Decision Server Insights.

© Copyright IBM Corporation 2015

Figure 1-27. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

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



Checkpoint answers

- 1. False.** You can use the launchpad to install Operational Decision Manager Express or Standard editions. Decision Server Insights is part of the Advanced edition and is installed with IBM Installation Manager.
- 2. True.**
- 3. True.**
- 4. False.** Decision Server Events and Decision Server Insights are included with ODM Advanced, but not ODM Standard or ODM Express.

© Copyright IBM Corporation 2015

Figure 1-28. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 1



Exploring the Operational Decision Manager installation

© Copyright IBM Corporation 2015

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

10.1

Figure 1-29. Exercise 1

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Describe the Operational Decision Manager installation
- Modify installed software packages

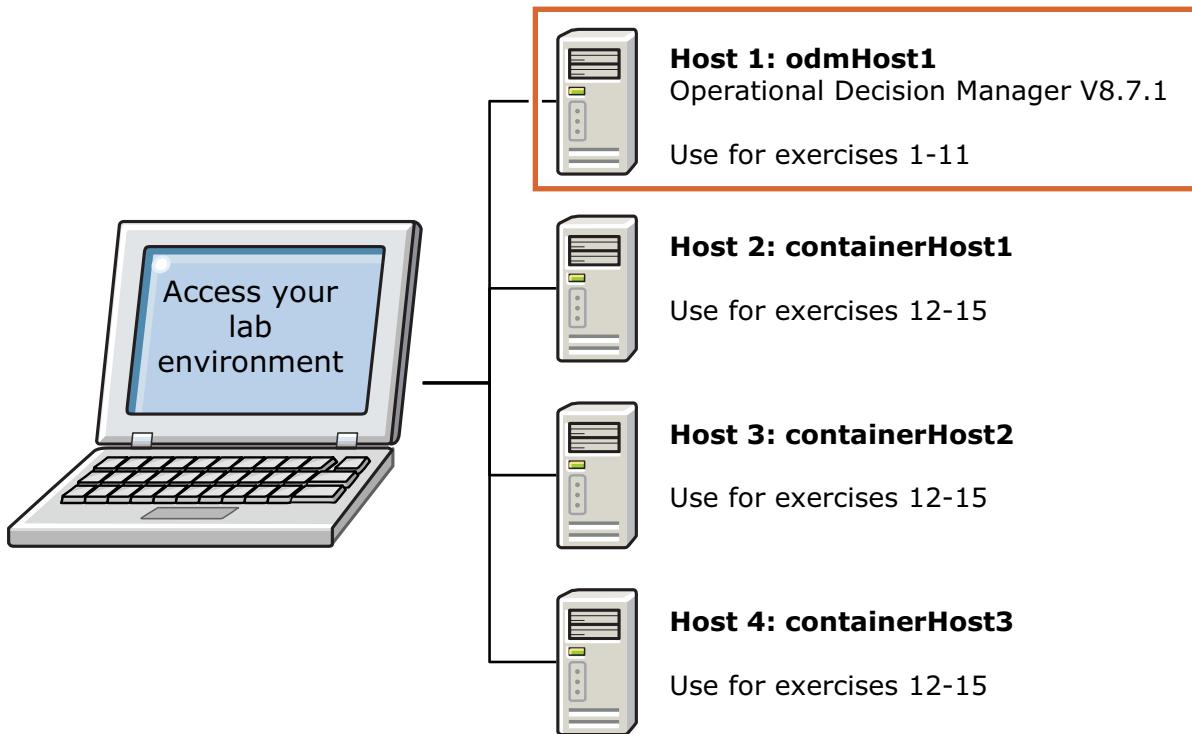
© Copyright IBM Corporation 2015

Figure 1-30. Exercise objectives

WB393 / ZB3931.0

Notes:

Lab environment



© Copyright IBM Corporation 2015

Figure 1-31. Lab environment

WB393 / ZB3931.0

Notes:

Your lab environment includes four VMware images. For the first exercises, you work entirely on Host 1.



By default, Host 1 is named: odmHost1.

A different host name might be assigned to your main host.

The main host, odmHost1, has ODM Advanced installed, but without the Decision Server Insights component. You continue on this host for the first 11 exercises.

During Exercise 12, you install Decision Server Insights on all four hosts, and you work with Decision Server Insights for the remainder of the course.

Unit 2. Configuring Operational Decision Manager

What this unit is about

In this unit, you learn how to configure Rule Execution Server and Decision Center for production. You also learn about profile templates and clustering.

What you should be able to do

After completing this unit, you should be able to:

- Configure Rule Execution Server and Decision Center
- Explain how to deploy Decision Server and Decision Center for high availability and scalability
- Describe the IBM ODM Application Pattern

How you will check your progress

- Checkpoint
- Exercises



Unit objectives

After completing this unit, you should be able to:

- Configure Rule Execution Server and Decision Center
- Explain how to deploy Decision Server and Decision Center for high availability and scalability
- Describe the IBM ODM Application Pattern

© Copyright IBM Corporation 2015

Figure 2-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Configuring ODM servers
- Configuring Rule Execution Server
- Configuring Decision Center
- Completing the installation
- Profile templates for WebSphere Application Server
- Configuring Rule Execution Server for production
- Configuring ODM servers for high availability
- IBM ODM Pattern V8.7.1

© Copyright IBM Corporation 2015

Figure 2-2. Topics

WB393 / ZB3931.0

Notes:

2.1. Configuring ODM servers

Configuring ODM servers



© Copyright IBM Corporation 2015

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

10.1

Figure 2-3. Configuring ODM servers

WB393 / ZB3931.0

Notes:

Configuring Operational Decision Manager Java EE and Java SE servers

- ODM servers can be configured on:
 - Java SE platform
 - WebSphere Application Server
 - Tomcat
 - JBoss
 - WebLogic Server
- For distributed platforms, application server-specific archives are provided for you to deploy and configure
 - When you use the WebSphere Application Server profile templates, the scripts handle the deployment of these archives

© Copyright IBM Corporation 2015

Figure 2-4. Configuring Operational Decision Manager Java EE and Java SE servers

WB393 / ZB3931.0

Notes:

2.2. Configuring Rule Execution Server

Configuring Rule Execution Server



© Copyright IBM Corporation 2015

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

10.1

Figure 2-5. Configuring Rule Execution Server

WB393 / ZB3931.0

Notes:



Configuring Rule Execution Server

- Using Rule Execution Server on an instance of an application server requires deployment and configuration of the provided archives
- Rule Execution Server is supported on:
 - Java SE platform
 - WebSphere Application Server
 - Tomcat
 - JBoss
 - WebLogic Server

© Copyright IBM Corporation 2015

Figure 2-6. Configuring Rule Execution Server

WB393 / ZB3931.0

Notes:

Rule Execution Server provides a reliable and scalable execution environment for your business rule application. You can configure Rule Execution Server for various environments.

Configuring Rule Execution Server on WebSphere Application Server (1 of 4)

- You must deploy the provided archives and perform a number of configuration steps
- Some configuration steps are optional or dependent on the persistence type for your environment
- You can use profile templates to automate deployment and configuration
 - Use profile templates to configure a cluster
- After deployment and configuration, you must complete the installation by running the Installation Settings wizard in Rule Execution Server console
 - Sign in to the console as the administrator
 - You can use the console to create resources in an empty database schema and run SQL drop statements that clear an existing Rule Execution Server database

© Copyright IBM Corporation 2015

Figure 2-7. Configuring Rule Execution Server on WebSphere Application Server (1 of 4)

WB393 / ZB3931.0

Notes:

Configuring Rule Execution Server on WebSphere Application Server (2 of 4)

Step 1: Select and apply the persistence type	You can change the default data source RuleApp and Java XOM persistence settings by running an Ant script that generates a new Rule Execution Server management archive. The Ant script is provided in the product distribution: <InstallDir>/ODM/executionserver/bin/ressetup.xml
Step 2: Enforce the database user permissions	If Rule Execution Server data is stored in a database, you can restrict the type of operations that a user can perform on a database by defining access privileges. Database privilege types differ across the supported databases.
Step 3: Create an empty database schema	If you set persistence to <code>file</code> (in Step 1), you skip all the database-related tasks and proceed to Step 6 . If persistence is set to <code>datasource</code> (default) or <code>jdbc</code> , you must create an empty dedicated database schema and then populate the database. Use SQL scripts to create a dedicated schema in the database.
Step 4: Set up a data source and connection pool	You must create a JDBC provider (<code>datasource</code>) and connection pool for WebSphere Application Server.

© Copyright IBM Corporation 2015

Figure 2-8. Configuring Rule Execution Server on WebSphere Application Server (2 of 4)

WB393 / ZB3931.0

Notes:

The steps that are outlined here can be done manually or you can use the profile template to automate them. During the exercise, you walk through the manual configuration to understand what is involved at each step.

Configuring Rule Execution Server on WebSphere Application Server (3 of 4)

Step 5: Activate security on WebSphere Application Server	To activate access control for Rule Execution Server in WebSphere Application Server, you must create users and groups, and map the resAdministrative groups to the Monitor role.
Step 6: Deploy the MBean descriptors	Rule Execution Server architecture is based on the Java Management Extension (JMX) API so you must deploy the MBean descriptors either globally for all Rule Execution Server instances or for a single Rule Execution Server instance.
Step 7: Deploy the XU RAR	After you deploy the MBean descriptors, you deploy a resource adapter archive (RAR) for the Execution Unit (XU) which also contains the persistence layer. Optionally, you can define more than one XU resource adapter when you have more than one node in your environment or if you want to isolate the development environment and testing environment in one single node.
Step 8: Deploy the Rule Execution Server Management EAR	After you activate the security, you deploy the Rule Execution Server EAR to WebSphere Application Server.

© Copyright IBM Corporation 2015

Figure 2-9. Configuring Rule Execution Server on WebSphere Application Server (3 of 4)

WB393 / ZB3931.0

Notes:

Configuring Rule Execution Server on WebSphere Application Server (4 of 4)

Step 9: Populate a Rule Execution Server database	After the empty database schema is created, you can populate the Rule Execution Server database by running SQL scripts. You can run the SQL scripts either from the Rule Execution Server console in the Installation Settings wizard (on Windows) or from the SQL tool of your database.
Step 10: Deploy the hosted transparent decision service EAR	Optional. You can deploy the EAR file for hosted transparent decision services.
Step 11: Define the DecodeUrlAsUTF8 custom property	Optional. After you deploy the hosted transparent decision service EAR file, you define the web container custom property DecodeUrlAsUTF8 to support localized ruleset paths.
Step 12: Verify the deployment and configuration	Optional. It is good practice to verify that Rule Execution Server was successfully deployed and configured by running the diagnostic tools.

© Copyright IBM Corporation 2015

Figure 2-10. Configuring Rule Execution Server on WebSphere Application Server (4 of 4)

WB393 / ZB3931.0

Notes:

2.3. Configuring Decision Center

Configuring Decision Center



© Copyright IBM Corporation 2015

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

10.1

Figure 2-11. Configuring Decision Center

WB393 / ZB3931.0

Notes:



Configuring Decision Center

- To use Decision Center on any instance of an application server (other than the Sample Server that you use during this course), you must deploy and configure the provided archives
- Supported application servers include:
 - WebSphere Application Server
 - Tomcat
 - JBoss
 - WebLogic Server

© Copyright IBM Corporation 2015

Figure 2-12. Configuring Decision Center

WB393 / ZB3931.0

Notes:

Decision Center is the central hub that coordinates the decision lifecycle across the business and IT environments. Decision Center provides various environments and tools for business users to manage their decisions. You can configure Decision Center on various application servers.

Configuring Decision Center on WebSphere Application Server (1 of 3)

- You must deploy the provided archives and perform a number of configuration steps
 - Some configuration steps are optional
- You can use profile templates to automate deployment and configuration
 - Use profile templates to configure a cluster
- After deployment and configuration, you must complete the installation by running the Installation Settings wizard in Rule Execution Server console
 - Sign in to the console as the administrator
 - You can use it to create resources in an empty database schema and run SQL drop statements that clear an existing Rule Execution Server database

© Copyright IBM Corporation 2015

Figure 2-13. Configuring Decision Center on WebSphere Application Server (1 of 3)

WB393 / ZB3931.0

Notes:



Configuring Decision Center on WebSphere Application Server (2 of 3)

Step 1: Enforcing database user permissions	Decision Center data is stored in a database. You restrict the type of operations that a user can run on a database by defining access privileges.
Step 2: Creating a data source and connection pool	When your database is already running, you can create a JDBC provider, a data source, and a connection pool. You can change the data source properties, if necessary. Then, you establish the connection.
Step 3: Configuring security	The application server security manages access to Decision Center. To access Decision Center in WebSphere Application Server, you must define a user registry. You can also manage the security policies.
Step 4: Deploying the Decision Center management EAR	To deploy the Decision Center EAR file for user-to-role mappings on WebSphere Application Server, you must declare custom groups, deploy the archive file, and change the class loader sequence.

© Copyright IBM Corporation 2015

Figure 2-14. Configuring Decision Center on WebSphere Application Server (2 of 3)

WB393 / ZB3931.0

Notes:

The steps that are outlined here can be done manually or you can use the profile template to automate them. During the exercise, you walk through the manual configuration to understand what is involved at each step.



Note

When you configure the Decision Center Business console, you must also configure the Solr search engine to provide search function in the console.

You do not work with the Business console during this course.

Configuring Decision Center on WebSphere Application Server (3 of 3)

Step 5: Verifying the deployment of the Decision Center Enterprise console	After you finish configuring Decision Center for your application server, verify that you deployed the archive successfully.
Step 6: Completing the configuration of the Decision Center consoles	After you set database user permissions, created a data source, and deployed the Decision Center EAR file, you complete the configuration either from the Decision Center console or by running Ant tasks.
Optional steps to configure the Decision Center Business console	The search function in the Decision Center Business console is based on the Solr search engine. You must configure the engine to provide this functionality.

© Copyright IBM Corporation 2015

Figure 2-15. Configuring Decision Center on WebSphere Application Server (3 of 3)

WB393 / ZB3931.0

Notes:

2.4. Completing the installation

Completing the installation



© Copyright IBM Corporation 2015

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

10.1

Figure 2-16. Completing the installation

WB393 / ZB3931.0

Notes:



Installation Settings wizard

- After profile configuration is finished for Rule Execution Server and Decision Center applications, you must complete the installation in the consoles for these modules with the Installation Settings wizard
- This step is the same regardless of whether you configured manually or used a profile template

© Copyright IBM Corporation 2015

Figure 2-17. Installation Settings wizard

WB393 / ZB3931.0

Notes:




Completing the Rule Execution Server installation (1 of 2)

- Open the console: `http://host_name:port/res`
 - Use the port that is defined for this application server profile
 - The port is assigned when you create the profile
 - Sign in as the administrator

Administrative console port: 9064

Administrative console secure port: 9047

HTTP transport port: 9084 (highlighted with a red box)

HTTPS transport port: 9447

Bootstrap port: 2813

SOAP connector port: 8884

Installation Settings Wizard

RuleApps persistence details

Database Product Name	Apache Derby
Database Product Version	10.8.3.1 - (1452645)
Driver Name	Apache Derby Embedded JDBC Driver
Driver Product Version	10.8.3.1 - (1452645)
JDBC URL	<code>jdbc:derby:C:\Program Files\IBM\ODM87\WAS\AppServer\profiles\AppSrv01\databases\resdb</code>
Schemas	APP, NULLID, SQL1, SYS, SYSCAT, SYSCS_DIAG, SYSCS_UTIL, SYSFUN, SYSIBM, SYSPROC, SYSSTAT
Username	resAdminDB

© Copyright IBM Corporation 2015

Figure 2-18. Completing the Rule Execution Server installation (1 of 2)

WB393 / ZB3931.0

Notes:



Completing the Rule Execution Server installation (2 of 2)

- For Rule Execution Server, the Installation Settings wizard helps you create the correct type of database schemas

Step 1: RuleApps - Welcome to the Installation Settings Wizard of Rule Execution Server

Use the Installation Settings Wizard to create resources in your database schema so that your repository can be loaded.

The repository has not been loaded because of the following reasons:

```

The persistence check failed. Diagnostic report:
DAO Class Name = ilog.rules.res.persistence.impl.jdbc.IlrGenericRepositoryDAO
Database Product Name = Apache Derby
Database Product Version = 10.8.3.1 - (1452645)
Driver Name = Apache Derby Embedded JDBC Driver
Driver Product Version = 10.8.3.1 - (1452645)
Ruleset enabled view test passed = False
RuleApp properties table test passed = False
RuleApps table test passed = False
Ruleset properties table test passed = False
Ruleset resources table test passed = False
Rulesets table test passed = False
Is Transaction Supported = True
JDBC URL = jdbc:derby:C:\Program Files\IBM\ODM87\WA5\AppServer\profiles\AppSrv01\databases\resdb
Username = resAdminDB.
  
```

Click **Next** to select a database schema type.

[Back](#) [Next](#)

© Copyright IBM Corporation 2015

Figure 2-19. Completing the Rule Execution Server installation (2 of 2)

WB393 / ZB3931.0

Notes:



Completing the Decision Center installation (1 of 2)

- Open the console: `http://host_name:port/teamserver`
 - Use the port that is defined for this application server
- Sign in as the administrator
 - The Installation Settings wizard opens automatically

Installation Settings Wizard

Use the Installation Settings Wizard to complete or modify the installation. To complete an initial installation, click Next. To modify an existing installation, click on the corresponding step

[Click here for help on using the Installation Settings Wizard](#)

[Exit](#) | [Previous](#) | [Next](#) | [Finish](#)

© Copyright IBM Corporation 2015

Figure 2-20. Completing the Decision Center installation (1 of 2)

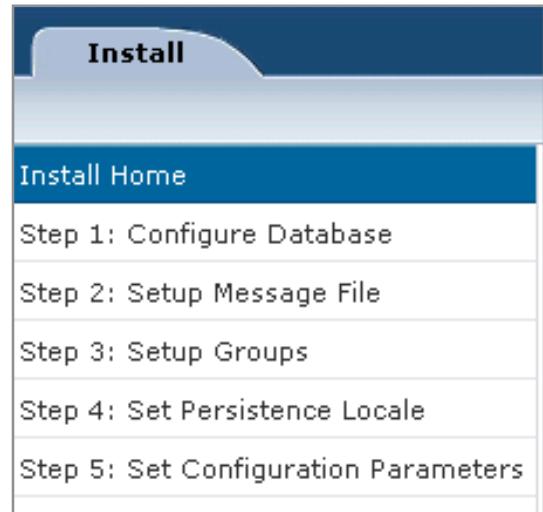
WB393 / ZB3931.0

Notes:



Completing the Decision Center installation (2 of 2)

- For Decision Center, the Installation Settings wizard helps you create the correct groups in your application server when you set up security access
- Decision Center is ready for you to import projects



© Copyright IBM Corporation 2015

Figure 2-21. Completing the Decision Center installation (2 of 2)

WB393 / ZB3931.0

Notes:

2.5. Profile templates for WebSphere Application Server

Profile templates for WebSphere Application Server



© Copyright IBM Corporation 2015

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

10.1

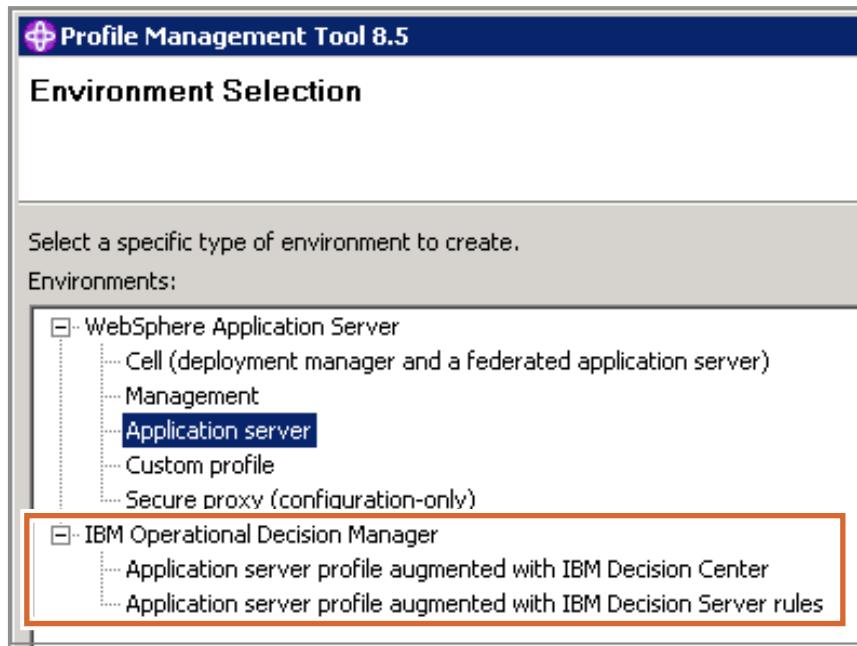
Figure 2-22. Profile templates for WebSphere Application Server

WB393 / ZB3931.0

Notes:

Using profile templates in the Profile Management tool

- Automatically deploy and configure a new application server profile with Decision Server Rules, Decision Server Events, or Decision Center



© Copyright IBM Corporation 2015

Figure 2-23. Using profile templates in the Profile Management tool

WB393 / ZB3931.0

Notes:

You can use the Profile Management Tool to create stand-alone servers or to augment an existing server profile with Decision Server Rules (Rule Execution Server), Decision Center, or Decision Server Events.

Decision Server Rules profile template files

- When you install Decision Server rules, the installer copies files for cluster configuration to these directories:

ODM_InstallDir/executionserver/
applicationservers/WebSphere85

- The EAR files for the Rule Execution Server console, Scenario Service Provider (SSP), and hosted transparent decision services
- The RAR file for the Execution Unit (XU)

ODM_InstallDir/shared/profiles/
profileTemplates/rules

- The Decision Server profile template to augment a deployment manager profile
- The `management/ds` subdirectory that contains the scripts to augment a deployment manager profile

WAS_InstallDir/profileTemplates/
rules

- The `management/ds` subdirectory that contains the scripts to augment a deployment manager profile

© Copyright IBM Corporation 2015

Figure 2-24. Decision Server Rules profile template files

WB393 / ZB3931.0

Notes:

ODM_InstallDir: Installation directory for Decision Server

WAS_InstallDir: Installation directory for WebSphere Application Server

Decision Center profile template files

- When you install Decision Center, the installer copies files for cluster configuration to these directories:

<i>ODM_InstallDir/teamserver/applicationservers/WebSphere85</i>	<ul style="list-style-type: none"> The Decision Center profile template to augment a deployment manager profile and the EAR files
<i>ODM_InstallDir/shared/profiles/profileTemplates/rules</i>	<ul style="list-style-type: none"> The Decision Center profile template to augment a deployment manager profile
<i>ODM_InstallDir/shared/profiles/profileTemplates/rules</i>	<ul style="list-style-type: none"> At profile augmentation, the <code>management/dc</code> subdirectory that contains the scripts to augment a deployment manager profile
<i>WAS_InstallDir/profileTemplates/rules</i>	<ul style="list-style-type: none"> At profile augmentation, if you specified <code><WAS_InstallDir></code>, the installer also copies to this directory the <code>management/dc</code> subdirectory that contains the scripts to augment a deployment manager profile

© Copyright IBM Corporation 2015

Figure 2-25. Decision Center profile template files

WB393 / ZB3931.0

Notes:

ODM_InstallDir: Installation directory for Decision Server

WAS_InstallDir: Installation directory for WebSphere Application Server

Cluster scripts

- For high availability and scalability of your applications, you can configure Rule Execution Server and Decision Center on a WebSphere Application Server cluster by running a configuration script
- You must first augment a deployment manager profile so that you can create other nodes by using the provided template
 - Run the `configureDSCluster` script to deploy Rule Execution Server to a cluster
 - Run the `configureDCCluster` script to deploy Decision Center to a cluster
- When the script finishes, the cluster is up and running
 - You should then configure the load-balancing system, such as IBM HTTP Server

© Copyright IBM Corporation 2015

Figure 2-26. Cluster scripts

WB393 / ZB3931.0

Notes:

See the product documentation:

- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.config.was/config_ds_res_was/tsk_was_ds_cluster_run_script.html
- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.config.was/config_dc_websphere/tsk_was_dc_cluster_run_script.html



Running the Decision Server script

The configureDSCluster script performs these actions:

- Install the JDBC provider, execution unit (XU), and data source at node level
- Install the Rule Execution Server console to the cluster
- Deploy the hosted transparent decision services and Scenario Service Provider (SSP) to the cluster member
 - Users are mapped to application groups when an application is deployed
- Start the deployment manager server if it is not already started
- Configure security
- Create the resAdmin, resDeployer, and resMonitor users
 - These users belong to the WebSphere Application Server monitor so that JMX notifications work on all cluster nodes
- Configure users and groups
- Map users and groups to roles
- Start the cluster, servers, and applications

© Copyright IBM Corporation 2015

Figure 2-27. Running the Decision Server script

WB393 / ZB3931.0

Notes:



Running the Decision Center script

The configureDCCluster script performs the following actions:

- Install the JDBC provider and the data source at node level
- Install the Decision Center application at the cluster level
 - Users are mapped to application groups when an application is deployed
- Start the deployment manager server if it is not already started
- Configure security
- Create the rtsAdmin, rtsInstaller, rtsUser1, and rtsConfig users
- Configure users and groups
- Map users and groups to roles
- Start the cluster, servers, and applications

© Copyright IBM Corporation 2015

Figure 2-28. Running the Decision Center script

WB393 / ZB3931.0

Notes:

2.6. Configuring Rule Execution Server for production

Configuring Rule Execution Server for production



© Copyright IBM Corporation 2015

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

10.1

Figure 2-29. Configuring Rule Execution Server for production

WB393 / ZB3931.0

Notes:

Configuring Rule Execution Server in different environments (1 of 2)

- Development lifecycle requires Rule Execution Server deployment to multiple environments for various stakeholders
 - Development
 - Testing
 - Production
- A production environment requires more than a single deployment of Rule Execution Server
- To set up different environments in a single cell:
 - Set up different data sources
 - Deploy and configure an XU for each environment and define a J2C connection factory
 - Deploy the Rule Execution Server console for each environment
 - Restart the node agents after you complete the configuration
 - Call the XU instances to register the XU with the Rule Execution Server console

© Copyright IBM Corporation 2015

Figure 2-30. Configuring Rule Execution Server in different environments (1 of 2)

WB393 / ZB3931.0

Notes:

The development lifecycle of a business rule application is similar to any other software development process, and includes stages for implementation, testing, deployment, and maintenance. At a minimum, you are likely to need an environment for your development team, one for your QA team, and another one for in-production applications.

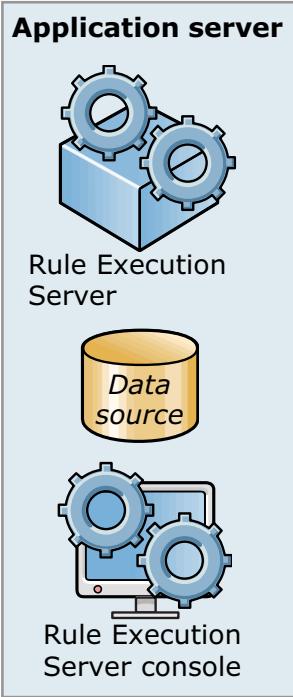
When you configure Rule Execution Server in a single cell, it is a good practice to isolate the rulesets on each server, and ensure that Execution Units (XUs) do not interfere with each other.

Configuring Rule Execution Server in different environments (2 of 2)

Development and testing



Designer



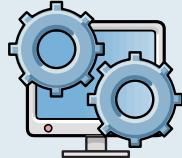
Production

Main node

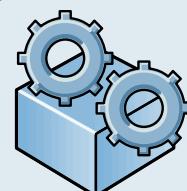
Failover node

Application server cluster

Application server



Application server



© Copyright IBM Corporation 2015

Figure 2-31. Configuring Rule Execution Server in different environments (2 of 2)

WB393 / ZB3931.0

Notes:



Setting up Rule Execution Server for production

- System administrators are responsible for setting up and installing the hardware and software that is used to run Rule Execution Server
- A production application server often requires multiple Rule Execution Server deployments on a cluster spread over multiple machines
- The simplest production server consists of a cluster with a main node and a failover node
 - Install an application server on each node to create the cluster
 - Install Rule Execution Server on each application server
 - All Rule Execution Server instances in the cluster are managed by using a single Rule Execution Server console that runs on the main node

© Copyright IBM Corporation 2015

Figure 2-32. Setting up Rule Execution Server for production

WB393 / ZB3931.0

Notes:

2.7. Clustering ODM servers for high availability

Clustering ODM servers for high availability



© Copyright IBM Corporation 2015

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

10.1

Figure 2-33. Clustering ODM servers for high availability

WB393 / ZB3931.0

Notes:



Clustering on WebSphere Application Server

- A cluster is a group of application server instances that run your Java EE application as if it were a single entity
- Clusters provide workload balancing and failover support to ensure minimal downtime and maximum scalability
- Use scripts to create and configure Decision Server clusters or Decision Center clusters

© Copyright IBM Corporation 2015

Figure 2-34. Clustering on WebSphere Application Server

WB393 / ZB3931.0

Notes:

Operational Decision Manager provides scripts that you can run to create and configure Decision Server clusters or Decision Center clusters for high availability.



WebSphere Application Server clustering and the ideal topology

- An *ideal topology* is typically configured for a particular usage scenario
 - Balances availability and scalability requirements, and other desirable system characteristics
 - Simplifies planning by providing a model that is based on the experience of knowledgeable Operational Decision Manager users
- Two ideal topologies for WebSphere Application Server cells:
 - Topology based on Operational Decision Manager cells to facilitate rule authoring
 - Topology based on Decision Server cells that are intended for the rules and event runtime environments

© Copyright IBM Corporation 2015

Figure 2-35. WebSphere Application Server clustering and the ideal topology

WB393 / ZB3931.0

Notes:

Planning a clustered topology with WebSphere Application Server Network Deployment can be complicated because many combinations are possible but only a subset suits your needs. A *gold* topology helps you to make the right choices.

ODM ideal topology

- Operational Decision Manager cell
 - Enables all of the applications that Operational Decision Manager provides
 - Includes a Decision Center cluster and a Decision Server cluster

Operational Decision Manager cell

Cell

Decision Center cluster

Decision Server cluster

- Decision Server cell
 - Contains the subset of Operational Decision Manager components that support the rules and event runtime execution environments and testing capabilities

Decision Server cell

Cell

Decision Server cluster

© Copyright IBM Corporation 2015

Figure 2-36. ODM ideal topology

WB393 / ZB3931.0

Notes:

Operational Decision Manager cell

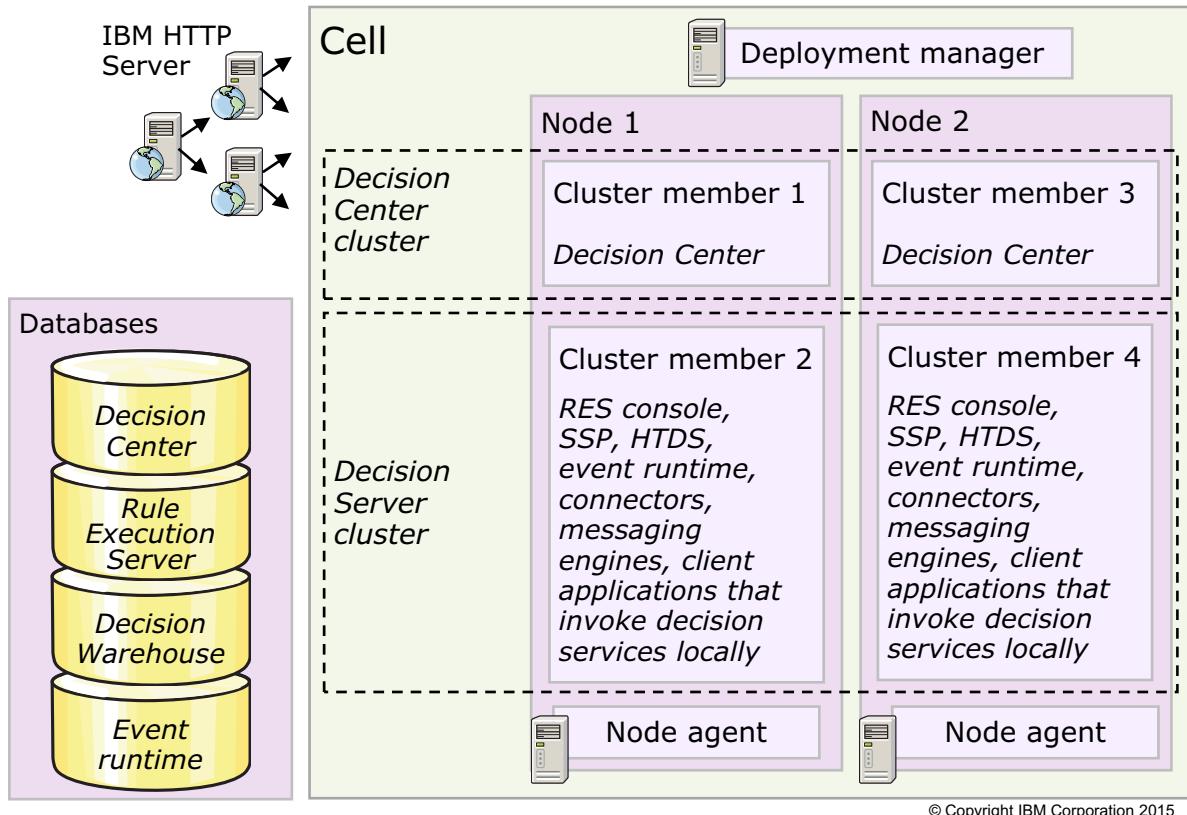


Figure 2-37. Operational Decision Manager cell

WB393 / ZB3931.0

Notes:

The Operational Decision Manager ideal topology cell includes the full functionality of Decision Center and Decision Server.

The Operational Decision Manager cell can be used to stage your authoring, testing, and production environments. If some Decision Server components are not relevant to your particular requirements, you can remove them to reduce resource usage. Each node contains a Decision Server instance and a Decision Center instance.

As shown in the diagram, the Decision Server instances in the cell form a cluster. The Decision Center instances in the cell form a separate cluster. Members of the same cluster are configured identically to enable workload balancing and failover support.

Decision Center cluster

This cluster contains all the server-side components of Decision Center for business authoring and simulation management. It also contains an EAR file for the web consoles and an EAR file for the event widgets.

Decision Server cluster

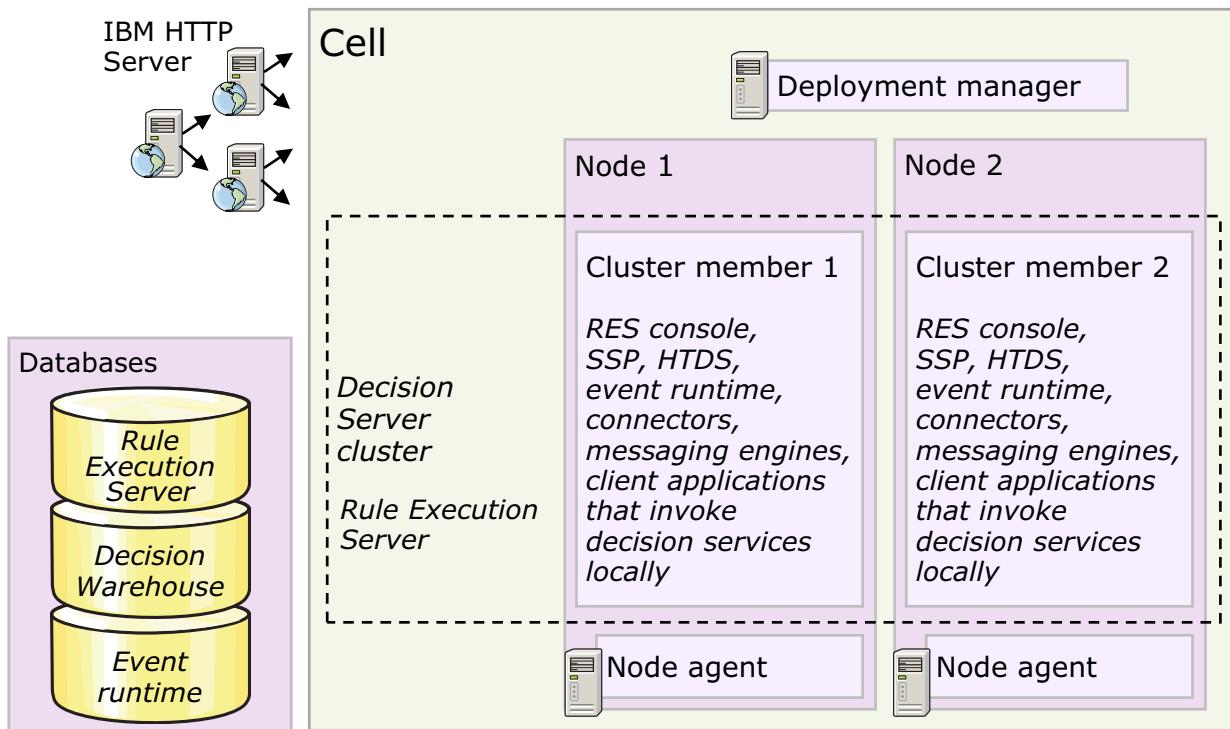
The Decision Server cluster includes all of the components for executing business rules and event logic. It also handles the execution of business rule test suites and simulations. This cluster includes the following Java EE artifacts:

- An EAR file for the Rule Execution Server console that is deployed to the cluster
- A RAR file for the rules execution unit that is deployed at the node level
- An EAR file that contains the Scenario Service Provider, which executes business rule test suites and simulations
- An EAR file that contains the hosted transparent decision service, which dynamically publishes and runs decision web services
- An event runtime to execute and test event rules
 - The event runtime includes runtime and test EAR files
 - The test EAR file must be on the same cluster as the runtime EAR file
- Event connectors
- Message engines that are required for event processing and that are used by message-driven rule beans (MDBs)

Rule and event runtimes are collocated in the same JVMs. To enable a decision service that is triggered by an event detection, you must place the two engines in the same JVM for a local invocation.

Node 1 in this diagram includes a Rule Execution Server console that is shared by the Decision Server instances. The Rule Execution Server console is deployed as an EAR file on a separate server outside the Decision Server cluster.

Decision Server ideal topology cell



© Copyright IBM Corporation 2015

Figure 2-38. Decision Server ideal topology cell

WB393 / ZB3931.0

Notes:

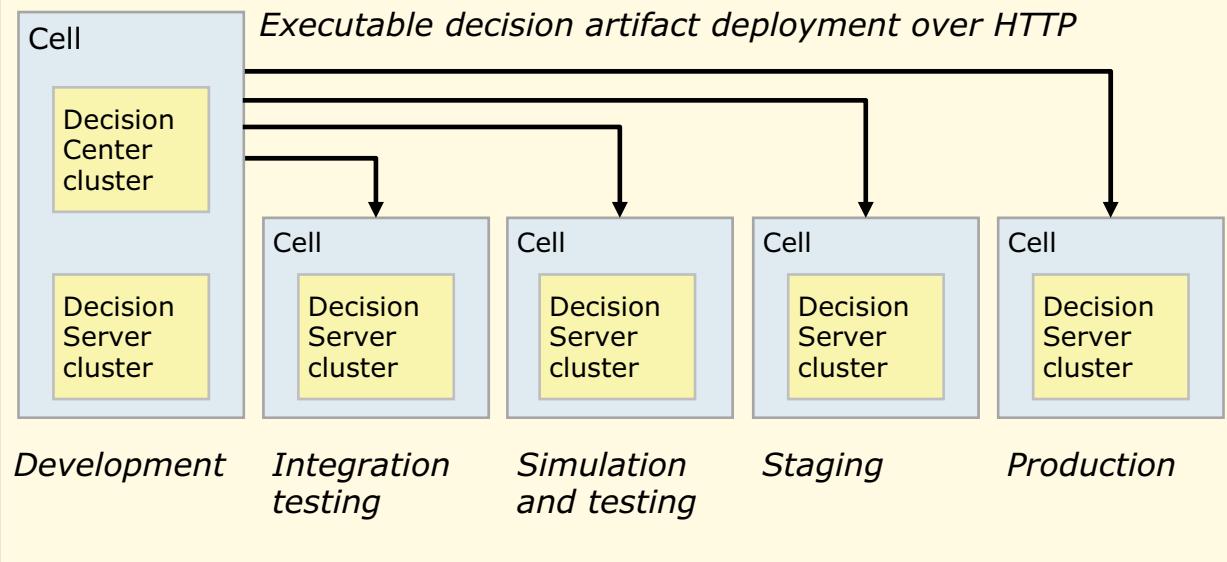
The Decision Server ideal topology cell includes the Decision Server Rules and Decision Server Events runtimes.

The Decision Server cell can be used to host your authoring, testing, and production environments. If some Decision Server components are not relevant to your particular requirements, you can remove them to minimize resource usage.

As shown in the diagram, each node contains a Decision Server instance. The Decision Server instances in each cell form a cluster. The members of the cluster are configured identically to enable failover support and workload balancing.

Shared environment

Decision Center serving multiple Decision Server instances



© Copyright IBM Corporation 2015

Figure 2-39. Shared environment

WB393 / ZB3931.0

Notes:

Shared environment is a single authoring server that feeds multiple runtime servers, which are specialized and isolated by lifecycle phases. You deploy the executable artifacts to run business rules and events over HTTP and HTTPS. A shared environment has the following advantages:

- Provides functional isolation between business authoring and execution
- Offers a single source for authoring and deployment
- Supports branching and merging so that you can deploy executable rules on selected servers
- Enables high availability for all Operational Decision Manager capabilities
- Separates execution and simulation workloads

A shared environment has the following disadvantages:

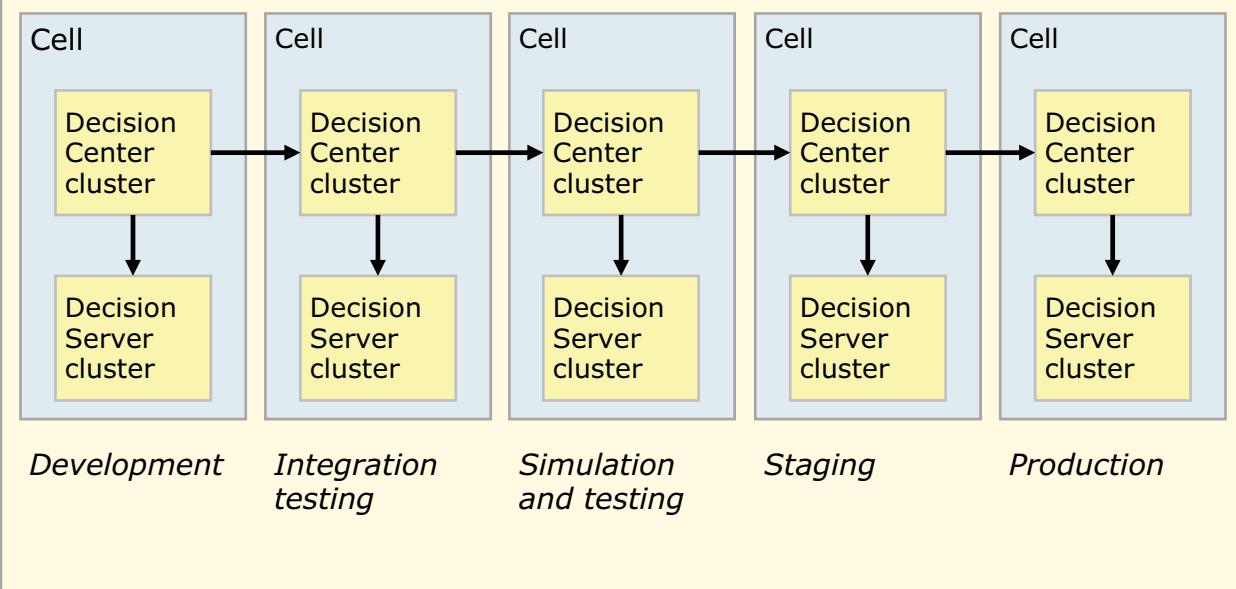
- Lacks the ability to customize Decision Center for a single phase (development, test, or production).
- Introduces a single point of failure for authoring because of the shared Decision Center repository. You can use a database replication to avoid this weakness.

- Requires management of access to actions and servers.

This diagram shows how you can use a single Decision Center to propagate decision artifacts to multiple Decision Server instances.

Staged environment

Operational Decision Manager staged environment



© Copyright IBM Corporation 2015

Figure 2-40. Staged environment

WB393 / ZB3931.0

Notes:

Each cell can be managed independently and includes a Decision Center and a Decision Server cluster. A staged environment has the following advantages:

- Provides full isolation between development lifecycle stages
- Isolates authoring and execution by stage and cell
- Offers the ability to customize Decision Center in each cell, including security
- Supports high availability for clustered Decision Center and Decision Server instances

A staged environment has the following disadvantages:

- Requires the provisioning and management of multiple JVMs and Decision Center databases
- Requires the synchronization of Decision Center repository content across cells from the development cell to the production cell

You must propagate rule and event projects from one stage to another by using Export and Import commands. You can automate the propagation process by using the Java APIs and Ant tasks provided.

This diagram shows how the Operational Decision Manager cell is used in each stage of the development lifecycle.

2.8. IBM ODM Pattern V8.7.1

IBM ODM Pattern V8.7.1



© Copyright IBM Corporation 2015

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

10.1

Figure 2-41. IBM ODM Pattern V8.7.1

WB393 / ZB3931.0

Notes:



IBM ODM Pattern

- Provides common patterns that enable the ODM environment to run in a private cloud
- Optimized to run in IBM PureApplication System V2.1.0.0 and later fix packs
 - PureApplication System provides generic framework for designing, deploying, and managing virtual applications
- PureApplication System provides:
 - Virtualized application workloads
 - Scalable infrastructure
- Delivered as an additional deployment option with IBM Operational Decision Manager Advanced

© Copyright IBM Corporation 2015

Figure 2-42. IBM ODM Pattern

WB393 / ZB3931.0

Notes:

With cloud computing, users can access applications or computing resources as services from anywhere through their connected devices by using a simplified user interface and API, which is the infrastructure that supports such services. Data and services can then be accessed from the cloud through connected devices over the Internet.

IBM Operational Decision Manager provides common patterns that enable the Operational Decision Manager environment to run in a private cloud.

IBM PureApplication System is an integrated hardware and software system that provides system-centric and application-centric computing models in a cloud environment. By using PureApplication System, you can efficiently manage complex applications, tasks, and processes that the application starts. The system implements a virtual-computing environment in which different resource configurations are tailored to different workloads.

IBM ODM Pattern 8.7.1 artifacts

- Software components
 - IBM ODM 871 Deployment Manager
 - IBM ODM 871 Custom Node
 - IBM ODM 871 HTTP Server
 - IBM ODM 871 Decision Center
 - IBM ODM 871 Decision Server Rules
 - IBM ODM 871 Decision Server Events
 - IBM ODM 871 WebSphere eXtreme Scale
- Base scaling policy
 - Enables dynamic vertical scaling for an instance
 - Policy can be applied globally at system level or specified for individual components
- Script packages:
 - Configure Decision Center
 - Configure Rules
 - Configure Events
 - Configure Custom Node
 - Others (for example, configure IBM HTTP Server, configure IBM Security Directory Server, install RuleApps)

© Copyright IBM Corporation 2015

Figure 2-43. IBM ODM Pattern 8.7.1 artifacts

WB393 / ZB3931.0

Notes:

IBM PureApplication System provides the framework to deploy the ODM virtual applications into the cloud and convert it from a logical model to a topology of virtual machines. ODM virtual application patterns provide artifacts that define a running ODM environment after the pattern is deployed.

Software components

Software components install the software and operations that are required to create an operational IBM ODM environment. To provide a working IBM ODM environment in the cloud, you define the properties of the related software components. When the patterns that contain those software components are deployed, the software components become virtual machines in the virtual system instance.

The ODM Application Pattern 8.7.1 for Virtual System 1.0 comprises these components:

- IBM ODM 871 Deployment Manager
 - Administrative manager of an IBM ODM deployment cell
- IBM ODM 871 Custom Node software component

- Contains a node agent and managed IBM ODM servers, which are federated to an IBM ODM deployment manager
- IBM ODM 871 HTTP Server
 - Composed of a web server instance; distributes the work among the nodes in IBM ODM Pattern
- IBM ODM 871 Decision Center
 - Includes a rule repository and collaborative web consoles for business users to author, manage, validate, and deploy rules
- IBM ODM 871 Decision Server Rules
 - Automate and govern frequent and repeatable business decisions
- IBM ODM 871 Decision Server Events
 - Orchestrate complex business event processing
- IBM ODM 871 WebSphere eXtreme Scale
 - Improve performance and scalability when complex business events are processed by filtering large volumes of events into a smaller set of events
 - The set of filtered events can be forwarded to Decision Server Events for processing

Script packages

When IBM ODM Pattern packages are installed, script packages are added to virtual system pattern topology parts to customize the behavior of virtual system pattern topology parts. Script packages contain executable files and artifacts.

- Configuration scripts for Decision Center:
These scripts configure ODM Deployment Manager for Decision Center on either DB2 or Oracle.
 - IBM ODM 871 Config Dmgr for DC on DB2
 - IBM ODM 871 Config Dmgr for DC on Oracle
- Configuration scripts for Rules:
These scripts configure ODM Deployment Manager for Decision Server Rules on either DB2 or Oracle.
 - IBM ODM 871 Config Dmgr for Rules on DB2.
 - IBM ODM 871 Config Dmgr for Rules on Oracle.
- Configuration scripts for Events:
These scripts configure ODM Deployment Manager for Decision Server Events on either DB2 or Oracle.
 - IBM ODM 871 Config Dmgr for Events on DB2.
 - IBM ODM 871 Config Dmgr for Events on Oracle.
- Configuration scripts for Custom Node:
These scripts configure the ODM custom node. They should be used with the IBM ODM 871

Custom Node software component, and the appropriate software component for the script (Decision Center, Decision Server Rules, or Decision Server Events).

- IBM ODM 871 Config Custom Node for DC
- IBM ODM 871 Config Custom Node for Events
- IBM ODM 871 Config Custom Node for Rules
- Other scripts are also available:
 - IBM ODM 871 Config IHS
 - IBM ODM 871 Config TDS
 - IBM ODM 871 Install RuleApp
 - IBM ODM 871 Config Standby Dmgr
 - IBM ODM 871 Config Auxiliary DB2

For more information about script packages, see the product documentation for IBM ODM Pattern V8.7.1.

Links

Each of the IBM ODM links listed here describes a connection between two of the components:

- Link from Decisions component to Decision Center component
- Link from Decision Center component to IBM ODM Database component
- Link from Decisions component to IBM ODM Database component

The direction of the link defines the dependency relationship. For example, the Decision Center component depends on the Operational Decision Manager database component.

Base Scaling Policy

You can add a base scaling policy to a software component in a virtual system pattern. A base scaling policy helps to relieve heavy workloads by enabling dynamic vertical scaling for an instance. The policy can be applied in two ways:

- Globally at the system level
- Specifically for individual components

Two virtual applications might include identical components but require different policies to achieve different service-level agreements. For example, you can add a decision management scaling policy to the application component and specify requirements, such as a processor usage threshold to trigger scaling the application. At deployment time, the topology of the virtual application is configured to dynamically scale the application.

Some base scaling properties apply only to automatic scaling, and other properties apply to both automatic and manual scaling. For more information about the base scaling policy and its properties, see the product documentation for ODM Pattern V8.7.1.



IBM ODM Pattern V8.7.1 contents

- Pattern installation package includes:
 - IBM ODM product Installation Manager repository
 - IBM ODM V8.7.1 Installation media
 - IBM ODM pattern type definition
 - IBM ODM pattern plug-ins
- Operating system is not packaged with the IBM ODM Pattern package
 - Software components and operating system images are separate
 - PureApplication System provides the common base image for deployments

© Copyright IBM Corporation 2015

Figure 2-44. IBM ODM Pattern V8.7.1 contents

WB393 / ZB3931.0

Notes:

The pattern package provides operational decision management capabilities that are configured for typical decision management projects.



Virtual system patterns

- Logical representation of a recurring topology for a set of deployment requirements
 - Enable efficient and repeatable deployment of systems
 - Systems include one or more virtual machine instances and the applications that run on them
- Use a predefined virtual system pattern or build your own
 - Predefined patterns provide starting points for deploying ODM environments
 - Can configure your own virtual system pattern in the IBM Pattern Builder
- Deploy and manage system pattern in a cloud environment
 - Deployed system pattern is a working IBM ODM environment
 - Components in the patterns become running virtual machines in the virtual system instance

© Copyright IBM Corporation 2015

Figure 2-45. Virtual system patterns

WB393 / ZB3931.0

Notes:

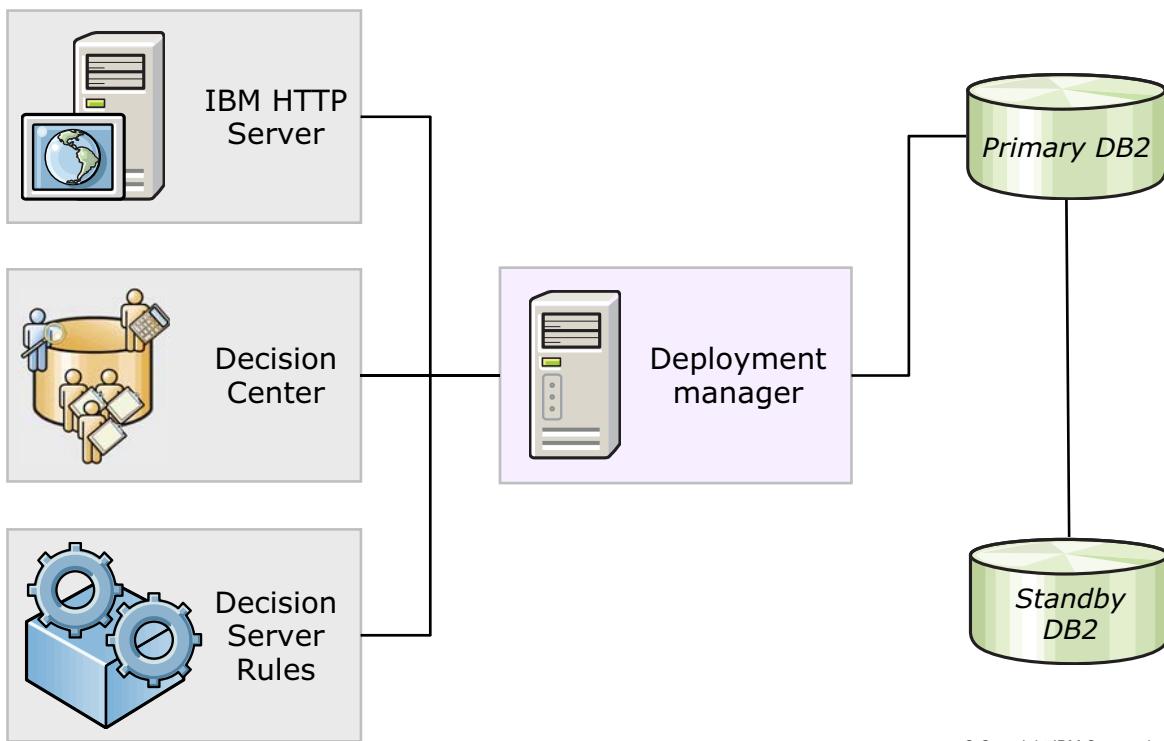
A *pattern type* is a collection of plug-ins that define artifacts that are packaged in a .tgz file. Based on pattern type, IBM Operational Decision Manager Pattern uses artifacts to create an IBM ODM environment.

An *artifact* is an entity that is used or produced by a software development process.

IBM ODM allows you to either build your own virtual system pattern or use a predefined virtual system pattern. A deployed pattern is called an *instance*. A deployed pattern is converted from a logical model to a topology of virtual machines that run in the cloud.

You can view and administer the virtual machines in the virtual system instance, which is an ODM environment.

Example: Predefined system pattern for IBM ODM with DB2 for HADR



© Copyright IBM Corporation 2015

Figure 2-46. Example: Predefined system pattern for IBM ODM with DB2 for HADR

WB393 / ZB3931.0

Notes:

This slide shows an example of a predefined virtual system pattern for IBM ODM with DB2 for HADR.

The default pattern includes:

- Decision Center custom nodes and Decision Server custom nodes with two-cluster topology
- DB2 HADR
- Automatic scaling policies



Virtual system instances

- Virtual system instance results from deploying virtual system pattern into the cloud
 - Is a working ODM environment
- View and manage virtual system instance and the virtual machines that are running in it
 - Manage virtual system instance itself
 - Manage virtual machines
 - Manage HTTP endpoints
 - Change processor count or memory size
 - Work with scaling, whether automatic or manual
 - Start or stop servers

© Copyright IBM Corporation 2015

Figure 2-47. Virtual system instances

WB393 / ZB3931.0

Notes:

When you deploy a virtual system pattern into the cloud, the resulting virtual system instance is a working IBM ODM environment. You can view and manage the virtual system instance and the virtual machines that are running in it to manage your IBM ODM environment.



Unit summary

Having completed this unit, you should be able to:

- Configure Rule Execution Server and Decision Center
- Explain how to deploy Decision Server and Decision Center for high availability and scalability
- Describe the IBM ODM Application Pattern

© Copyright IBM Corporation 2015

Figure 2-48. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** After installing Decision Center and Rule Execution Server, you must manually complete the installation.
2. **True or False:** To run Rule Execution Server in a cluster, you must deploy a Rule Execution Server console to each node in the cluster.
3. **True or False:** Decision Center cannot be configured for high availability.

© Copyright IBM Corporation 2015

Figure 2-49. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

- 1. True.**
- 2. False.** *You can deploy Rule Execution Server console to a single node to monitor the entire cluster.*
- 3. False.** *Decision Center can be deployed to a cluster. Profile templates and cluster scripts are provided.*

© Copyright IBM Corporation 2015

Figure 2-50. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 2



Configuring Rule Execution Server
on WebSphere Application Server

© Copyright IBM Corporation 2015

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

10.1

Figure 2-51. Exercise 2

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Configure Rule Execution Server

© Copyright IBM Corporation 2015

Figure 2-52. Exercise objectives

WB393 / ZB3931.0

Notes:

Exercise 3



Configuring Decision Center on
WebSphere Application Server

© Copyright IBM Corporation 2015

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

10.1

Figure 2-53. Exercise 3

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Configure Decision Center

© Copyright IBM Corporation 2015

Figure 2-54. Exercise objectives

WB393 / ZB3931.0

Notes:

Exercise 4



Using a profile template to configure
Rule Execution Server

© Copyright IBM Corporation 2015

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

10.1

Figure 2-55. Exercise 4

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Configure Rule Execution Server on a new profile by using the profile template

© Copyright IBM Corporation 2015

Figure 2-56. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 3. Managing user roles and permissions

What this unit is about

In this unit, you learn how to manage security and permissions, and how to customize user access in Decision Center.

What you should be able to do

After completing this unit, you should be able to:

- Customize user access
- Administer security in Decision Center

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Customize user access
- Administer security in Decision Center

© Copyright IBM Corporation 2015

Figure 3-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Application server security
- Configuring security for ODM modules on WebSphere Application Server
- Configuring single sign-on (SSO)
- Defining custom groups for Decision Center
- Administering security in Decision Center
- Troubleshooting security in Decision Center

© Copyright IBM Corporation 2015

Figure 3-2. Topics

WB393 / ZB3931.0

Notes:

Approaching ODM as an administrator

- Responsibilities:
 - Deploying and configuring the server and database for Decision Center and Rule Execution Server
 - Managing user access to Decision Center and Rule Execution Server
 - Configuring trace data sources for testing purposes
 - Deploying applications
 - Redeploying rulesets and event assets as changes are made
 - Generating detailed execution reports
 - Tracking and monitoring rule execution
 - Restoring a particular application state
- Tools: Servers for Decision Center or runtime environments



Administrator

© Copyright IBM Corporation 2015

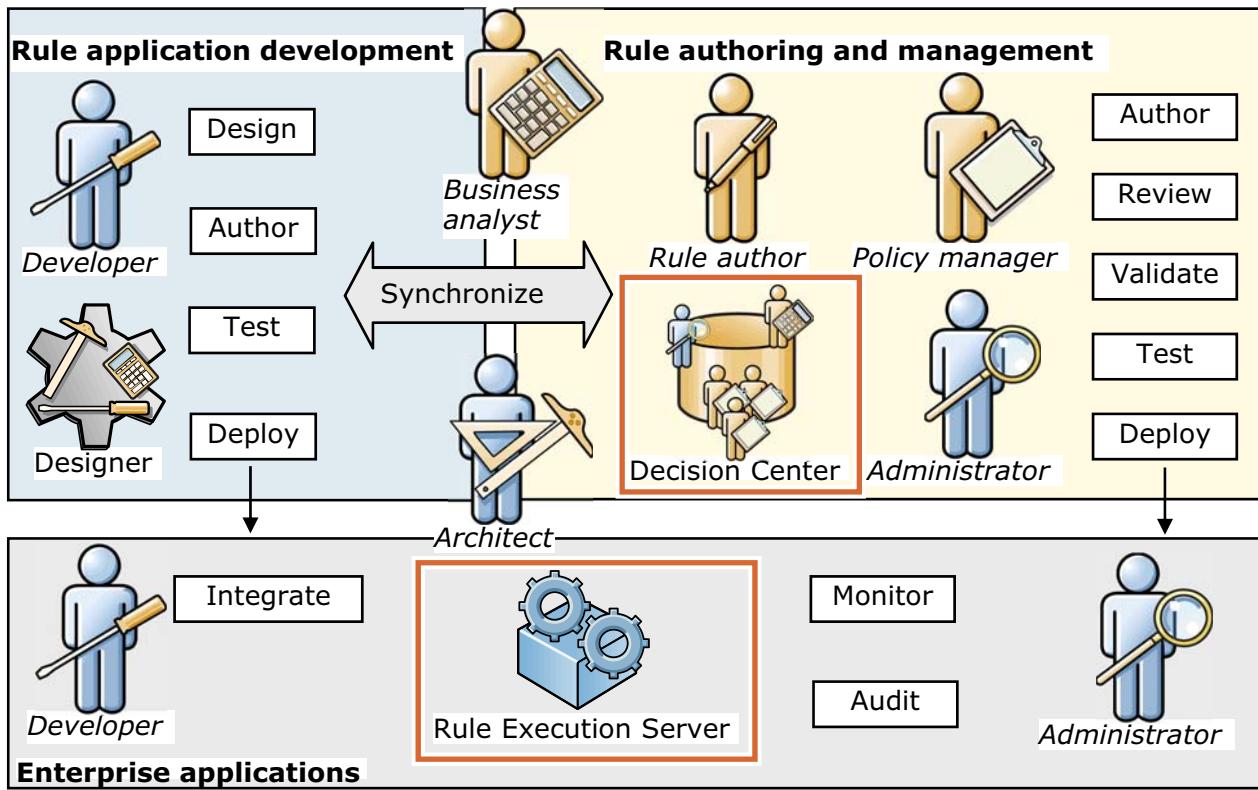
Figure 3-3. Approaching ODM as an administrator

WB393 / ZB3931.0

Notes:

As administrator, you are responsible for managing user access to Decision Center and Rule Execution Server.

Recall: Roles and activities



© Copyright IBM Corporation 2015

Figure 3-4. Recall: Roles and activities

WB393 / ZB3931.0

Notes:

Recall that various roles require varying levels of access to projects and servers.

- Business users (rule authors, policy managers) require basic access to Decision Center. Business analysts might need more permissions if they intend to deploy projects.
- Developers must have administrator access to Decision Center so that they can make sure that the correct artifacts are synchronized with the business environment and provide Decision Center customizations.
- Developers also require administrator access to Rule Execution Server so that they can integrate managed execution into their application and monitor deployment.

As an administrator, you ensure that the correct levels of access are granted to these roles. You can also help business users enforce security on their projects by helping them to specify fine-grained permissions on rule artifacts within the project.

3.1. Application server security

Application server security



© Copyright IBM Corporation 2015

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

10.1

Figure 3-5. Application server security

WB393 / ZB3931.0

Notes:

Secure mode on the application server

- To help you configure applications safely, application servers check access to the resources and manage access rights
- Operational Decision Manager is certified to support these standards:
 - Federal Information Processing Standards (FIPS) 140-2
 - National Institute of Standards and Technology (NIST) Special Publications 800-131a
- Configuration of a secure mode on an application server provides ways to increase the security of the application that you deploy
- In secure mode, an application server:
 - Checks access to the resources: Access to a class by using the Java reflection mechanism is not permitted without the appropriate security rights
 - Manages access rights: Rule Execution Server console access is managed with this mechanism

© Copyright IBM Corporation 2015

Figure 3-6. Secure mode on the application server

WB393 / ZB3931.0

Notes:

The FIPS 140-2 standard is an information technology security accreditation program to certify cryptographic modules for use in government departments and regulated industries that collect, store, transfer, share, and disseminate sensitive but unclassified information.

The National Institute of Standards and Technology (NIST) Special Publications (SP) 800-131a standard strengthens algorithms and increases the key lengths to improve security. The standard also provides for a transition period to move to the new standard. The transition period enables a user to run in a mixed environment of settings that are not supported under the standard along with settings that are supported. The NIST SP800-131a standard requires that users be configured for strict enforcement of the standard by a specific time frame. For more information, see The National Institute of Standards and Technology website:

<http://csrc.nist.gov/publications/PubsFIPS.html>

WebSphere Application Server can be configured to run SP800-131a in a transition mode or a strict mode. For more information about configuring transition mode, see the topic "Transitioning WebSphere Application Server to the SP800-131 Security Standard" in the product documentation.



Security layers in WebSphere Application Server

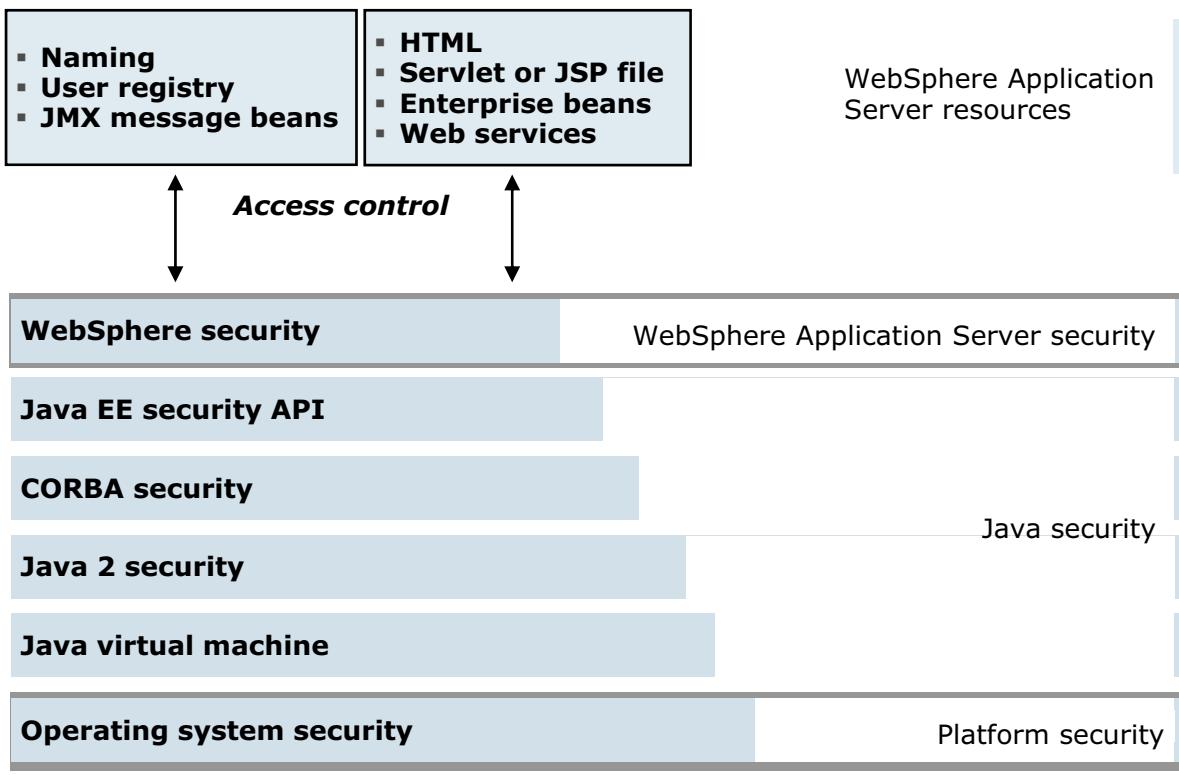


Figure 3-7. Security layers in WebSphere Application Server

WB393 / ZB3931.0

Notes:

Access to Rule Execution Server and Decision Center is handled through the authentication mechanism at the application server level.

WebSphere Application Server provides security infrastructure and mechanisms to protect sensitive Java EE resources and administrative resources and to address enterprise end-to-end security requirements on authentication, resource access control, data integrity, confidentiality, privacy, and secure interoperability.

In WebSphere Application Server, security is organized in layers, from the platform security up to the WebSphere Application Server-specific layer (based on the Java EE model) over the Java security layer.

Java 2 Security provides a policy-based, fine-grain access control mechanism that increases overall system integrity. Java 2 Security is independent of Java EE role-based authorization. Java 2 Security can be disabled and enabled independently of global security. However, on certain application servers when global security is enabled, Java 2 Security is also enabled. Java 2 security can be disabled even though global security is enabled.

**Note**

When you enable Java 2 Security on WebSphere Application Server, you must update the `was.policy` file to give read and write permissions on each directory that contains published RuleDocs. For example:

```
permission java.io.FilePermission "<path to my ruledoc folder>${/}-", "read, write, delete";
```

If you do not update the `was.policy` file, users cannot synchronize RuleDocs to the file system.

For more information about security, see the product documentation.



Configuring secure mode on the application server

- Security roles are defined for each application and the WebSphere Application Server administrative roles, such as the Monitor role
- To activate access control for Decision Server and Decision Center in WebSphere Application Server:
 - Create the users and groups in a user registry setup with the application server
 - Map your users and groups to the Decision Center and Decision Server application security roles

© Copyright IBM Corporation 2015

Figure 3-8. Configuring secure mode on the application server

WB393 / ZB3931.0

Notes:

The only security requirements that are specific to ODM are that the WebSphere Application Server roles map to the mandatory user roles defined by the application.

By default, the Rule Execution Server console does not require security in WebSphere Application Server. However, to activate access control for Rule Execution Server in WebSphere Application Server, you must create users and groups, and map the resAdministrative groups to the Monitor role.

3.2. Configuring security for ODM modules on WebSphere Application Server

Configuring security for ODM modules on WebSphere Application Server



© Copyright IBM Corporation 2015

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

10.1

Figure 3-9. Configuring security for ODM modules on WebSphere Application Server

WB393 / ZB3931.0

Notes:

Application security roles and user-defined groups

- Security is enforced through application security roles and user-defined groups
 - A role defines a set of actions that the user can perform
 - These actions are made available in the consoles according to the rights of the logged-in user
 - Every user must belong to a group
 - Permissions are handled at the group level
- Map existing roles to these predefined roles
 - If you already have groups or users that are defined in your organization that can match these predefined roles, you can map them to the roles during application deployment
- You create and define the groups in the application server
- Roles are also defined in the Java EAR files for these ODM applications:
 - Rule Execution Server console
 - Decision Center Enterprise console
 - Decision Center Business console
 - Decision Server Events

© Copyright IBM Corporation 2015

Figure 3-10. Application security roles and user-defined groups

WB393 / ZB3931.0

Notes:

As you learned in the previous unit, to activate access control for Rule Execution Server and Decision Center in WebSphere Application Server, you first create the users and groups on the application server. Then, you map your groups to the WebSphere Application Server roles.

The only security requirements that are specific to ODM are that the WebSphere Application Server roles map to the mandatory user roles defined by the application. Adherence to these roles determines the parts of the application that a user can access.



Mandatory role definitions to access Rule Execution Server through the console

- Default role definitions to access Rule Execution Server through the console:
 - resAdministrators
 - resDeployers
 - resMonitors
- You must map your users and groups to these mandatory roles when you deploy the Rule Execution Server console
 - All Rule Execution Server users must belong to one of these roles
 - You cannot customize the mandatory roles
- Role definitions are declared in the deployment descriptor of the Rule Execution Server EAR file:
 - Deployment descriptor: WEB-INF/web.xml
 - EAR for WebSphere Application Server V8.5:
<InstallDir>/executionserver/applicationservers/WebSphere85/jrules-res-management-WAS85.ear

© Copyright IBM Corporation 2015

Figure 3-11. Mandatory role definitions to access Rule Execution Server through the console

WB393 / ZB3931.0

Notes:

By default, the Rule Execution Server console does not require security in WebSphere Application Server.

However, to activate access control for Rule Execution Server in WebSphere Application Server, you must create users and groups, and map the resAdministrative groups to the Monitor role to have access to the MBeans of the model.



Rule Execution Server role definitions

- Monitor (read-only access)
 - Users with monitoring rights can view and explore resources, decision services, execution units, and statistics, but cannot modify them
 - They can also select a trace configuration and view and filter trace information in Decision Warehouse
- Deployer
 - In addition to their monitoring rights, users with deploying rights can deploy RuleApp archives and to edit and remove entities (RuleApps, rulesets, decision services, Java XOM resources, and libraries)
 - Can also run diagnostic tests
- Administrator (full access)
 - Users with administrator rights have full control over the deployed resources and access to information on the server
 - Can deploy, browse, and modify RuleApps, Java XOM resources, and libraries
 - Can monitor the decision history, purge, and back up the history
 - Can select a trace configuration, view and filter trace information, and clear trace information in Decision Warehouse
 - Can run diagnostics and view server information

© Copyright IBM Corporation 2015

Figure 3-12. Rule Execution Server role definitions

WB393 / ZB3931.0

Notes:

Mandatory role definitions to access Decision Center through the console

- Default role definitions to access to Decision Center through the Business and Enterprise consoles:
 - rtsAdministrator
 - rtsConfigManager
 - rtsInstaller
 - rtsUser
- You must map your users and groups to these mandatory roles when you deploy the Decision Center console
 - All Decision Center users must belong to one of these roles
- Role definitions are declared in the deployment descriptor of the Decision Center EAR file:
 - Deployment descriptor: WEB-INF/web.xml
 - EAR for WebSphere Application Server V8.5:
<InstallDir>/teamserver/applicationservers/WebSphere85/jrules-teamserver-WAS85.ear

© Copyright IBM Corporation 2015

Figure 3-13. Mandatory role definitions to access Decision Center through the console

WB393 / ZB3931.0

Notes:

 WebSphere Education 

Mandatory role definitions to access Decision Server Events

- Default group definitions to access Decision Server Events:
 - WBEUsers
 - WBEAdmins
 - WBEDashboardUsers
 - WBERestApiUsers
 - WBERestApiPrivilegedUsers
 - WBEHTTPConnectorUsers
- All Decision Server Events users must belong to one of these roles
- Role definitions are declared in the deployment descriptor of the EAR file:
 - Deployment descriptor: WEB-INF/web.xml
 - EAR file: <InstallDir>/runtime/wberuntimeear.ear

© Copyright IBM Corporation 2015

Figure 3-14. Mandatory role definitions to access Decision Server Events

WB393 / ZB3931.0

Notes:

3.3. Configuring single sign-on (SSO)

Configuring single sign-on (SSO)



© Copyright IBM Corporation 2015

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

10.1

Figure 3-15. Configuring single sign-on (SSO)

WB393 / ZB3931.0

Notes:



Single sign-on

- Configure single sign-on at the application server level
 - Specific to the application server version and configuration
 - SSO is offered as an option during installation
- By design, Decision Center is a regular Java EE application and receives user authentication only from the application server
 - If the application server is configured to use an SSO solution, Decision Center correctly relies on the validation, and does not request the application server to present a login screen

© Copyright IBM Corporation 2015

Figure 3-16. Single sign-on

WB393 / ZB3931.0

Notes:

For more information, see the technote:

<http://www.ibm.com/support/docview.wss?uid=swg21400684>

3.4. Defining custom groups for Decision Center

Defining custom groups for Decision Center



© Copyright IBM Corporation 2015

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

10.1

Figure 3-17. Defining custom groups for Decision Center

WB393 / ZB3931.0

Notes:

Defining custom groups for Decision Center (1 of 2)

Every user of Decision Center, including new custom users, must belong to at least one of the mandatory roles:

- **rtsUser**: Rule artifact management
- **rtsConfigManager**: Deployment, advanced management
- **rtsAdministrator**: Security, advanced management
- **rtsInstaller**: Installation

To define custom groups:

1. Declare them in the deployment descriptor of the Decision Center EAR file
2. Map the groups to the users or groups of the user registry that is defined in the application server
3. Map each user, or the group of users, to at least one of these mandatory roles
4. Upload the groups to the database so you can use them to set permissions in Decision Center

© Copyright IBM Corporation 2015

Figure 3-18. Defining custom groups for Decision Center (1 of 2)

WB393 / ZB3931.0

Notes:

You can employ any custom authentication mechanism, if the user (or user's group, as defined in the user registry) is mapped to at least one of the mandatory roles that are defined for the ODM applications.

It is not necessary for the user or group names to be identical to the role names. You can define your own custom user roles or groups if you make sure that they are mapped to the mandatory roles.

When you define custom group, you define it in the deployment descriptor and add it to the application server user registry.

When a user signs in, Decision Center relies on the application server to authenticate the user. This authentication is accomplished through the communication between the server and the user registry, according to the user credentials that are defined in the user registry.



Note

For testing purposes, create a default user and password for each of your custom groups.

Within Decision Center, you must also do the last step that is listed here, which enables you to use the Decision Center permission mechanism and specify fine-grained permissions on individual artifacts within a project. The authenticated user can access only artifacts and operations for which permission is authorized according to their role.

For more information, see the following websites:

- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.config.was/config_dc_websphere/con_dc_was.html
- <http://www.ibm.com/support/docview.wss?rs=0&uid=swg21439999>

Defining custom groups for Decision Center (2 of 2)

- Custom groups must be declared in the deployment descriptor (`web.xml`) in these files that are packaged in the EAR file:
 - Decision Center Enterprise console: `teamserver.war`
 - Decision Center Business console: `decisioncenter.war`
- Add custom groups in the **security-role** section of the `web.xml` file


```
<security-role>
  <role-name>my_custom_group</role-name>
</security-role>
```
- To edit the EAR file, which is compressed:
 - Extract the files that must be changed, edit them in a text editor, and then replace them in the EAR file
 - You can use Ant commands to repackage the EAR file
 - Make a backup of the EAR file before modifying it

© Copyright IBM Corporation 2015

Figure 3-19. Defining custom groups for Decision Center (2 of 2)

WB393 / ZB3931.0

Notes:

For information about using Ant commands to repackage the EAR file, see:

http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.config.was/shared_config_topics/tpc_rts_ant_repackaging.html

3.5. Administering security in Decision Center

Administering security in Decision Center



© Copyright IBM Corporation 2015

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

10.1

Figure 3-20. Administering security in Decision Center

WB393 / ZB3931.0

Notes:

Adding custom groups to Decision Center database

- After creating groups on the application server, you use the **Setup Groups** page in the **Installation Settings** wizard to upload groups to the database

The screenshot shows the 'Step 3: Setup Groups' page of the Installation Settings wizard. On the left is a navigation menu with links to 'Install Home', 'Step 1: Configure Database', 'Step 2: Setup Message File', 'Step 3: Setup Groups' (which is highlighted in blue), 'Step 4: Set Persistence Locale', and 'Step 5: Set Configuration Parameters'. The main content area is titled 'Setup groups' and contains a note: 'To use Decision Center project security as well as its permissions be declared in both the application server and here.' Below the note is a list of checkboxes for group types: 'Groups', 'Eligibility', 'Validator', 'rtsConfigManager', and 'rtsUser'. At the bottom of the list are two buttons: 'New' and 'Delete'.

© Copyright IBM Corporation 2015

Figure 3-21. Adding custom groups to Decision Center database

WB393 / ZB3931.0

Notes:

After creating groups in your application server when you set up security access, you must use the Setup Groups page in the Installation Settings wizard to upload groups to the database.

You do this step only if you want to use the Decision Center project access and permission mechanisms. You must add all the groups that you want to see in the available list when enforcing project security or setting permissions in Decision Center. For more information, see the topics on Groups and Permissions in the Decision Center help documentation.



Note

In Decision Center, the groups are the roles in the application server, *not* the groups that are defined in the user registry. Decision Center uses the group information to verify whether a user belongs to a role in the application server.

You do not have to upload the `rtsAdministrator` or `rtsInstaller` group. The administrator group has access to everything, and an installer user must belong to another group.



Enforcing security on a project

To enforce security:

1. Enforce security on the project branch
2. Specify permissions for the group

Branch Security

Save | Cancel | Help

You are currently editing the security settings for project: miniloan-rules, branch: main

Do not enforce security for this branch
 Enforce and configure security for this branch

Select the groups that can access the branch:

Available groups:
Eligibility
Validator
myTestGroup
rtsConfigManager
rtsUser

Selected groups:
myGroup



© Copyright IBM Corporation 2015

Figure 3-22. Enforcing security on a project

WB393 / ZB3931.0

Notes:

Projects can have several branches that users are working on. The default branch is called "main."

When you enforce security, you can choose which branch the security access applies to.

Security is not enforced on a project until you explicitly enable it. When you enforce security, you must also specify which groups can access the project. Users who do not belong to at least one of the specific groups cannot see that project in the list of projects. As an administrator, you sign in with administrative permissions and can always access all projects.

Fine-grained permissions

- You can specify separate permissions for each operation on each type of artifact in a project
- Permission for the *update* operation can be defined separately for each property of an artifact (such “status”)

Operation	Possible values
View	yes, no, group
Create	yes, no
Update	yes, no, group
Delete	yes, no, group

© Copyright IBM Corporation 2015

Figure 3-23. Fine-grained permissions

WB393 / ZB3931.0

Notes:

You can establish which project elements the users of a specific group can *create*, *update*, *view*, or *delete*. By default, when you enforce security on a project, all permissions are denied.

You must explicitly specify permissions for each group. If you do not specify permissions for a group, the members of the group cannot see the project.

You can define different permissions on each artifact within the project, which allows fine-grained control when necessary.

Permission definitions

Permission	Meaning
Yes	Users can update, view, or delete the project element
No	Users cannot update, view, or delete the project element
Group	Users can update, view, or delete the project element when its Group property corresponds to a group to which the user belongs

© Copyright IBM Corporation 2015

Figure 3-24. Permission definitions

WB393 / ZB3931.0

Notes:

The values for permission settings can be yes, no, or group, with the following meanings:

- Yes: The permission is granted.
- No: The permission is denied.
- Group: The permission is granted (that is, the user can view, create, update, or delete the project element) if the **Group** property of this element corresponds to a group to which the user belongs.

For users that belong to multiple groups, the set of permissions across those groups are called **effective permissions**.



Specifying fine-grained permissions

- The following example shows permissions for a fictitious group named **Eligibility**

You are currently editing the permissions for the group: **Eligibility**

Actions	PERMISSION	TYPE	PROPERTY
<input type="checkbox"/>	View	*	-

Permission Value Type Property

Create	Yes	Action Rule	-
--------	-----	-------------	---

Apply **Cancel**

- The Eligibility group has permission to create **action rules** because:
 - Permission** is set to **Create**
 - Value** is set to **Yes**
 - Element Type** is set to **Action Rule**

© Copyright IBM Corporation 2015

Figure 3-25. Specifying fine-grained permissions

WB393 / ZB3931.0

Notes:

Permissions in action

- When a user signs in, Decision Center evaluates what the user can do based on the permissions that are granted to the group or groups to which the user belongs:
 - To which of the basic groups (`rtsUser`, `rtsConfigManager`, `rtsInstaller`, `rtsAdministrator`) does the user belong?
 - To which "other" groups, as customized for your enterprise, does the user belong?
 - What permissions are set for that group?

© Copyright IBM Corporation 2015

Figure 3-26. Permissions in action

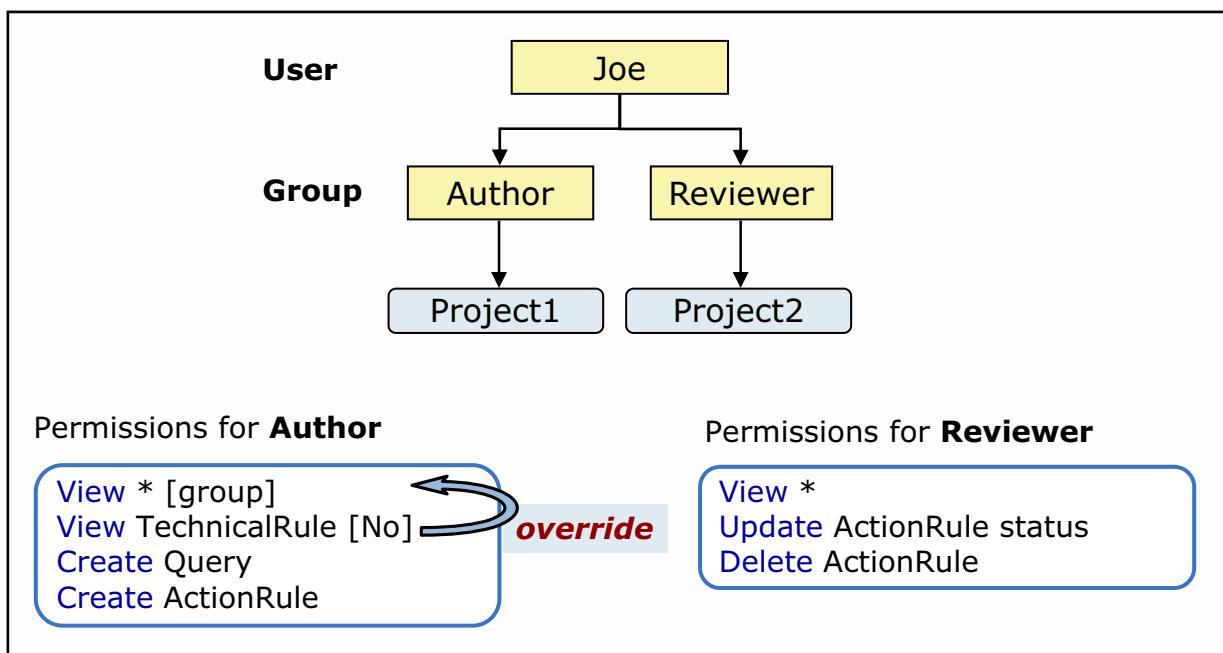
WB393 / ZB3931.0

Notes:

When a user logs in, Decision Center evaluates what the user can do based on the permissions that are granted to the group or groups to which the user belongs.

- To which of the three basic groups (`rtsUser`, `rtsConfigManager`, and `rtsAdministrator`) does the user belong? The user group determines which tabs of Decision Center are available.
By default, the `rtsAdministrator` group has full permissions.
- To which "other" groups, as customized for your enterprise, does the user belong? Create customized user groups to establish project-specific permissions.
- What permissions are set for that group to create, update, view, or delete specific types of project elements (smart folders, folders, rules) and their properties (status, name, definition)?

Permissions in action: Example



© Copyright IBM Corporation 2015

Figure 3-27. Permissions in action: Example

WB393 / ZB3931.0

Notes:

Consider this example of how permissions can be used to control access.

- Joe is a user who has different roles on different projects: for project 1, Joe is an **Author**, and for project 2, Joe is a **Reviewer**.
- The roles are reflected as groups in Decision Center.
- For the **Author** group, the permissions are as follows:
 - View * [group]: Only people of the group can see everything.
 - View TechnicalRule [No]: Joe cannot see technical rules.
 Because of the order, this permission overrides the previous one.
 - Create Query: Joe can create queries.
 - Create ActionRule: Joe can create action rules.
- As a member of the **Reviewer** group, Joe can:
 - View all rule artifacts from any group

- Update the status of action rules
- Delete action rules
- If Joe is both an **Author** and a **Reviewer** for the same project, the role that grants the more permissions would prevail.



Example

Reviewer permissions allow Joe to view technical rules, although the **Author** permissions do not.

3.6. Troubleshooting security in Decision Center

Troubleshooting security in Decision Center



© Copyright IBM Corporation 2015

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

10.1

Figure 3-28. Troubleshooting security in Decision Center

WB393 / ZB3931.0

Notes:



Issue: Authorization failed

- Error messages:
 - Authorization failed for user
Authorization failed, Not granted any of the required roles:
resAdministrators resDeployers
 - This user does not have the correct role
 - Invalid username/password
 - Your username and password were not recognized
- Diagnosing the problem
 - Find the error in the server logs and verify the security settings of the application at the application server level
- Resolving the problem
 - Make sure to map all the groups or users who are granted access to the module to one of the mandatory roles, as indicated in the installation guide for the corresponding module
 - Make sure that the user name exists in the user registry setup for the application server
 - Make sure that the password defined for that user in the same registry is the same password that was entered

© Copyright IBM Corporation 2015

Figure 3-29. Issue: Authorization failed

WB393 / ZB3931.0

Notes:

For more information, see this technote:

<http://www.ibm.com/support/docview.wss?uid=swg21578192>



Issue: Incorrect permissions

- Error
 - Users log in to the Decision Center consoles, but are unexpectedly denied access to a project or do not have the correct permissions for the project
- Diagnosing the problem
 - Verify that the user/group is mapped to a role that is assigned permission for that project
- Resolving the issue:
 - Verify that the user/group that was used to sign in is properly mapped
 - If the user/group is mapped to a custom group, make sure that the custom group that is in the Decision Center descriptors matches the group that is listed in the Installation Settings wizard
 - Make sure that the correct group was used to define the permissions for that project

© Copyright IBM Corporation 2015

Figure 3-30. Issue: Incorrect permissions

WB393 / ZB3931.0

Notes:



Troubleshooting support

- For more information about resolving security issues, see IBM Support

© Copyright IBM Corporation 2015

Figure 3-31. Troubleshooting support

WB393 / ZB3931.0

Notes:

For more examples of security issues and resolutions, see these technotes:

- <http://www.ibm.com/support/docview.wss?uid=swg21456170>
- <http://www.ibm.com/support/docview.wss?uid=swg21584856>
- <http://www.ibm.com/support/docview.wss?uid=swg21461650>



Unit summary

Having completed this unit, you should be able to:

- Customize user access
- Administer security in Decision Center

© Copyright IBM Corporation 2015

Figure 3-32. Unit summary

WB393 / ZB3931.0

Notes:

Checkpoint questions

1. **True or False:** To customize security for Decision Center, you must modify the deployment descriptor, plus add custom groups to the Decision Center database.
2. **True or False:** To set permissions in Decision Center, you must be signed in with the rtsUser role.
3. **True or False:** To enforce security on a project in Decision Center, you can specify view, create, update, and delete permissions for groups.
4. **True or False:** When you sign in to Decision Center, your effective permissions are based on the order in which your permissions are listed. The least restrictive permissions apply.

© Copyright IBM Corporation 2015

Figure 3-33. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

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



Checkpoint answers

1. **True.**
2. **False.** *You must be signed in with administrative privileges (rtsAdmin).*
3. **True.**
4. **True.**

© Copyright IBM Corporation 2015

Figure 3-34. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 5



Customizing user access and
enforcing security in Decision Center

© Copyright IBM Corporation 2015

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

10.1

Figure 3-35. Exercise 5

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Set up custom groups and user roles
- Enforce security in Decision Center by specifying fine-grained permissions on Decision Center projects

© Copyright IBM Corporation 2015

Figure 3-36. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 4. Synchronizing across environments

What this unit is about

In this unit, you learn how to ensure that project artifacts are defined consistently for all phases of the project and can be shared across environments.

What you should be able to do

After completing this unit, you should be able to:

- Describe the synchronization mechanisms
- Determine which repository to use as the master source

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Describe the synchronization mechanisms
- Determine which repository to use as the master source

© Copyright IBM Corporation 2015

Figure 4-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Collaboration between business and technical users
- Choosing the master source
- Understanding the synchronization mechanism
- What to synchronize
- Importing and exporting projects with Decision Center

© Copyright IBM Corporation 2015

Figure 4-2. Topics

WB393 / ZB3931.0

Notes:

4.1. Collaboration between business and technical users

Collaboration between business and technical users



© Copyright IBM Corporation 2015

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

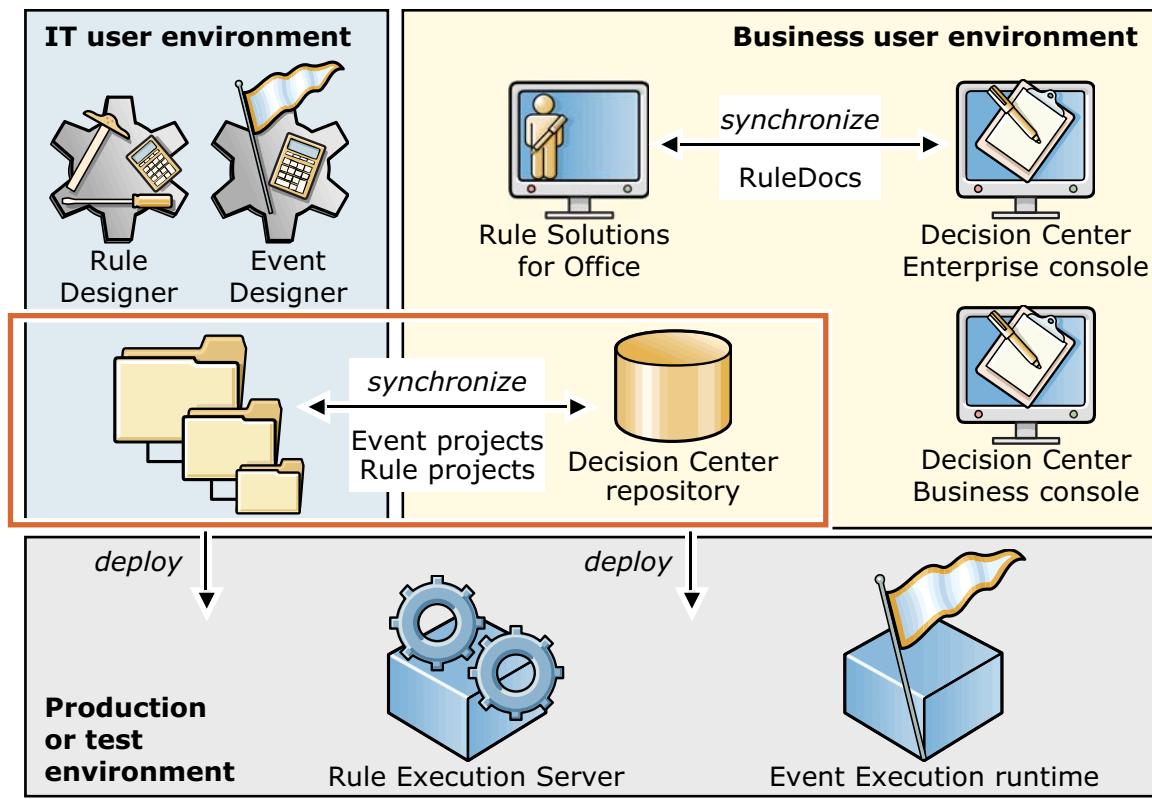
10.1

WB393 / ZB3931.0

Figure 4-3. Collaboration between business and technical users

Notes:

Simultaneous development across environments



© Copyright IBM Corporation 2015

Figure 4-4. Simultaneous development across environments

WB393 / ZB3931.0

Notes:

Synchronization is the key to collaboration between business and IT users who work on the same projects but in separate environments.

Decision Center is a rule authoring and management environment that is dedicated to business users. Business users store rule and event projects in a database, called the Decision Center repository, and access them through the Decision Center Business and Enterprise consoles. The Decision Center repository handles multi-user concurrency and version control.

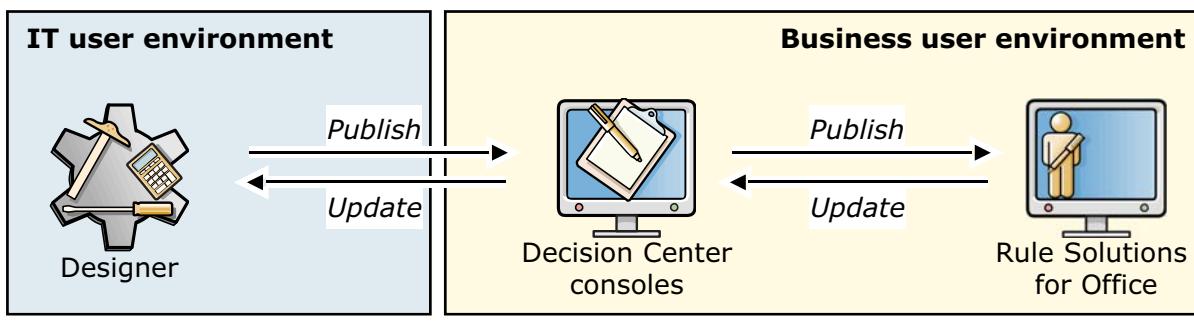
Developers store rules in a file system, and access them with Rule Designer and Event Designer. Developers work on copies of a rule project or event project in their Eclipse workspace, and keep a master copy in a source code control (SCC) system that handles file sharing, conflict resolution, and version management.

To work in collaboration with IT users, the Decision Center repository must be synchronized with the development tools in Designer.

Synchronization between users

- Synchronization across business and development environments
 - Initiated and controlled from Designer
 - Compares the versions of every project element that are stored in the Decision Center repository with the copy stored through Designer

- Synchronization between business users
 - Initiated and controlled from Decision Center
 - Compares rule artifacts that are stored in the Decision Center repository with the rules that are edited in the RuleDocs



© Copyright IBM Corporation 2015

Figure 4-5. Synchronization between users

WB393 / ZB3931.0

Notes:

Synchronization between Rule Designer and Decision Center involves *publishing* and *update* operations that are controlled from Rule Designer.

- You publish rule projects from Rule Designer to Decision Center.
- After the rule project evolves in Decision Center, you can update its contents back into Rule Designer.

Synchronization between Decision Center and Rule Solutions for Office follow the same principles. Working with rules in Microsoft Office 2007 documents starts by publishing your rules as RuleDocs from Decision Center. Then, to maintain synchronization between Decision Center and your RuleDocs after the initial publication, you must:

- Publish changes from Decision Center to the RuleDocs
- Update changes in the RuleDocs back into Decision Center

**Important**

There is no direct synchronization between Rule Designer and Rule Solutions for Office. When rule artifacts are modified with Rule Solutions for Office, business users must first update the rule projects in Decision Center before you can update the rule projects in Rule Designer.



Synchronization workflow

- Projects are originated in Designer by developers, and then ***published*** to Decision Center so that business users can access them
 - Decision Center then becomes the copy of record for rules
 - Control of the business object model remains with the rule project in Designer, and developers must maintain it
- To implement BOM changes that affect the rule vocabulary, a technical user typically:
 1. Updates the Designer rule projects with the latest version of the rule projects from Decision Center
 2. Implements the BOM changes, which usually leads to *refactoring* of the rules
 3. Publishes the modified BOM and refactored rules to Decision Center

© Copyright IBM Corporation 2015

Figure 4-6. Synchronization workflow

WB393 / ZB3931.0

Notes:

After the project is available in the Decision Center repository, business users can add or edit rules in the project. Developers can also continue working on the same project, for example, to complete the business object model (BOM) that supports the rule vocabulary.

To keep the developer view of the project in Designer synchronized with the work of the business users, developers periodically update the rule project from Decision Center. Changes to the BOM are likewise published to Decision Center so that business users can work with the latest vocabulary.

In Decision Center, a rule project corresponds to a specific organization of rules and other project elements, and is stored in a *rule repository*, that is, a database for your rules.

The Decision Center repository can contain many rule projects, but you work with one rule project at a time in Decision Center.

Depending on your organization, the Decision Center repository might be considered the *source of truth* for business rules. “Source of truth” means that the other copies of the rule project that are outside of the repository are considered to be temporary copies. When you provide support to

business users by updating the project with new vocabulary, complex rules, or templates, you start with the latest version of the project in Decision Center repository.

**Important**

The Decision Center repository does not contain the artifacts that are required for rule execution, like XOMs, .jar files, and libraries.

Synchronizing between Decision Center and Rule Solutions for Office

- Similar principles apply between Decision Center and Rule Solutions for Office
- Rules are published from Decision Center to the file system as RuleDocs
- After changes are made, you synchronize the Decision Center projects with the updated rules in the RuleDocs
 - RuleDocs are considered temporary copies
 - The Decision Center repository is considered the master copy for business users
- There is no direct synchronization between Rule Solutions for Office (RuleDocs) and Rule Designer

© Copyright IBM Corporation 2015

Figure 4-7. Synchronizing between Decision Center and Rule Solutions for Office

WB393 / ZB3931.0

Notes:

Rule Solutions for Office is a Microsoft Office plug-in that allows business users to manipulate and edit the artifacts rule projects as *RuleDocs* in Word or Excel.

RuleDocs are Office documents in which business users can present the rules together with documentation, supporting information, and even the original policy.

- Rules and ruleflows in a project are published as Word documents. The rule editor that is integrated with Word supports the writing of business rules with a natural language syntax, while ensuring correct use of business vocabulary and compliance with business rule syntax.
- Decision tables in a rule project are published as Excel documents. The decision table editor that is integrated into Excel provides a convenient way to visualize and edit sets of similar business rules.
- You can also configure the Enterprise console to create and edit RuleDocs directly in Rule Solutions for Office:
 - Open the Installation Settings Wizard on the **Configuration** tab.
 - Set the configuration parameter `teamserver.rsoediting.enable` to true.

To synchronize the Decision Center projects and the RuleDocs, from Decision Center:

- Publish any changes that are made in Decision Center to the RuleDocs.
- Update any changes that are made to the RuleDocs back into Decision Center.



Important

As for Decision Center repository, RuleDocs do not contain the artifacts that are required for rule execution.

Rule Solutions for Office is not covered or demonstrated during this course.

4.2. Choosing the master source

Choosing the master source



© Copyright IBM Corporation 2015

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

10.1

Figure 4-8. Choosing the master source

WB393 / ZB3931.0

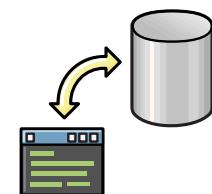
Notes:

Choosing the master source

- Establish which source to use as the master:
 - Decision Center repository
 - Source code control (SCC)

- Business-user-centric approach:
 - Decision Center repository is considered master
 - Repository provides rule management features, such as version control, baselines, and access control
 - Copies that are stored through Designer are considered temporary

- Developer-centric approach:
 - Projects that are stored and managed through SCC
 - A technical user synchronizes the source with Decision Center repository
 - Execution-related artifacts are stored with projects



© Copyright IBM Corporation 2015

Figure 4-9. Choosing the master source

WB393 / ZB3931.0

Notes:

To avoid conflicts when rule projects are shared between business users and developers, you must establish clearly which source is the master: Decision Center or source code control (SCC).

Technical users synchronize their rule projects through SCC, and business users use Decision Center to synchronize their work.

A **developer-centric approach** is better adapted for an initial development situation. Technical users synchronize their rule projects through SCC. One dedicated technical user publishes rule projects to Decision Center for testing, and synchronizes any changes between the two repositories.

A **business user-centric approach** is better adapted for the later stages of project development, when the business object model stabilizes. You consider the Decision Center repository to be the source of truth and any Designer developer copy in SCC as a temporary copy of the project.

Dedicated synchronization manager

- Have one dedicated technical user synchronize the two repositories
- Suggested practice: One person + one workspace to control synchronization of a set of rule projects
- All rule project changes should be channeled through the synchronization manager
 - If others are involved in updates, make sure that a single technical user coordinates the changes

© Copyright IBM Corporation 2015

Figure 4-10. Dedicated synchronization manager

WB393 / ZB3931.0

Notes:

One dedicated technical user should handle synchronizing the two repositories. As a suggested practice, one person should use one workspace to manage synchronization of a set of rule projects. This approach can help avoid synchronization errors that might occur when one person uses several workspaces or several users share a workspace.

If another technical person needs to publish changes to Decision Center, follow these steps:

1. Working in a clean workspace, retrieve current projects from Decision Center by selecting **Create new Rule Project from Decision Center**
2. Make the necessary changes to the projects in Designer
3. Publish the changes back to Decision Center



Warning

Because the involvement of another user is not a suggested practice, limit the duration of this type of task. It should not become an ongoing separate thread of publishing and updating.

4.3. Understanding the synchronization mechanism

Understanding the synchronization mechanism



© Copyright IBM Corporation 2015

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

10.1

Figure 4-11. Understanding the synchronization mechanism

WB393 / ZB3931.0

Notes:



Synchronization tools

- Publish an existing rule or event project from Designer to Decision Center
- Create a project in Designer from an existing Decision Center project
- Synchronize the Designer and Decision Center copies of the project to take into account changes on either side
- Resolve conflicts
 - When the same artifact is modified in both environments, compare differences and choose which version to keep

© Copyright IBM Corporation 2015

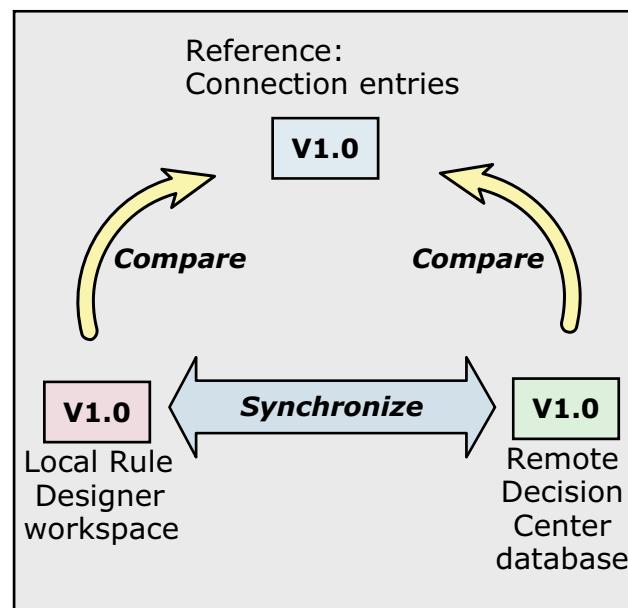
Figure 4-12. Synchronization tools

WB393 / ZB3931.0

Notes:

Synchronization architecture

- Synchronization uses three-way comparison of:
 - Project in local workspace of Designer
 - Remote project in Decision Center repository
 - Reference state, also known as common ancestor
- Reference state is created as a connection entry file in workspace when you connect to Decision Center:
 - Connection entries file: .syncEntries
- Three-way comparison creates a checksum on both remote and local rules, and then compares them to the reference state



© Copyright IBM Corporation 2015

Figure 4-13. Synchronization architecture

WB393 / ZB3931.0

Notes:

The screenshot shows the WebSphere Education interface. At the top left is the 'WebSphere Education' logo, and at the top right is the 'IBM' logo. Below the header, the title 'Connection entries' is displayed in blue. To the right of the title is a large window titled 'Decision Center Configuration'. This window contains fields for 'URL' (http://localhost:9080/teamserver), 'User name' (rtsAdmin), and 'Password' (*****). It also includes a 'Data source' dropdown and a 'Connect' button. Below these fields is a 'Synchronization Filter' section with a checkbox for 'Use the specified query to filter rule elements for synchronization'. Under 'Project configuration', there are two radio buttons: 'Create a new project on Decision Center' (selected) and 'Synchronize with the existing Decision Center project'. A 'Project' dropdown and a 'Select a branch or release' dropdown are also present. At the bottom right of the dialog are 'Finish' and 'Cancel' buttons. Copyright information '© Copyright IBM Corporation 2015' is visible at the bottom right. At the very bottom of the screen, the Eclipse IDE toolbar is visible, with the 'Team Synchronizing perspective' tab highlighted.

Figure 4-14. Connection entries

WB393 / ZB3931.0

Notes:



Side-by-side comparison

- Text Compare view shows changes side by side

```

<?xml version="1.0" encoding="UTF-8"?>
<model.brl:ActionRule xmi:version="2.0">
  <name>checkIncome</name>
  <uuid>f2a66955-ee5f-46e6-973c-b8e28cd
  <locale>en_US</locale>
  <definition><![CDATA[<definitions>
    <rule>
      <id>1</id>
      <name>Check Income</name>
      <description>This rule checks if the minimum income is less than 0.37 times the maximum income.</description>
      <condition>
        <script>
          <language>JavaScript</language>
          <expression>
            <script>
              if (minIncome < 0.37 * maxIncome) {
                // ...
              }
            </script>
          </expression>
        </script>
      </condition>
      <actions>
        <script>
          <language>JavaScript</language>
          <expression>
            <script>
              if (minIncome < 0.37 * maxIncome) {
                // ...
              }
            </script>
          </expression>
        </script>
      </actions>
    </rule>
  </definitions></![CDATA]>
  </definition>

```

© Copyright IBM Corporation 2015

Figure 4-15. Side-by-side comparison

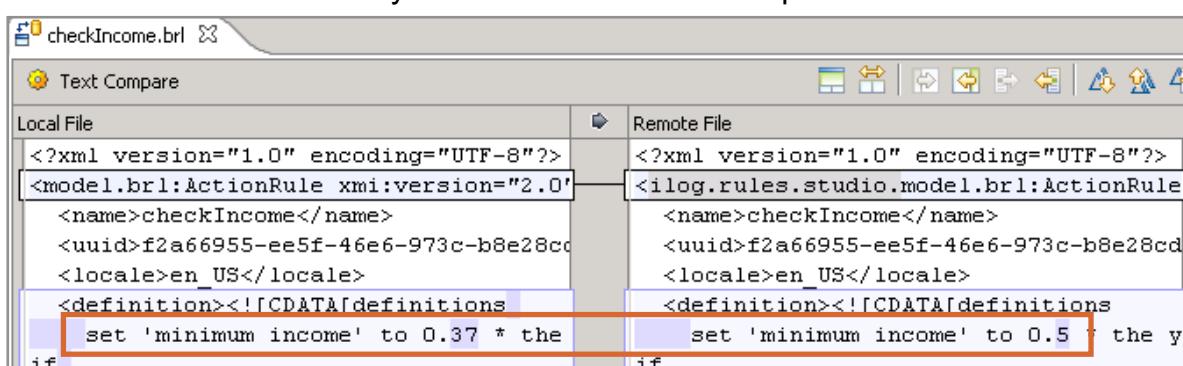
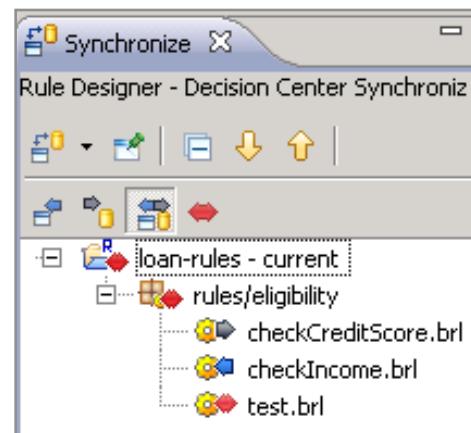
WB393 / ZB3931.0

Notes:

When the Synchronize view shows a difference between the local Designer and remote Decision Center versions of a rule, you can double-click the entry to display both rules and their differences in the Text Compare Editor.

Choosing how to synchronize

- Synchronize view lists changes and direction:
 - Outgoing
 - Incoming
 - Conflict
- Text Compare view
 - Detailed differences so you can determine how to update each artifact



© Copyright IBM Corporation 2015

Figure 4-16. Choosing how to synchronize

WB393 / ZB3931.0

Notes:

After you check the contents of the local and remote versions, you can update, publish, or override to resolve any conflicts.

UUIDs

- Decision Center uses Universal Unique Identifiers (UUIDs) to identify rules in a unique, unambiguous way
- Synchronization uses the UUID when establishing whether rules in Designer and Decision Center are synchronized

Same name and same UUID

<name>checkCreditScore</name>	<name>checkCreditScore</name>
<uuid>afb53b16-eb8e-4d39-a1d3-da3342e07328	<uuid>afb53b16-eb8e-4d39-a1d3-da3342e0
<locale>en_US</locale>	<locale>en_US</locale>
<definition><![CDATA[definitions set 'minimum score' to 250 ;	<definition><![CDATA[definitions set 'minimum score' to 200 ;

- Manually changing the UUID can cause conflicts

Same name but different UUID shows as a conflict

 test.brl	<name>test</name>	<name>test</name>
	<uuid>ea3fe847-ed55-4b90-84c8-21c61f61bd9	<uuid>67c35412-5c24-4f76-bc5f-c77c2632

© Copyright IBM Corporation 2015

Figure 4-17. UUIDs

WB393 / ZB3931.0

Notes:



Synchronization commands

- **Publish** sends artifacts from Designer to Decision Center
- **Update** receives artifacts from Decision Center into Designer
- **Override and Publish** resolves conflict by replacing the Decision Center artifact with the Designer version
- **Override and Update** resolves conflict by replacing the Designer artifact with the Decision Center version

© Copyright IBM Corporation 2015

Figure 4-18. Synchronization commands

WB393 / ZB3931.0

Notes:

4.4. What to synchronize

What to synchronize



© Copyright IBM Corporation 2015

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

10.1

Figure 4-19. What to synchronize

WB393 / ZB3931.0

Notes:

Rule project items (1 of 2)

- In Rule Designer, a project is composed of a series of files and folders
 - Manage versions of Designer artifacts through the source code control (SCC) tool of your choice
 - Some SCC systems also use the Team Synchronization feature of Eclipse, which you access through the **Team > Share** menu
- All artifacts that are used for rule authoring are synchronized with Decision Center
- Execution-related artifacts are managed through SCC, except for dynamic XOMs (XML schemas)
- Do not commit these resources to SCC or synchronize them with Decision Center:
 - The `output` folder, which contains compiled files that are generated during project builds
 - The `.syncEntries` file, which is used to share the synchronization state between Designer and Decision Center

© Copyright IBM Corporation 2015

Figure 4-20. Rule project items (1 of 2)

WB393 / ZB3931.0

Notes:

When you create a rule project in Rule Designer, several folders and files are created as part of the project. The artifacts in the rule projects, including the ones that are required for rule execution, have a file-based representation. Not all of these items must be synchronized with Decision Center or managed through SCC.

All artifacts that are related to rule authoring artifacts are synchronized with Decision Center. However, execution-related elements of the rule project, such as Java XOMs, JAR files, and other referenced libraries are not synchronized with Decision Center. These items must be managed through the SCC system.

Managing **all** files of a rule project in an SCC tool is a good practice. However, you do not have to track the following files, which are transient:

- The `output` folder and its contents, which are updated each time that you build the project
- The connection entries (`.syncEntries`) files, which are created at synchronization time



Rule project items (2 of 2)

- These rule artifacts are stored as XML files with the file extensions that are listed here
- When you synchronize these artifacts, you can open the Text Compare view in Designer to compare the XML or modified text format

File or folder	File extension
Rule project	.ruleproject
Rule packages	.rulepackage
Action rules	.brl
Decision tables	.dta
Decision trees	.dtr
Functions	.fct
Ruleflows	.rfl
Technical rules	.trl
Variable sets	.var
Action rule templates	.brt
Queries	.qry
BOM entries	.bom _<locale>.voc .b2x

© Copyright IBM Corporation 2015

Figure 4-21. Rule project items (2 of 2)

WB393 / ZB3931.0

Notes:

WebSphere Operational Decision Management implements the different types of elements in a rule project in Rule Designer as folders, files, or both, as follows:

- **Rule project:** The root folder with a `.project` XML file to store the general Eclipse project information. It also has a `.ruleproject` XML file to store the project properties that are specific to a rule project (such as categories, BOM path, and XOM path).
- **Rule package:** A rule package folder (called a *folder* in Decision Center), and a `.rulepackage` file that stores the information about the rule package.
- **Action rule:** A `.brl` file that stores the properties and the definition of an action rule.
- **Decision table:** A `.dta` file that stores the properties and the definition of a decision table.
- **Decision tree:** A `.dtr` file that stores the properties and the definition of a decision tree.
- **Function:** A `.fct` file that stores the properties and the definition of a function.
- **Ruleflow:** A `.rfl` file that stores the properties, the task definitions, and the description of a ruleflow.
- **Technical rule:** A `.trl` file that stores the properties and the definition of a technical rule.

- **Variable set:** A `.var` file that stores a list of variables.
- **Action rule template:** A `.brt` file that stores the action rule template properties.
- **Query:** A `.qry` file that stores the properties and the definition of a query.
- **BOM entry:** A `.bam` file that stores the structure of a BOM entry, a `_<locale>.voc` file that stores the verbalization information that is attached to BOM elements. It also has a `.b2x` XML file that stores the functions that map the BOM to the XOM.

A rule project in Rule Designer also contains the following files and folders, which are not considered to be items of the rule projects as such:

- **source** folder: The container for the rule artifacts.
- **bam** folder: The container for the BOM entries; all the BOM entries that are directly under the `bam` folder or its subfolders are part of the BOM.
- **queries** folder: The container for the queries that can be used in the project.
- **resources** folder: The container for files that are not part of the rule model.
- **templates** folder: The container for the templates that can be used in the project and any dependent projects.
- **output** folder: The container for compiled files that are generated when you build the project.
- **connection entries:** The `.syncEntries` files that are used to share the synchronization state between Designer and Decision Center.

For the execution-related elements of a rule project, Rule Designer or the SCC tool is the "source of truth."

4.5. Importing and exporting projects with Decision Center

Importing and exporting projects with Decision Center



© Copyright IBM Corporation 2015

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

10.1

Figure 4-22. Importing and exporting projects with Decision Center

WB393 / ZB3931.0

Notes:

Export and import tools

- Decision Center provides export and import tools to support exchange of projects as compressed files
 - Export projects from Decision Center
 - Import projects into Decision Center or Designer

Administration

[Installation Settings Wizard](#)

Modify an existing installation of Decision Center

[Diagnostics](#)

Run diagnostics to check the Decision Center system

[Clean Decision Center Cache](#)

Cleans the cache generated by the ruleset generation

[Import Projects](#)

Import a .zip file containing one or more projects

[Export Current Project State](#)

Export and download the current project for the selected branch or baseline

[Erase Current Project](#)

Erase the current project, its branches, and its history. This operation cannot be undone

© Copyright IBM Corporation 2015

Figure 4-23. Export and import tools

WB393 / ZB3931.0

Notes:

As a Decision Center administrator, you can export a project from Decision Center to be imported into either Designer or another instance of Decision Center. You cannot merge an imported project with an existing one.

Exporting projects

- Export a project for the following use cases:
 - Working with multiple instances of Decision Center
 - Troubleshooting a project from Decision Center in Designer
- The exported project contains only one version of each artifact
 - The history of changes to artifacts is not included in the export
- Dynamic XOMs (XML schemas) are included in the export
 - Other execution-related artifacts are not included

© Copyright IBM Corporation 2015

Figure 4-24. Exporting projects

WB393 / ZB3931.0

Notes:

You can export a project for the following use cases:

- When working with multiple instances of Decision Center for different lifecycle environments (development, test, production).
- When you must troubleshoot a project from Decision Center in Designer and want to take the entire project rather than synchronize from Designer.



Reminder

Decision Center repository does not contain the execution-related elements of a rule project, except for dynamic XOMs (XML schemas). When you create the rule project in Rule Designer from Decision Center, you must also import the execution-related elements, such as the Java XOM, into your workspace.



Importing projects into Decision Center

- When you import projects into Decision Center, the compressed file can contain one or more projects
- If the project that you import exists in Decision Center, you can choose to synchronize or override changes

Administration

[Installation Settings Wizard](#)
Modify an existing installation of Decision Center

[Diagnostics](#)
Run diagnostics to check the Decision Center system

[Clean Decision Center Cache](#)
Cleans the cache generated by the ruleset generation

[Import Projects](#)
Import a .zip file containing one or more projects

[Export Current Project State](#)
Export and download the current project for the selected branch or baseline

[Erase Current Project](#)
Erase the current project, its branches, and its history. This operation cannot be undone

Tip: If the project exists, consider erasing the project *first* before importing, instead of synchronizing.

© Copyright IBM Corporation 2015

Figure 4-25. Importing projects into Decision Center

WB393 / ZB3931.0

Notes:

You can create a rule project in Decision Center from a project that exists in another instance of Decision Center. You can also import projects into Decision Center from a .zip file that contains rule projects that are either exported from Rule Designer or from Decision Center.

When you import projects from a compressed file, the compressed file can contain one or more projects.

If the project that you import exists in Decision Center, you are notified and can then initiate synchronization.

During this synchronization, Decision Center handles differences the following way:

- New elements that are found in the .zip file are added to the Decision Center version of the project.
- Deleted elements are not synchronized. If you deleted an element in Decision Center and not in the .zip version, the element is added to the Decision Center version. Similarly, if an element is deleted in the .zip version, it is not removed from the Decision Center version.

- Changes that are made to the elements in the **.zip** version override elements in Decision Center if you select the **Also override changes in Decision Center** option. If this option is not selected, the Decision Center version remains intact.

Importing projects into Designer

- When you create a rule project in Rule Designer from Decision Center:
 - Only rule project artifacts that are used for authoring are included in the import
 - The project shows errors in Rule Designer if the execution-related elements of the rule project are not already in the workspace
 - Make sure that you retrieve the execution-related elements of the rule project from the SCC tool
- You cannot copy a project or a branch of a project into a Rule Designer workspace that already contains the same project

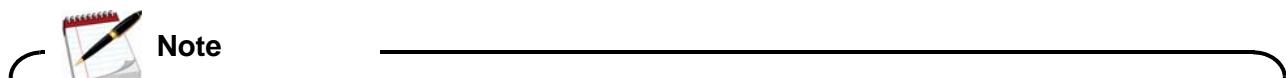
© Copyright IBM Corporation 2015

Figure 4-26. Importing projects into Designer

WB393 / ZB3931.0

Notes:

You can create a rule project in Rule Designer from a rule project in Decision Center. For your new rule project in Rule Designer to build, you must also retrieve any other projects or files that this rule project uses. You can find other referenced rule projects in Decision Center repository, and create corresponding rule projects in Rule Designer. Apart from dynamic XOMs, you cannot retrieve execution elements (like Java XOMs, JAR files, or libraries) that are not stored in Decision Center. You must retrieve these executable elements by other means.



You do not need the administrator rights to copy a project from Decision Center to Rule Designer. However, you must have View permissions on the project elements in Decision Center to retrieve them from Rule Designer.



Unit summary

Having completed this unit, you should be able to:

- Describe the synchronization mechanisms
- Determine which repository to use as the master source

© Copyright IBM Corporation 2015

Figure 4-27. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

- 1. True or False:** Synchronization between Decision Center and Designer is initiated and controlled from Decision Center Enterprise console.

- 2. True or False:** Synchronization uses three-way comparison of the Designer project, Decision Center repository, and the `.syncEntries` file.

© Copyright IBM Corporation 2015

Figure 4-28. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

1.

2.



Checkpoint answers

1. **False.** *Synchronization is initiated and controlled from Designer.*
2. **True.**

© Copyright IBM Corporation 2015

Figure 4-29. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 6



Synchronizing across business and development environments

© Copyright IBM Corporation 2015

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

10.1

Figure 4-30. Exercise 6

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Synchronize rule projects between Rule Designer and Decision Center
- Create a rule project in Rule Designer from a project in Decision Center
- Export a project from Decision Center for import to Rule Designer or other instances of Decision Center
- Erase a project from Decision Center

© Copyright IBM Corporation 2015

Figure 4-31. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 5. Managing deployment

What this unit is about

This unit teaches you how to manage the deployment of rule artifacts to Rule Execution Server for their managed execution. The unit also covers XOM management.

What you should be able to do

After completing this unit, you should be able to:

- Explain deployment principles and options
- Package rulesets as RuleApps for deployment
- Manage RuleApp deployment from Rule Designer and Decision Center
- Describe the basic aspects of XOM management

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Explain deployment principles and options
- Package rulesets as RuleApps for deployment
- Manage RuleApp deployment from Rule Designer and Decision Center
- Describe the basic aspects of XOM management

© Copyright IBM Corporation 2015

Figure 5-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Deployment principles
- Deployment options
- RuleApp and ruleset properties
- Version policies for deployment
- Managing XOMs

© Copyright IBM Corporation 2015

Figure 5-2. Topics

WB393 / ZB3931.0

Notes:

Recall: Administrator tasks

- Responsibilities:
 - Deploying and configuring the server and database for Decision Center and Rule Execution Server
 - Managing user access to Decision Center and Rule Execution Server
 - Configuring trace data sources for testing purposes
 - Deploying applications
 - Redeploying rulesets and event assets as changes are made
 - Generating detailed execution reports
 - Tracking and monitoring rule execution
 - Restoring a particular application state
- Tools: Servers for Decision Center or runtime environments



Administrator

© Copyright IBM Corporation 2015

Figure 5-3. Recall: Administrator tasks

WB393 / ZB3931.0

Notes:

At the beginning of the course, you saw how to deploy the ODM modules to the application server. This unit describes the deployment of business rules from ODM modules to make decision services available to client applications.

5.1. Deployment principles

Deployment principles



© Copyright IBM Corporation 2015

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

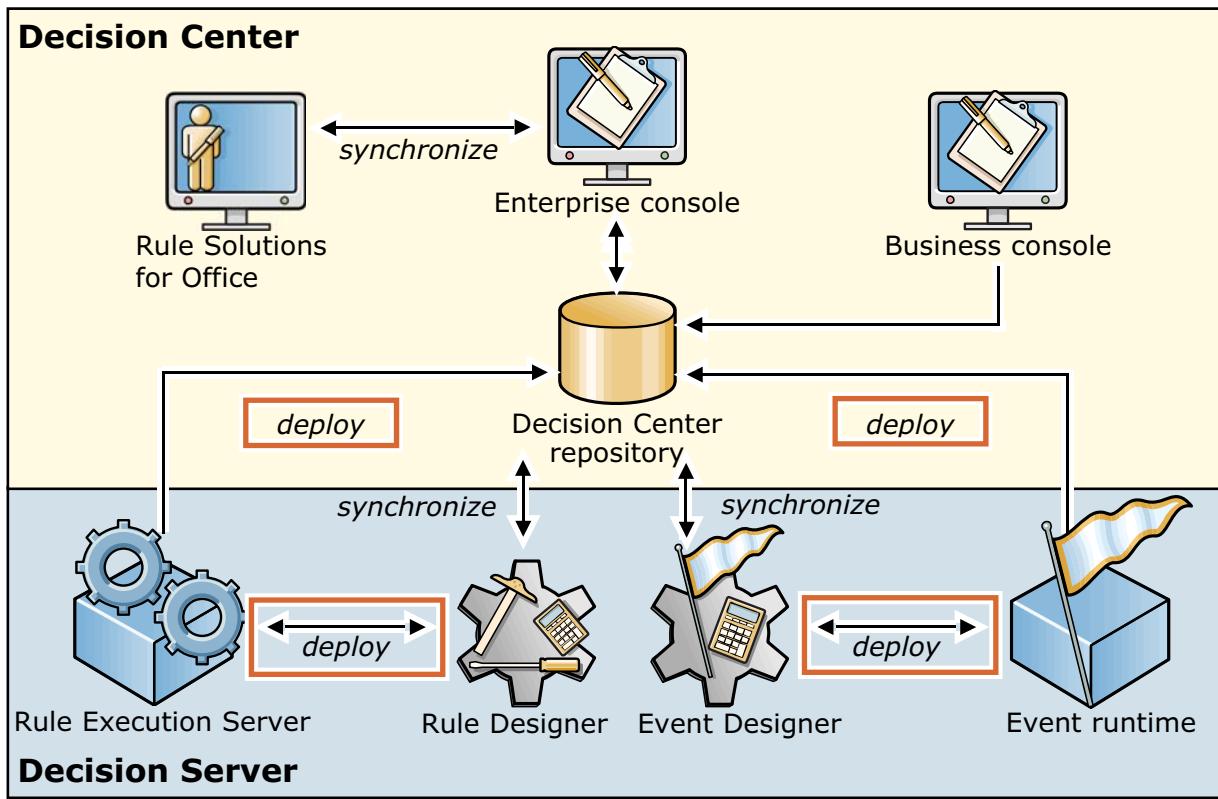
10.1

Figure 5-4. Deployment principles

WB393 / ZB3931.0

Notes:

Recall: ODM interaction



© Copyright IBM Corporation 2015

Figure 5-5. Recall: ODM interaction

WB393 / ZB3931.0

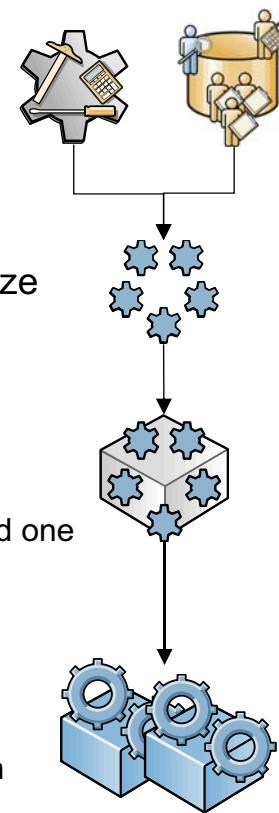
Notes:

To make rules available as a decision service to an application, the rules must be deployed to a rule engine that can execute those rules and return the decision. Rule Execution Server provides a managed environment for rule execution, either in a test environment or in production.

You can deploy to a Rule Execution Server from Rule Designer or Decision Center. Deployment tools are also available in the Rule Execution Server console.

Preparing for deployment

- Prerequisites:
 - Rulesets are defined
 - Target architecture is known
- Business users and developers author rules and finalize *rulesets*
 - Rulesets: The set of rules that produces a decision
- Rulesets are packaged as *RuleApps* for deployment
 - RuleApps: A deployable management container that can hold one or more rulesets
- Deployment target
 - Test or production environment
 - One or more Rule Execution Servers for managed execution



© Copyright IBM Corporation 2015

Figure 5-6. Preparing for deployment

WB393 / ZB3931.0

Notes:

As an administrator, when you are asked to manage deployment, you must know which rulesets are ready for deployment and where they are stored.

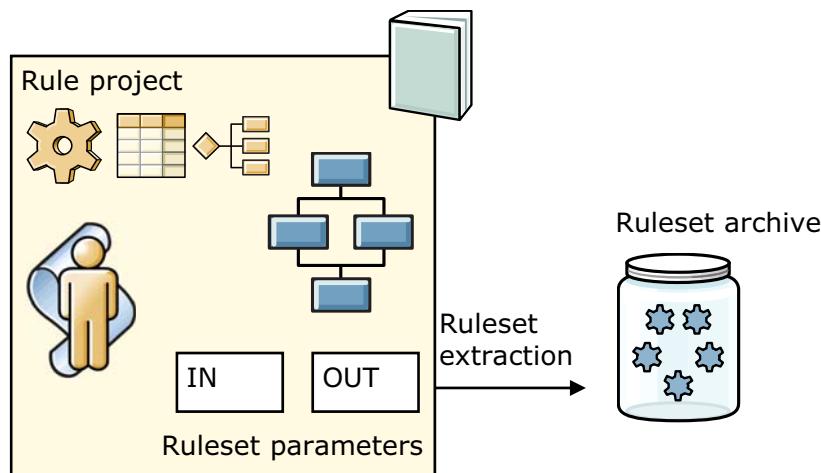
If the Decision Center repository is used as the copy of record, you might be required to support business users who want to deploy from Decision Center.

If developers finalize ruleset testing and maintain the copy of record through SCC, you might be asked to manage deployment from SCC storage.

Before rulesets can be used for rule execution, they must first be packaged as *RuleApps*. RuleApps are deployable management containers that can hold one or more rulesets.

Ruleset archive

- You package rules that relate to a specific decision in a ruleset archive to pass it to the rule engine
- A ruleset archive contains a technical language version of the rule artifacts, and all the supporting data necessary to execute a ruleset



© Copyright IBM Corporation 2015

Figure 5-7. Ruleset archive

WB393 / ZB3931.0

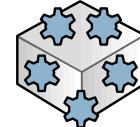
Notes:

The ruleset archive contains a technical language version of the rules and rule artifacts, and all the supporting data necessary to execute your ruleset, including functions, ruleflows, the BOM, and the BOM-to-XOM mapping. The ruleset archive also contains the URLs to the XOMs.

When you create a ruleset archive, the default behavior is to include all rules in the rule project in the archive.

RuleApps and RuleApp archives

- RuleApps are deployable management units that contain one or more rulesets
- A RuleApp keeps a record of the following details:
 - RuleApp name
 - RuleApp version
 - Number of rulesets that the RuleApp manages
 - RuleApp creation date
- You archive a RuleApp to a JAR file
 - Saved as *RuleApp_name.jar*



© Copyright IBM Corporation 2015

Figure 5-8. RuleApps and RuleApp archives

WB393 / ZB3931.0

Notes:

A deployed ruleset contains an archive, which supports its execution. A RuleApp is described in an XML file that is named a RuleApp descriptor. All executable resources are defined as elements of the RuleApp descriptor.

In Rule Designer, to deploy a ruleset to Rule Execution Server, you create a RuleApp project and add the ruleset archive to it.

You can also create the RuleApp from the rule project that defines the ruleset, and Rule Designer automatically adds the ruleset archive to it.

In both cases, you can later add more rulesets to your RuleApp project as required.

i **Information**

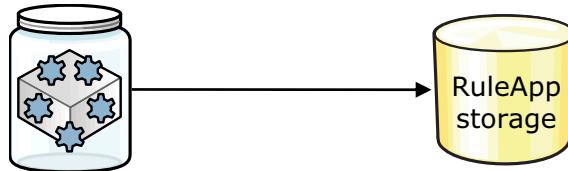
Further ruleset archives can be added to the created RuleApp project from either a ruleset archive file (a `.jar` file) or another rule project.

In Rule Designer, RuleApps are handled inside *RuleApp projects*. RuleApp projects contain the following two files:

- **archive.xml**: This file contains the RuleApp descriptor.
- **.ruleappproject**: This file contains information such as the Rule project dependencies and the ruleset archive paths.

RuleApp archives

- A *RuleApp archive* can be used to store RuleApps in a file system or database



- When you deploy a RuleApp to Rule Execution Server, you also deploy the RuleApp archive to the application server that has Rule Execution Server
- You can use the RuleApp archive (JAR) for deployment
 - Locate the RuleApp JAR file in the RuleApp storage and deploy
 - Result: A RuleApp is created on Rule Execution Server

© Copyright IBM Corporation 2015

Figure 5-9. RuleApp archives

WB393 / ZB3931.0

Notes:

You must create a *ruleset archive* to pass your rules to the rule engine for their execution.

Similarly, you must create a RuleApp archive from your RuleApp, and deploy this RuleApp archive to create the RuleApp on Rule Execution Server.

You can export a RuleApp archive by opening the `archive.xml` file in your RuleApp project and by selecting **Export a RuleApp archive** on the **Overview** tab.

In Rule Designer, when you ask to deploy the RuleApp to Rule Execution Server, Rule Designer creates the RuleApp archive for you.

Ruleset path

- Each ruleset in a RuleApp can be identified by using a ruleset path in this format:
`/ {RuleApp name} [/ {version}]/{ruleset name} [/ {version}]`
- The canonical ruleset path is defined by:
 - Name and the version number of the RuleApp
 - Name and the version number of the ruleset
- The ruleset path acts as the entry point for client applications to access the decision logic that is encapsulated in the RuleApp
- The client application uses the ruleset path to identify the ruleset to execute
 - To reference the *latest* ruleset, use the non-canonical path by omitting the version fields from the ruleset path

© Copyright IBM Corporation 2015

Figure 5-10. Ruleset path

WB393 / ZB3931.0

Notes:

Each ruleset in a RuleApp is identified with a unique ruleset path:

`/ {RuleApp name} [/ {RuleApp version}]/{ruleset name} [/ {ruleset version}]`

The ruleset path acts as the entry point for client applications to access the business logic that is encapsulated in the RuleApp. In a client application, the ruleset path identifies the ruleset to execute.

The following information defines a canonical ruleset path:

- Name and the version number of the RuleApp
- Name and the version number of the ruleset

In a client application, you might omit the version fields in the ruleset path. By doing so, you ask Rule Execution Server to execute the latest deployed ruleset.

5.2. Deployment options

Deployment options



© Copyright IBM Corporation 2015

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

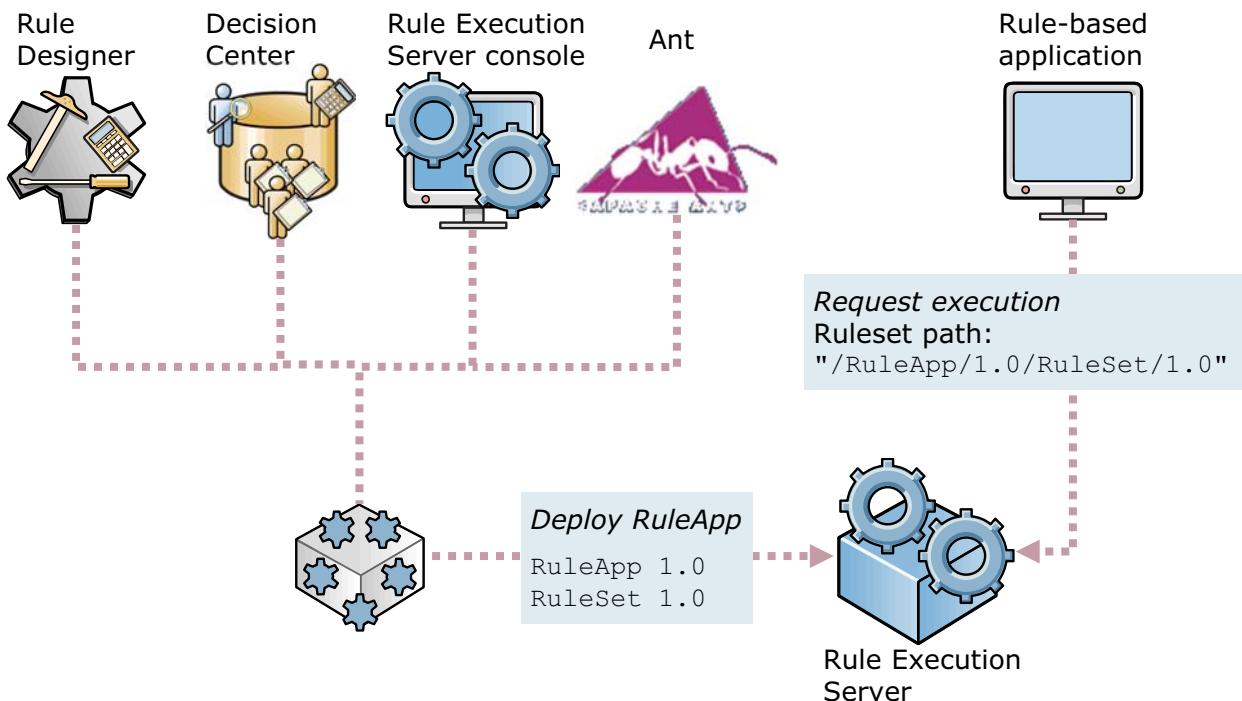
10.1

Figure 5-11. Deployment options

WB393 / ZB3931.0

Notes:

Deployment options



© Copyright IBM Corporation 2015

Figure 5-12. Deployment options

WB393 / ZB3931.0

Notes:

You can deploy rulesets from Rule Designer, from Decision Center, from the Rule Execution Server console, or by running Ant tasks. You can also deploy either with or without a managed XOM.

Decision Server includes tools for you to deploy and execute rules on the Java platform. You use Rule Execution Server to execute the rulesets that are packaged in RuleApps in either Java SE or Java EE. RuleApp deployment is the most common method to deploy or update rules.

Deployment from Rule Designer

- When you deploy a RuleApp from Rule Designer to Rule Execution Server, Rule Designer creates the RuleApp archive
- You can optionally deploy the XOM along with the RuleApp
 - Enables the XOM to be accessed and stored independently from Rule Designer

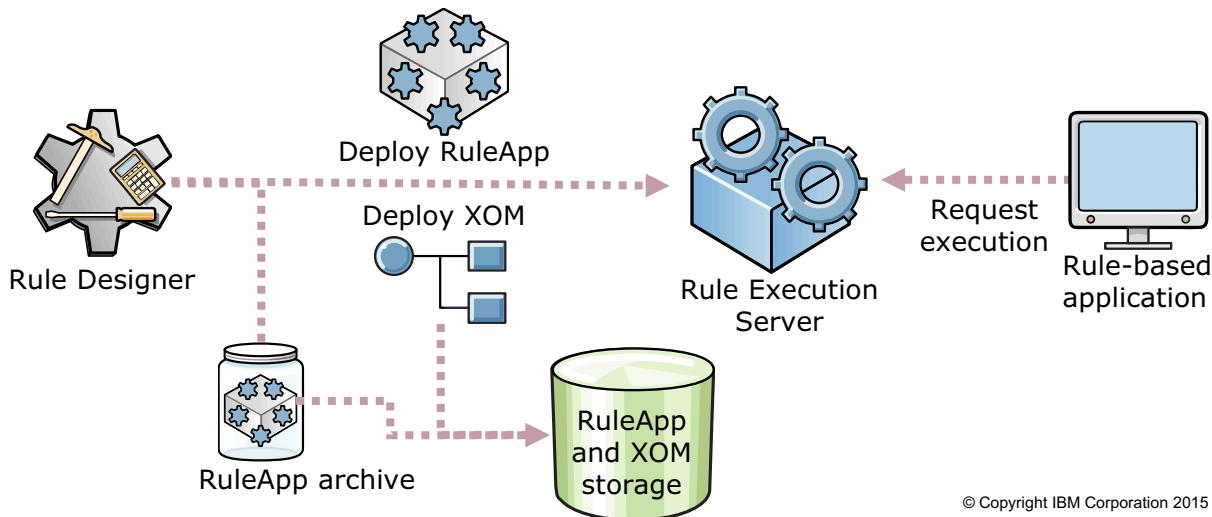


Figure 5-13. Deployment from Rule Designer

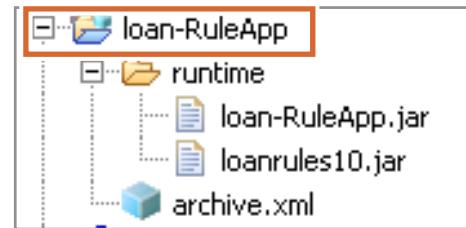
WB393 / ZB3931.0

Notes:



RuleApp projects in Rule Designer (1 of 2)

- Rule Designer project that handles RuleApps and contains:
 - A RuleApp descriptor in `archive.xml`
 - A `.ruleappproject` file with rule project dependencies and ruleset archive paths
- The RuleApp descriptor file is named `archive.xml` and contains the basic metadata of the RuleApp:
 - Name, version, and status of the RuleApp
 - Name, version, and status of each ruleset



© Copyright IBM Corporation 2015

Figure 5-14. RuleApp projects in Rule Designer (1 of 2)

WB393 / ZB3931.0

Notes:

In Rule Designer, to deploy a ruleset to Rule Execution Server, you create a RuleApp project and add the ruleset archive to it.

You can also create the RuleApp from the rule project that defines the ruleset, and Rule Designer automatically adds the ruleset archive to it.

In both cases, you can later add more rulesets to your RuleApp project as required.



Information

Further ruleset archives can be added to the created RuleApp project from either a ruleset archive file (a `.jar` file) or another rule project.



RuleApp projects in Rule Designer (2 of 2)

- Developers create a RuleApp project as a container for deploying rulesets to Rule Execution Server

Figure 5-15. RuleApp projects in Rule Designer (2 of 2)

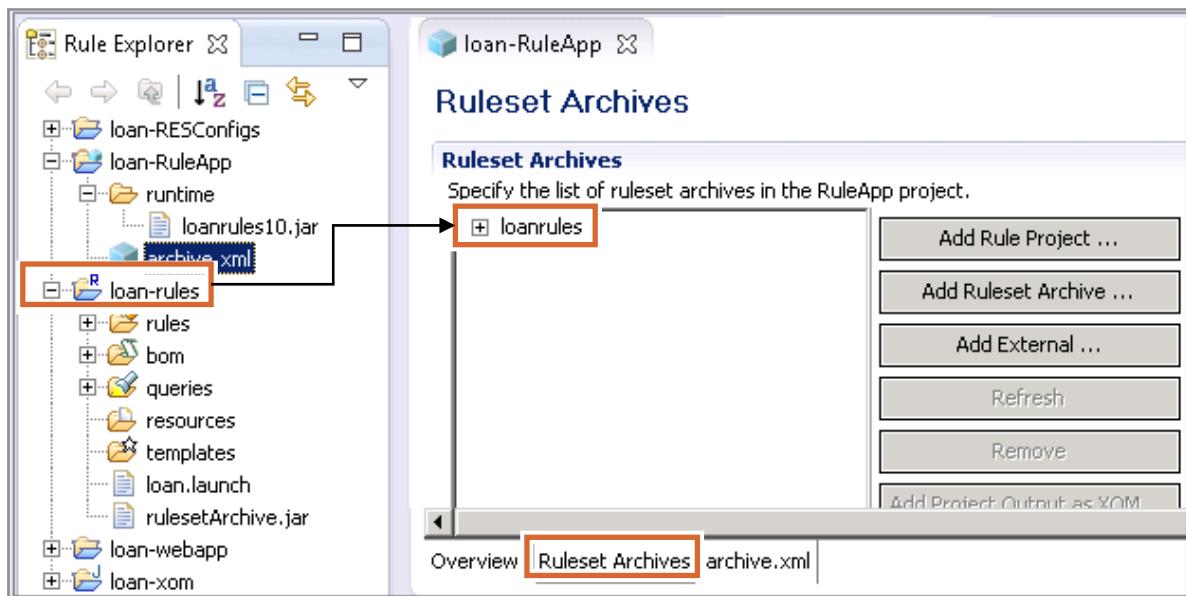
WB393 / ZB3931.0

Notes:

Use a RuleApp project when you want to deploy the same RuleApp simultaneously to multiple Rule Execution Servers.

Managing rulesets in the RuleApp

- Use the **Ruleset Archives** tab to add or remove rulesets within the RuleApp
 - Add a rule project or a ruleset archive



© Copyright IBM Corporation 2015

Figure 5-16. Managing rulesets in the RuleApp

WB393 / ZB3931.0

Notes:



Using a Rule Execution Server configuration (1 of 2)

- Use a *Rule Execution Server configuration* to identify your target Rule Execution Server for deployment

IBM WebSphere AS 8.5

Rule Execution Server Configuration: IBM WebSphere AS 8.5

Environment

an application server
 Rule Execution Server console deployed

Server Information

Server: IBM WebSphere AS 8.5
 Installation directory: C:/Program Files/IBM/ODM87/WAS/AppServer
 Deployment directory:

RuleApp Deployment

The RuleApp deployment used for this configuration is :
 to the Rule Execution Server console
 to a file system

Destination directory:

to a database
 Driver:
 Driver Path:

• You can manage configurations in a *Rule Execution Server configuration project* if you want the configuration persisted

© Copyright IBM Corporation 2015

Figure 5-17. Using a Rule Execution Server configuration (1 of 2)

WB393 / ZB3931.0

Notes:

Where you deploy the RuleApp depends on the architecture of your enterprise environment.

- The context of your project
- The application constraints (web application, Java EE, Java SE)
- Your business requirements (extraction, hot deployment)

How you deploy the RuleApp also depends on how the components of Rule Execution Server are deployed.

In Rule Designer, you create a *Rule Execution Server configuration* to define and persist such parameters and constraints.

For example, you can save these elements in a Rule Execution Server configuration:

- The connection parameters for the Rule Execution Server where to deploy, for example:
 - URL
 - User name
 - Password

- The parameters to locate the Rule Execution Server console, if it is deployed
- The persistence layer where to store the RuleApp, for example:
 - Database elements: In Rule Execution Server database persistence
 - File hierarchy: In Rule Execution Server file persistence

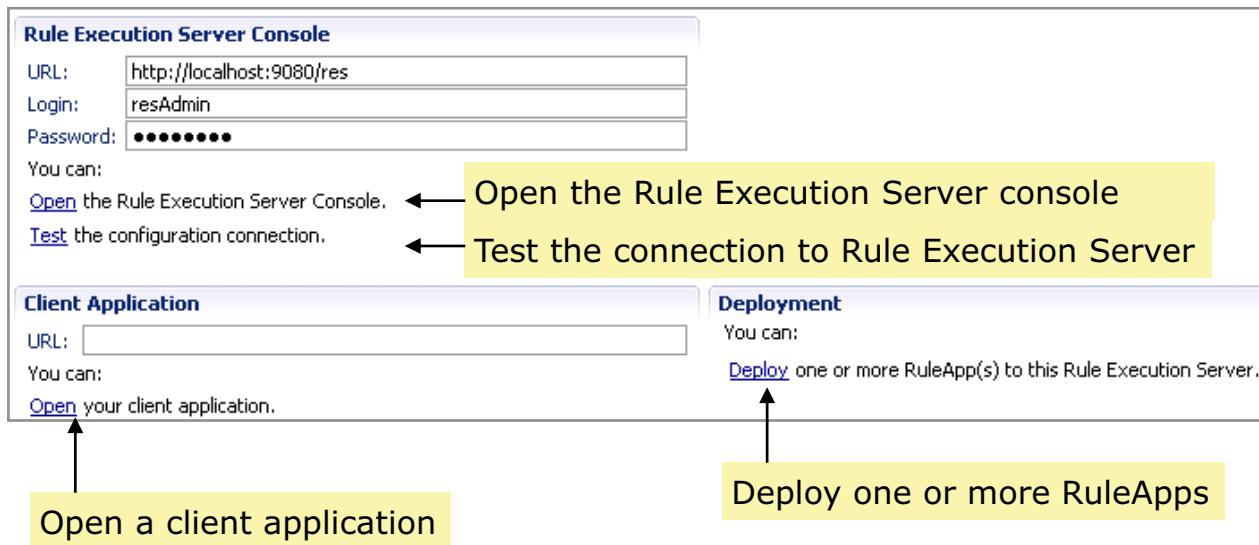
You create and manage Rule Execution Server configurations by using a *Rule Execution Server* configuration project.

When you use a configuration for deployment, you can deploy multiple RuleApps simultaneously to the same Rule Execution Server.



Using a Rule Execution Server configuration (2 of 2)

- Use the configuration to:



© Copyright IBM Corporation 2015

Figure 5-18. Using a Rule Execution Server configuration (2 of 2)

WB393 / ZB3931.0

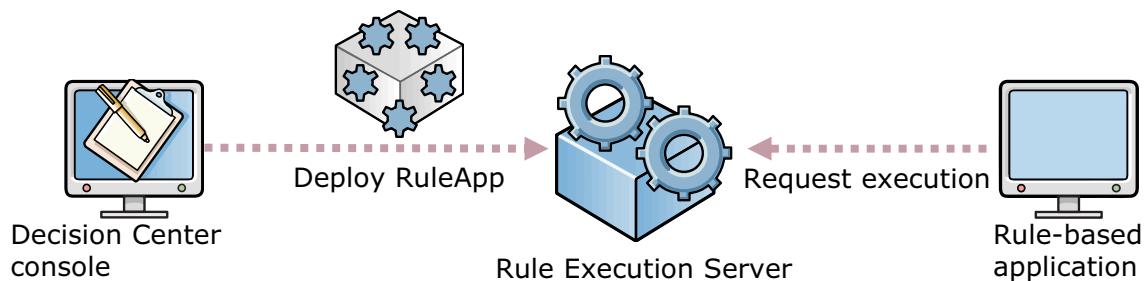
Notes:

You can use this configuration in Rule Designer to do the following actions:

- Deploy one or more RuleApps
- Test the connection to Rule Execution Server
- Open the Rule Execution Server console
- Open a client application

Deployment from Decision Center

- Business users can deploy RuleApps from Decision Center to a test environment
- As an administrator, you might manage deployment through the Enterprise console if Decision Center is considered to be the “source of truth” for the rules



© Copyright IBM Corporation 2015

Figure 5-19. Deployment from Decision Center

WB393 / ZB3931.0

Notes:

RuleApps can be deployed from Decision Center to Rule Execution Server for managed execution.

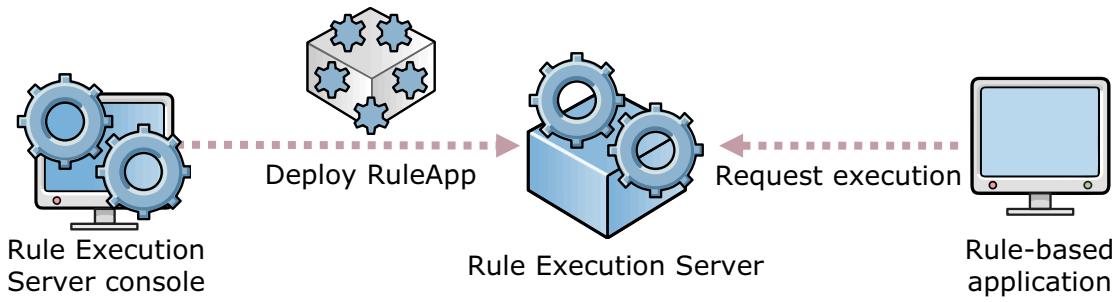


Important

From Rule Designer, you publish rule projects, not rulesets, to Decision Center. Rulesets are engine artifacts, not authoring artifacts. However, in the Decision Center Enterprise console, you can also create RuleApps from the rule projects and deploy them to Rule Execution Server.

Deployment from Rule Execution Server console

- Administrators and developers can use the Rule Execution Server console to deploy or redeploy updated rulesets
 - A RuleApp archive or a ruleset archive must first be generated from Rule Designer
 - Or, you can use a ruleset archive that was generated in Rule Designer, create the RuleApp on Rule Execution Server console, add the ruleset archive to it, and then deploy



© Copyright IBM Corporation 2015

Figure 5-20. Deployment from Rule Execution Server console

WB393 / ZB3931.0

Notes:

When you are logged in to the Rule Execution Server console, you can either:

- Deploy the RuleApp from the Rule Execution Server console

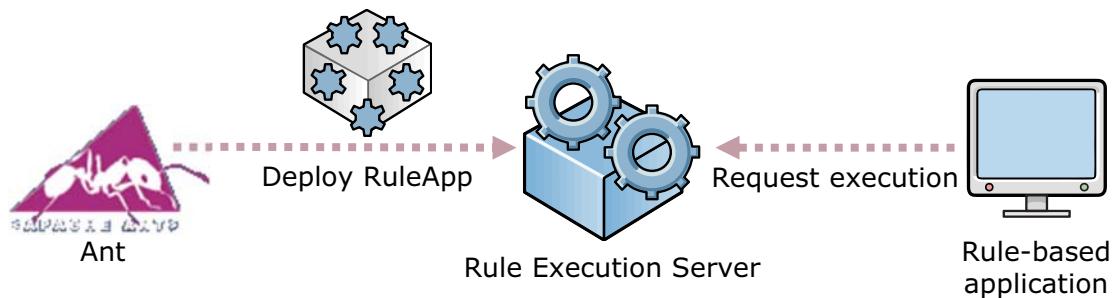
Or:

- Create a RuleApp in the Rule Execution Server console
- Add the ruleset archive to the RuleApp from the Rule Execution Server console, and then deploy

You can use the Rule Execution Server console for deployment and to monitor rule execution in Rule Execution Server. You learn more about the Rule Execution Server console in Unit 6, "Administering Rule Execution Server".

Deployment with Ant scripts

- ODM provides Ant tasks that you can use in scripts to automate your RuleApp management, particularly RuleApp deployment
- With these Ant tasks, you can:
 - Create RuleApps
 - Download RuleApp archives
 - Deploy or remove RuleApp archives
 - Store RuleApp archives
 - Remove RuleApp archives



© Copyright IBM Corporation 2015

Figure 5-21. Deployment with Ant scripts

WB393 / ZB3931.0

Notes:

ODM provides Ant scripts that you can use to automate your RuleApp management, in particular RuleApp deployment.

- res-jar:** To create RuleApps
- res-fetch:** To download a RuleApp archive
- res-fetch-all:** To download all your RuleApp archives
- res-deploy:** To deploy RuleApp archives
- res-undeploy:** To remove a RuleApp archive
- res-write-file:** To store RuleApp archives as files
- res-write-db:** To write a RuleApp to a database
- res-delete-file:** To remove RuleApp archives that are stored as files
- res-delete-db:** To remove RuleApp archives from a database

5.3. RuleApp and ruleset properties

RuleApp and ruleset properties



© Copyright IBM Corporation 2015

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

10.1

Figure 5-22. RuleApp and ruleset properties

WB393 / ZB3931.0

Notes:



Properties

- Before deployment, you can set predefined properties on RuleApps or rulesets

The screenshot shows the IBM Rational Rules interface. On the left, the 'Ruleset Properties' dialog is open, displaying fields for 'Display name' (loan-rules), 'Name' (loanrules), 'Description', 'Version' (1.0), and 'Ruleset extractor'. Below these are sections for 'Create your own properties:' and a table with two entries: 'rulesetSEQUENTIAL.trace.enabled' (true) and 'monitoring.enabled' (true). On the right, two 'Edit Property' dialogs are shown. The top one is for a RuleApp property, with 'Name' set to 'ruleapp.interceptor.classname' and 'Value' set to 'ruleapp.interceptor.description'. The bottom one is for a ruleset property, with 'Name' set to 'monitoring.filters' and 'Value' set to 'ilog.rules.engine.useStaticAgenda'. A scrollable list of other properties is visible in the bottom dialog.

Figure 5-23. Properties

WB393 / ZB3931.0

Notes:

Several RuleApp properties and ruleset properties are predefined to handle the most common requirements in terms of configuration of the rule engine behavior at run time.

For example:

- `ruleapp.interceptor.classname` is a predefined RuleApp property that you use to define the *ruleset interceptor* that applies to the rulesets in your RuleApp.

A ruleset interceptor is a piece of code in your client application that you use to select rulesets at run time, and to add services to execution components transparently. For example, with a ruleset interceptor, you can check your ruleset paths and change them as required, or modify the input parameters.



Note

Ruleset interceptors are not explained in further detail in this course.

- `rulesetSEQUENTIALtraceenabled` is a predefined ruleset property that you use to enable, or disable, the trace mode of the rule engine for business rules that are run with the sequential execution mode or the Fastpath execution mode.
- `rulesettraceenabled` is a predefined ruleset property that you use to enable, or disable, the trace mode of the rule engine for business rules that are run with the RetePlus execution mode.

You can set properties to a RuleApp that are applicable to all rulesets in this RuleApp. Properties that you set on a ruleset apply only to that ruleset.

To set ruleset properties, you open the `archive.xml` file in your RuleApp project and on the **Ruleset Archives** tab, select the ruleset where to set properties. You can then add, edit, or remove ruleset properties in the **Ruleset Properties** grid.



Note

You can also set RuleApp properties and ruleset properties by using the Rule Execution Server console. You learn more about the Rule Execution console in Unit 6, "Administering Rule Execution Server".

During the exercises, you learn how to set and use these properties to monitor execution activity and enhance performance.

Deployed RuleApp and ruleset names

- When you deploy to Rule Execution Server, only these characters are recognized for RuleApp and ruleset names:
 - Letters (a–z, A–Z)
 - Digits (0–9)
 - The underscore character (_)
- If your RuleApp or ruleset names contain other characters or spaces, the Rule Execution Server removes them
 - Names of deployed RuleApp and rulesets might appear differently in Rule Execution Server than in Rule Designer or Decision Center console
- The ruleset path is based on the RuleApp name and on the ruleset name as they appear in Rule Execution Server

© Copyright IBM Corporation 2015

Figure 5-24. Deployed RuleApp and ruleset names

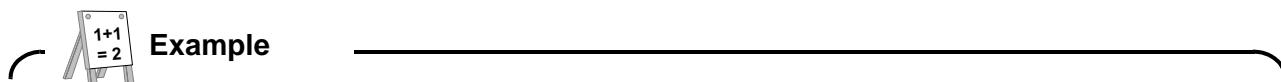
WB393 / ZB3931.0

Notes:

Names of RuleApps and rulesets that are deployed to Rule Execution Server can contain only letters (a–z, A–Z), digits (0–9), and the underscore character (_).

When a RuleApp is deployed to Rule Execution Server, the RuleApp name and the name of its rulesets in Rule Execution Server might differ from what you initially had in Rule Designer or Decision Center. Characters that are not authorized are removed.

When you use ruleset paths, for example in a client application, you must base them on the RuleApp names and on the ruleset names **as they exist in Rule Execution Server**.



Example

Consider a loanvalidation-RuleApp RuleApp packaging a loanvalidation-rules ruleset. After you deploy the loanvalidation-RuleApp RuleApp to Rule Execution Server, you can see the following elements in Rule Execution Server console:

- A loanvalidationRuleApp RuleApp
- A loanvalidationrules ruleset

The ruleset path for the `loanvalidationrules` ruleset is equal to:

`loanvalidationRuleApp/<RuleApp version>/loanvalidationrules/<ruleset version>`

5.4. Version policies for deployment

Version policies for deployment



© Copyright IBM Corporation 2015

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

10.1

Figure 5-25. Version policies for deployment

WB393 / ZB3931.0

Notes:



Version policy options (1 of 3)

- Use version options to control how RuleApp and ruleset versions are numbered and replaced during deployment
- RuleApp version numbering depends whether the RuleApp exists in the server memory:
 - If the RuleApp does not exist, it is deployed to Rule Execution Server with the version number 1.0
 - If the RuleApp exists, it is deployed according to the version policy

© Copyright IBM Corporation 2015

Figure 5-26. Version policy options (1 of 3)

WB393 / ZB3931.0

Notes:

You use version options to control how RuleApp and ruleset versions are numbered and replaced during deployment.

RuleApp version numbering is different depending on whether the RuleApp in the selected archive file exists in the server memory:

- If the RuleApp does not exist, it is deployed to Rule Execution Server with the version number 1.0, irrespective of the selected deployment policy.
- If the RuleApp exists, it is deployed according to the version policy.

Version policy options (2 of 3)

Policy	Deployed RuleApps	RuleApp to be deployed	Result after deployment
Increment RuleApp major version	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0 /RuleApp/2.0 /ruleset/1.0
Increment RuleApp minor version	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0 /RuleApp/1.1 /ruleset/1.0
Replace RuleApp version	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0

© Copyright IBM Corporation 2015

Figure 5-27. Version policy options (2 of 3)

WB393 / ZB3931.0

Notes:

When you increase either the major or minor version of the RuleApp, a RuleApp is created on the server. As you see in the example, when you replace a RuleApp, the newly deployed version overwrites the existing RuleApp and all its rulesets.

Version policy options (3 of 3)

Policy	Deployed RuleApps	RuleApp to be deployed	Result after deployment
Increment Ruleset(s) major version	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0 /ruleset/3.0
Increment Ruleset(s) minor version	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0 /ruleset/2.1
Replace Ruleset(s) versions	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0	/RuleApp/1.0 /ruleset/1.0	/RuleApp/1.0 /ruleset/1.0 /ruleset/2.0

© Copyright IBM Corporation 2015

Figure 5-28. Version policy options (3 of 3)

WB393 / ZB3931.0

Notes:

When you increase either the major or minor version of the ruleset, a ruleset is created within the existing RuleApp on the server. When you replace a ruleset, the newly deployed version overwrites the existing ruleset with the same version number.

5.5. Managing XOMs

Managing XOMs



© Copyright IBM Corporation 2015

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

10.1

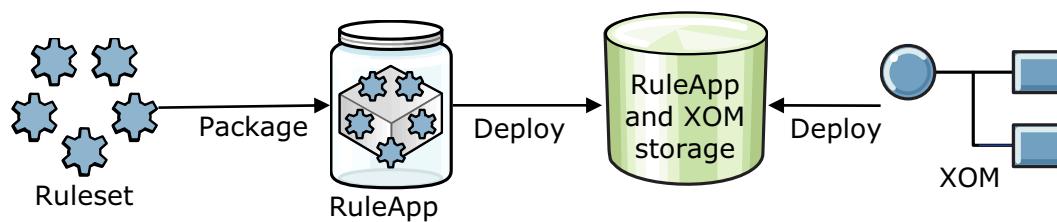
Figure 5-29. Managing XOMs

WB393 / ZB3931.0

Notes:

Introduction to Java XOM management

- Management of the Java XOM supports the execution of decision services on SOA platforms
- Management of the Java XOMs for rule projects means:
 - Deploy a Java XOM to Rule Execution Server
 - Manage the Java XOM resources or libraries through the Rule Execution Server console, Ant tasks, and Rule Designer
 - Resources are accessible to Java SE and Java EE execution components in Rule Execution Server



© Copyright IBM Corporation 2015

Figure 5-30. Introduction to Java XOM management

WB393 / ZB3931.0

Notes:

You can deploy a Java XOM to Rule Execution Server from Rule Designer or by using command-line Ant tasks.

You can manage the Java XOM resources or libraries through the Rule Execution Server console, Ant tasks, and Rule Designer.

After the Java XOM is deployed, its resources or libraries are stored in the same persistence layer as your RuleApps. (You can also select a different persistence type.) All Java SE and Java EE execution components in Rule Execution Server can then access these resources or libraries.

For more information about the persistence layer, the Java SE, and Java EE execution components in Rule Execution Server, see Unit 6, "Administering Rule Execution Server".



Managed Java XOM elements

- You can manage the Java XOM *resources* and *libraries* that are attached to a ruleset
- A Java XOM *resource* is a `.jar` file or a `.zip` file
 - You identify a Java XOM resource with a unique URI based on the `resuri` protocol
 - Example: `resuri://common-classes.jar/1.0`
- A Java XOM *library* is a set of resources or libraries, or both
 - You identify such a library with a unique URI based on the `reslib` protocol
 - Example: `reslib://loan-xom/1.4`

© Copyright IBM Corporation 2015

Figure 5-31. Managed Java XOM elements

WB393 / ZB3931.0

Notes:

The Java XOM resources are either:

- `.jar` files: Java archive files, with the `.jar` extension
- `.zip` files: Compressed files that contain Java classes and the files that they use

The `resuri` URI is created with the name of the file and a version, which is incremented each time the file checksum changes.



Link from a ruleset to a managed Java XOM element

- The `ruleset.managedxom.uris` ruleset property defines the link between a specific ruleset and a specific managed Java XOM element
- The value of this ruleset property is a list of URIs to `.jar` files, `.zip` files, or libraries
- A specific ruleset is linked to a specific managed Java XOM element if:
 - This ruleset defines the `ruleset.managedxom.uris` ruleset property
 - The value of this ruleset property contains the URI to this Java XOM element

© Copyright IBM Corporation 2015

Figure 5-32. Link from a ruleset to a managed Java XOM element

WB393 / ZB3931.0

Notes:

The `ruleset.managedxom.uris` ruleset property defines the link between a specific ruleset and a specific managed Java XOM element.

The value of this ruleset property is a list of URIs to `.jar` files, `.zip` files, or libraries. The value of the `ruleset.managedxom.uris` ruleset property must contain only URIs that follow the `resuri` protocol or the `reslib` protocol. In the value, the URIs are comma-separated.

A specific ruleset is linked to a specific managed Java XOM element if this ruleset defines the `ruleset.managedxom.uris` ruleset property and the value contains the URI to the XOM.



Java XOM deployment (1 of 2)

- You can deploy a Java XOM from Rule Designer:
 - When you deploy from the rule project
 - When you deploy the RuleApp
- The deployed Java XOM is stored in the Rule Execution Server persistence layer and can be managed in the same way as a RuleApp
- When you deploy a RuleApp along with its Java XOM, Rule Designer sets the `ruleset.managedxom.uris` ruleset property of the ruleset that is packaged in the deployed RuleApp

© Copyright IBM Corporation 2015

Figure 5-33. Java XOM deployment (1 of 2)

WB393 / ZB3931.0

Notes:

You can deploy a Java XOM from Rule Designer:

- From the rule project: In the Rule Explorer, right-click the rule project and click **Rule Execution Server > Deploy XOM**
- When you deploy the RuleApp for that rule project, either from a Rule Execution Server configuration or from the RuleApp itself

In both cases, with the deployment wizard, you can indicate whether you also want to deploy the Java XOM associated with the rule project that corresponds to the ruleset in the RuleApp. If you select the option to deploy the Java XOM when you deploy the RuleApp, Rule Designer defines and properly sets the `ruleset.managedxom.uris` ruleset property.



Java XOM deployment (2 of 2)

- When you deploy the Java XOM from a specific rule project to Rule Execution Server:
 - The `deployment.xml` file is created (or updated) in the **resources** folder of the rule project
 - This file contains the used URL, user name, password, and the list of deployed XOMs

© Copyright IBM Corporation 2015

Figure 5-34. Java XOM deployment (2 of 2)

WB393 / ZB3931.0

Notes:

As stated in the previous slide, you can deploy a Java XOM from Rule Designer from the rule project or when you deploy a RuleApp either from the RuleApp or the Rule Execution Server Configuration.

However, the `deployment.xml` file is created, or updated, in the **resources** folder of the rule project **only** if you deploy the Java XOM **from the rule project**.



Managed XOM and Decision Center

- If you are using Decision Center and want to deploy a RuleApp that is linked to a managed XOM, make sure that the rule project in Decision Center also contains the `description.xml` file
 - Contains the URL of the Rule Execution Server where the managed XOM is deployed
- If the XOM is modified after deployment:
 - Redeploy it to the Rule Execution Server to update the `description.xml` file
 - Synchronize Rule Designer and Decision Center
- If you deploy the RuleApp from Decision Center, and want to have the `ruleset.managedxom.uris` property added to the ruleset in the deployed RuleApp, use the URL to Rule Execution Server that exists in the `description.xml` file

© Copyright IBM Corporation 2015

Figure 5-35. Managed XOM and Decision Center

WB393 / ZB3931.0

Notes:

To make sure that the rule project in Decision Center contains the `deployment.xml` file, you must first create or update this file in Rule Designer. You deploy the XOM to Rule Execution Server from the rule project. You must then synchronize the rule project with Decision Center to share the `description.xml` file.

To redeploy a modified XOM to Rule Execution Server, and update the `deployment.xml` file, you must deploy the XOM from the rule project. Deploying the XOM at the same time that you deploy a RuleApp does not update the `deployment.xml` file in the rule project.

For example, if the URL to Rule Execution Server is `http://localhost:9081/res` in the `deployment.xml` file, then you must use that URL as the address of Rule Execution Server when you deploy the RuleApp from Decision Center. If you use `http://127.0.0.1:9081/res` instead, the URLs do not exactly match, and:

1. Decision Center considers that you are deploying to another Rule Execution Server.
2. Decision Center does not add the `ruleset` property, and the deployed RuleApp does not benefit from the managed XOM.

Managed XOM, Rule Designer, and Decision Center

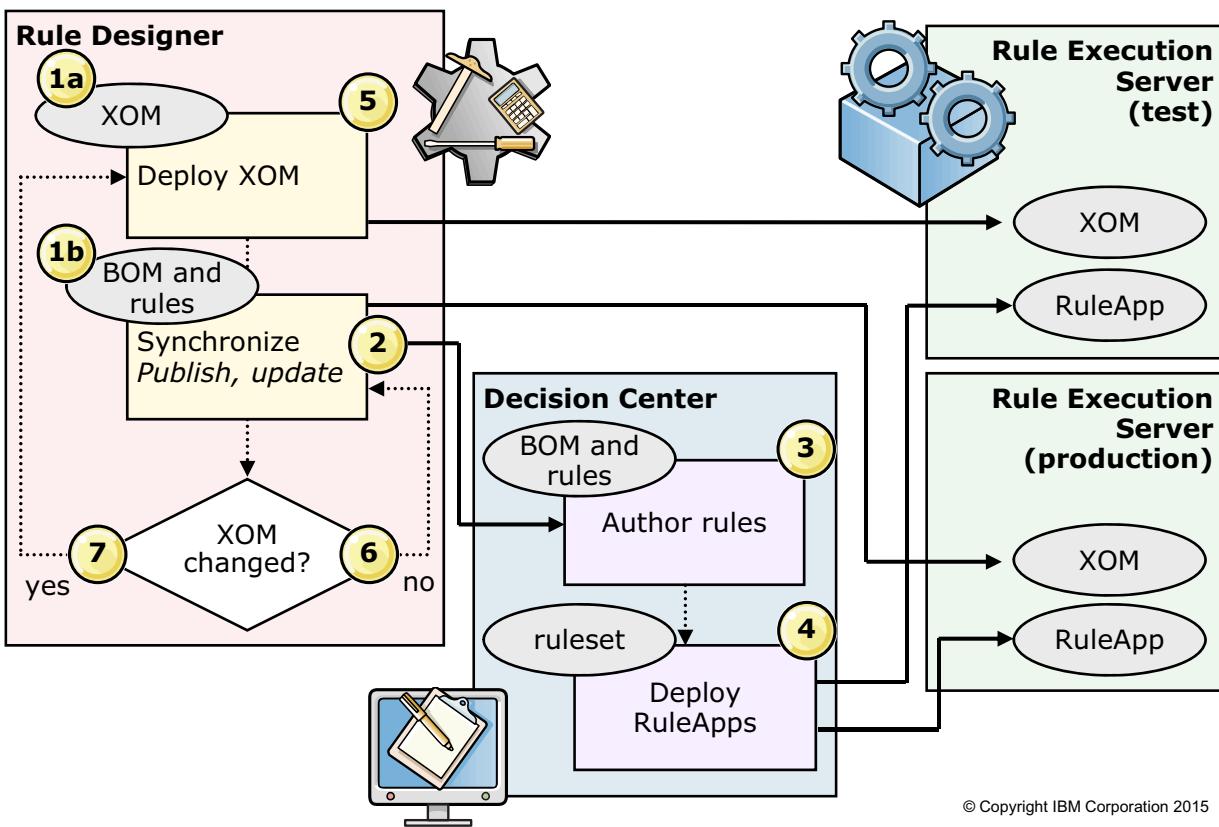


Figure 5-36. Managed XOM, Rule Designer, and Decision Center

WB393 / ZB3931.0

Notes:

This diagram recaps the actions that developers or business analysts complete in Rule Designer, the actions that business users complete in Decision Center regarding the managed XOM, and the data that these actions work on.

1. Programming with rules starts in Rule Designer, where:
 - a. Developers or business analysts design the XOM based on data model.
 - b. Developers or business analysts create the rule projects with the BOM based on the XOM, the associated vocabulary, the ruleflows, the rule packages, and possibly some of the rule artifacts.
2. Then, developers or business analysts synchronize the Rule Designer and Decision Center.
3. After synchronization, business users can author the required rule artifacts in Decision Center (and Rule Solutions for Office, synchronizing with Decision Center; not shown on this diagram).
4. To test or execute the rulesets, deploy the RuleApps. You can do so from Rule Designer, Decision Center (as shown in the diagram), or even from the Rule Execution Server console.

5. To manage the XOM, developers or business analysts must deploy the XOM to the Rule Execution Servers that are used for tests or for production.
6. If the XOM does not evolve, then it is possible to synchronize Decision Center and Rule Designer environments at any time, thus ensuring that all actors of the rule application are using the same artifacts.
7. If the XOM evolves in Rule Designer, then the XOM must be redeployed from Rule Designer to the test or production execution environment. The rule project must also be synchronized to ensure that the latest version of the XOM is also known in Decision Center.

As guidelines:

- Deploy the XOM each time it is changed.
- Between two XOM changes, the BOM and rule artifacts (the rule project) can be synchronized between the environments as required.

Managed XOM and Decision Validation Services

- You can deploy the XOM from Rule Designer to execute the ruleset in the tests and validations environment, with Decision Validation Services
- If you deploy the XOM to test the rulesets remotely by using Decision Validation Services, and want business users to use the same XOM for their tests in Decision Center, synchronize the project with Decision Center

© Copyright IBM Corporation 2015

Figure 5-37. Managed XOM and Decision Validation Services

WB393 / ZB3931.0

Notes:

For business users who are testing rulesets with Decision Validation Services, administrators or developers must ensure that the XOM is deployed to the test server.



Note

Decision Validation Services are not covered in this course.



Managed XOM storage

- You can store the managed Java XOM in the persistence layer of Rule Execution Server independently of the ruleset
 - For example, you can work with local `.jar` files if they are too large to work from a database in a test environment

© Copyright IBM Corporation 2015

Figure 5-38. Managed XOM storage

WB393 / ZB3931.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Explain deployment principles and options
- Package rulesets as RuleApps for deployment
- Manage RuleApp deployment from Rule Designer and Decision Center
- Describe the basic aspects of XOM management

© Copyright IBM Corporation 2015

Figure 5-39. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

- 1. True or False:** In Rule Designer, Rule Execution Server configurations are persisted in a Rule Execution Server configuration project.

- 2. What can you use to deploy rules to Rule Execution Server? Choose all that apply.**
 - A. Rule Designer
 - B. Decision Center Business console
 - C. Rule Execution Server console
 - D. Ant tasks

- 3. True or False:** The client application uses the ruleset path to identify which ruleset to execute.

© Copyright IBM Corporation 2015

Figure 5-40. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.

- 2.

- 3.



Checkpoint answers

- 1. True.**
- 2. A, C, and D.**
- 3. True.**

© Copyright IBM Corporation 2015

Figure 5-41. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 7



Managing deployment

© Copyright IBM Corporation 2015

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

10.1

Figure 5-42. Exercise 7

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Define RuleApp and ruleset properties
- Manage deployment from Rule Designer and Decision Center
- Deploy client applications

© Copyright IBM Corporation 2015

Figure 5-43. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 6. Administering Rule Execution Server

What this unit is about

This unit explains Rule Execution Server architecture and how to configure it for the managed execution of business rules. The unit also describes how to administer rule execution through the Rule Execution Server console.

What you should be able to do

After completing this unit, you should be able to:

- Describe the Rule Execution Server architecture
- Use the Rule Execution Server console
- Explain how to access and manage resources by using the REST API

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Describe the Rule Execution Server architecture
- Use the Rule Execution Server console
- Explain how to access and manage resources by using the REST API

© Copyright IBM Corporation 2015

Figure 6-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Setting up Rule Execution Server
- Rule Execution Server architecture
- Managed execution in action
- Platforms for Rule Execution Server
- Managing and monitoring through the Rule Execution Server console
- Managing resources with the REST API

© Copyright IBM Corporation 2015

Figure 6-2. Topics

WB393 / ZB3931.0

Notes:

6.1. Setting up Rule Execution Server

Setting up Rule Execution Server



© Copyright IBM Corporation 2015

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

10.1

Figure 6-3. Setting up Rule Execution Server

WB393 / ZB3931.0

Notes:

Preparing for deployment to Rule Execution Server

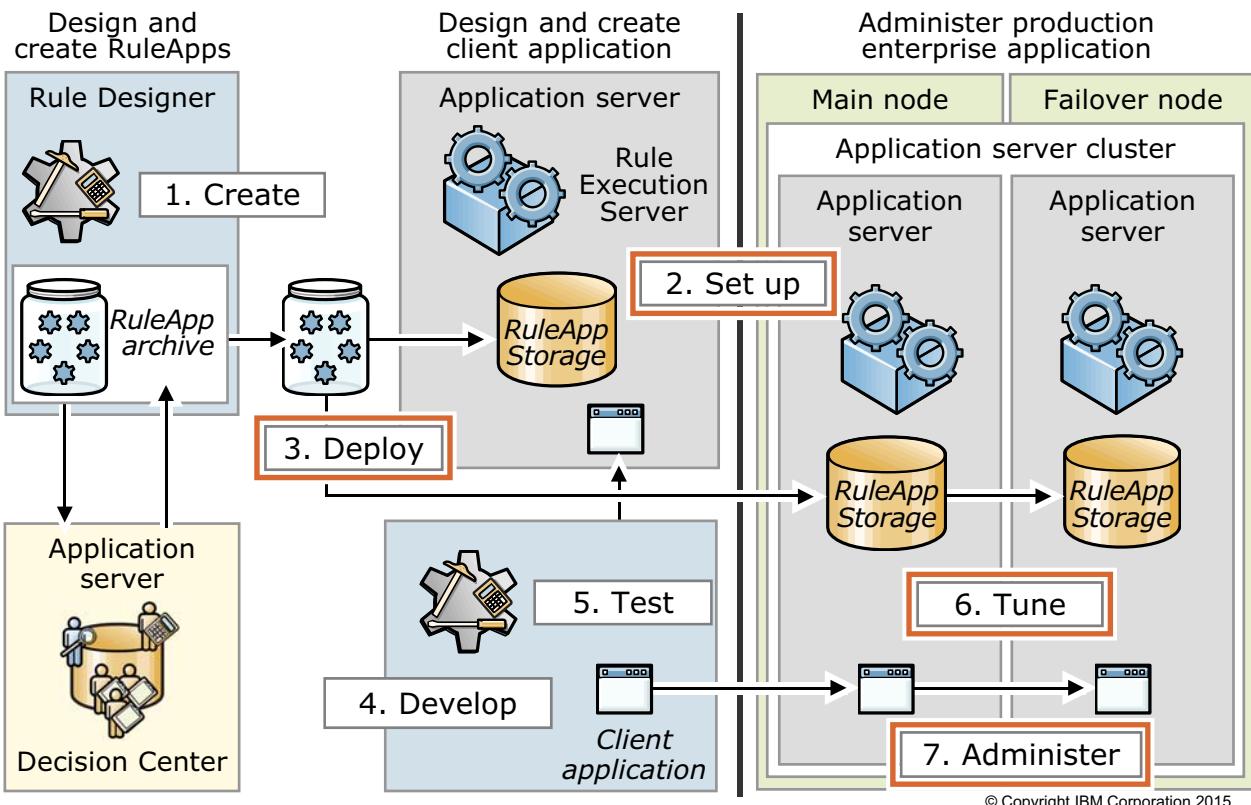


Figure 6-4. Preparing for deployment to Rule Execution Server

WB393 / ZB3931.0

Notes:

Architects, administrators, developers, and testers work together to create, deploy, tune, and administer RuleApps and client applications that run on Rule Execution Server instances on a production enterprise cell.

To use Rule Execution Server, follow the task flow to complete the steps that are applicable to you. The order in which tasks are done depends on your environment.

The figure shows the task flow to create a Rule Execution Server, and the infrastructure that is used at each step. The highlighted steps are administrative tasks that are explained in more detail on the next slide.

1. Create

Developers design and create a RuleApp from a rule project. A RuleApp is a deployment and management unit for Rule Execution Server. A RuleApp contains one or more rulesets.

RuleApps are deployed to the application server to make the ruleset available to a client application.

2. Set up

Administrators set up test servers and production servers. (See next slide.)

3. Deploy

Developers or administrators can deploy the RuleApp to one or more Rule Execution Servers.

4. Develop

Developers also design and create a client application that uses the deployed rules. The client application must contain execution code that calls the deployed ruleset that is contained in a RuleApp on Rule Execution Server.

5. Test

To test ruleset execution, you must run the client application, which calls the ruleset.

Before deploying RuleApps and client applications to a production server, developers and testers must test with sufficient data that the applications are stable and that the results are correct. You can also use Decision Validation Services to verify that your RuleApps return valid results and set up Decision Warehouse to store execution traces.

Client applications are standard Java applications that you test and debug with Rule Designer. Use the JUnit framework for formalized testing and performance monitoring.

6. Tune

Administrators can tune Rule Execution Server and server settings in a production cluster to achieve the best performance. Developers can also look at the rulesets that are executed as decision services. Performance can be improved by reducing the ruleset size and by using mutually exclusive rule tasks.

7. Administer

Administrators can work in Rule Execution Server console to monitor decision services to provide stability.



Administer production enterprise application

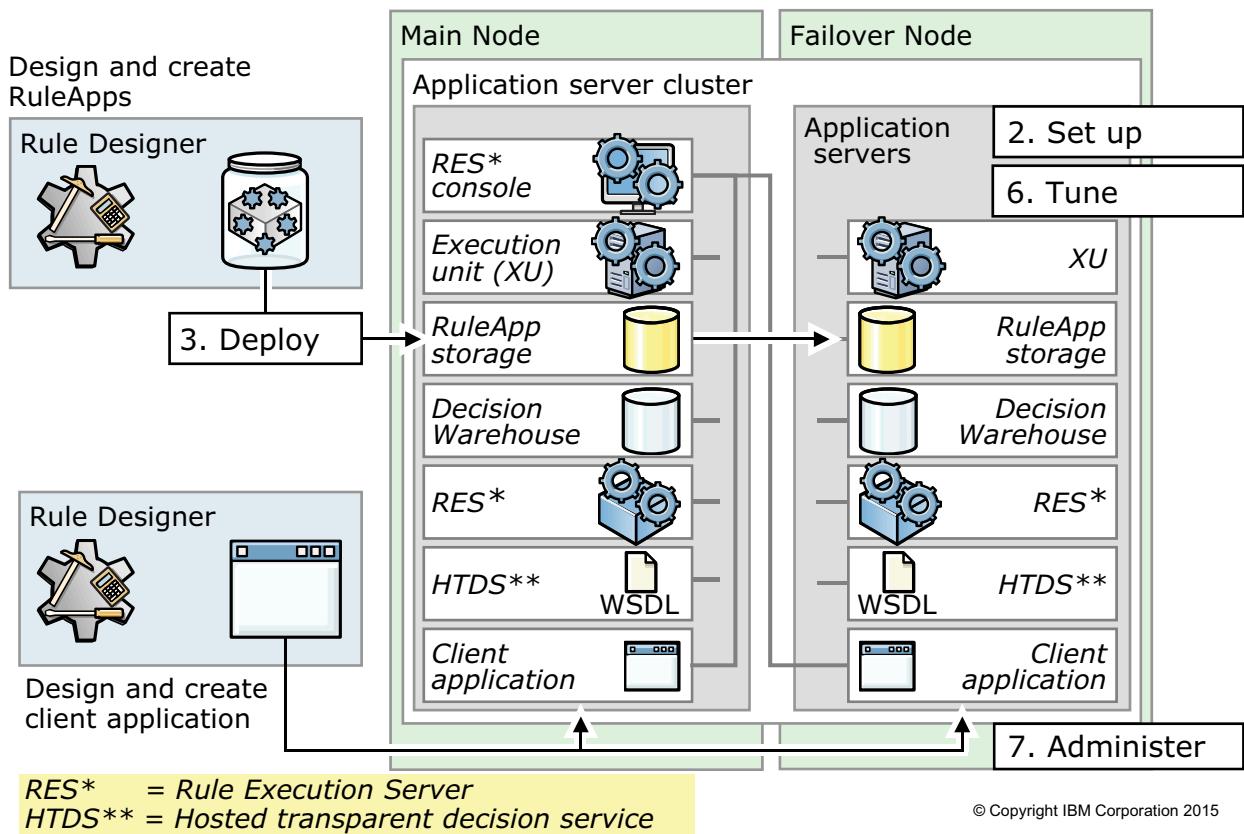


Figure 6-5. Administer production enterprise application

WB393 / ZB3931.0

Notes:

As administrator, you focus on the server setup, deployment, performance tuning, and administration.

2. Set up

Set up includes configuring Rule Execution Server on an application server for testing or on a cluster for production.

By default, Rule Execution Server is installed on the Sample Server, which is used to run the provided samples. You can also use this server to test your installation of Rule Execution Server. If you want to use a test server that matches your production server, see the documentation on how to install Rule Execution Server.

After a RuleApp project is created and configured a connector to an installed Rule Execution Server in Rule Designer, you use the deployment wizard to deploy a RuleApp to one or more Rule Execution Servers. Set up and configure Rule Execution Server and Decision Warehouse on a production application server cluster. You then redeploy your validated RuleApps and client applications.

A production server involves two or more application servers that are installed as a cluster on two or more computers.

3. Deploy

This unit describes the deployment tasks that administrators are involved in to make the rules accessible for execution.

6. Tune Rule Execution Server performance

The goal of performance tuning is to decrease the amount of time and resources that your application server requires to process requests. You can configure the following server settings in your cluster to achieve the most efficient performance possible:

- Log
- Trace level
- Thread pool
- Garbage collection

These tasks are described in later units.

7. Administer

You can use Rule Execution Server console to maintain and monitor your Rule Execution Server and decision services to provide stability and performance. You use the console interface to sign in, monitor execution traces stored in Decision Warehouse, ensure that the server is installed correctly by using the diagnostic tests, manage decision services, and consult various server information.

6.2. Rule Execution Server architecture

Rule Execution Server architecture



© Copyright IBM Corporation 2015

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

10.1

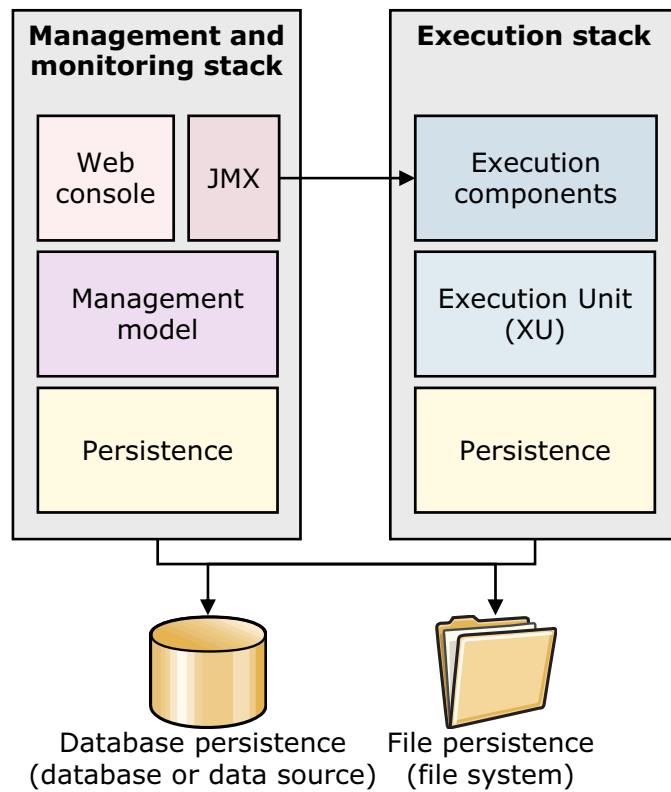
Figure 6-6. Rule Execution Server architecture

WB393 / ZB3931.0

Notes:

Rule Execution Server architecture

- Modular architecture
 - Management and monitoring stack
 - Execution stack
 - Persistence layer
- Set of independent, cooperating components that interact with the rule engine
 - Accommodates various enterprise environments
- Choose what and how to integrate within your enterprise infrastructure



© Copyright IBM Corporation 2015

Figure 6-7. Rule Execution Server architecture

WB393 / ZB3931.0

Notes:

Rule Execution Server has a flexible and modular architecture that can service different server clients and integration with enterprise infrastructure.

Rule Execution Server components are gathered within an execution stack, a management and monitoring stack, and a persistence layer.



Key functions of Rule Execution Server

- Managed environment for rule execution
- Ability to handle rule execution for multiple applications
- Execution scalability by using resource pooling
- Management through a web-based interface and JMX tools
- Full integration in Java EE and Java SE
- Automation with Ant tasks
- Logging, monitoring, and debugging integration

© Copyright IBM Corporation 2015

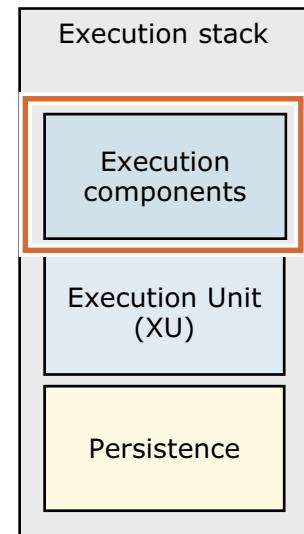
Figure 6-8. Key functions of Rule Execution Server

WB393 / ZB3931.0

Notes:

Execution stack: Execution components

- Execution components (Java SE and Java EE) authorize the execution of a ruleset by the Execution Unit (XU)
- Java SE execution components:
 - Stateless rule sessions
 - Stateful rule sessions
 - Decision traces
 - Ruleset execution interceptors
- Java EE execution components include the same components as Java SE, plus:
 - Message-driven (asynchronous) rule beans
 - Remote method invocation (RMI)



© Copyright IBM Corporation 2015

Figure 6-9. Execution stack: Execution components

WB393 / ZB3931.0

Notes:

The Execution stack comprises:

- Java execution components (Java SE and Java EE)
- JMX Execution Model components
- Execution Units (XU, see next slide)

Java SE execution components and Java EE execution components authorize the execution of a ruleset by the *Execution Unit* (XU). Java SE execution components comprise:

- Stateless ruleset sessions (further detailed next)
- Stateful ruleset sessions (further detailed next)
- Decision traces
- Ruleset execution interceptors

Java EE execution components comprise the same as Java SE execution components, plus:

- Message driven (asynchronous) rule beans (further detailed next)

- Remote method invocation (RMI)



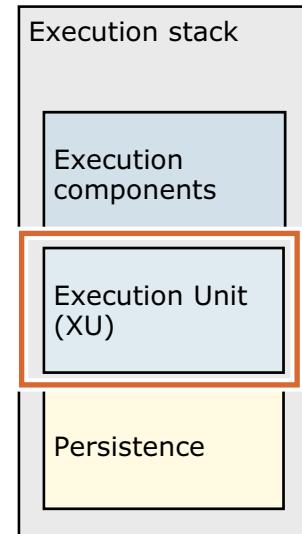
Information

The execution stack is in essence the same for both Java SE and Java EE. This similarity also exists for the management stack.



Execution Unit

- A resource adapter for Java EE Connector Architecture (JCA)
 - Handles the low-level aspects of ruleset execution
 - Collaborates with the server to provide several connector system-level contracts as a Service Provider Interface
- An XU can also run independently of the management model
 - Makes configuration and runtime data available to the management model
 - Implements the JCA contracts between the application server and the rule engine
- Benefits:
 - Scalability
 - High-performance execution of rulesets
 - Execution trace
 - An XU container to create and pool connections of the XU (JCA)
 - Logging
 - Debugging
 - Notification that a RuleApp was modified
 - Statistics



© Copyright IBM Corporation 2015

Figure 6-10. Execution Unit

WB393 / ZB3931.0

Notes:

In Rule Execution Server, an Execution Unit (XU) manages rule engines. The XU is a resource adapter for Java EE Connector Architecture (JCA), which handles the low-level details of ruleset execution and provides access to manage its resources.

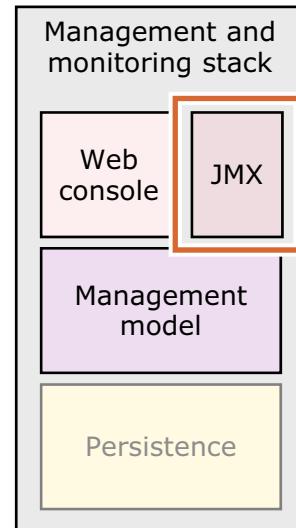
The XU manages rule engines, so the application server or application client uses the XU to connect to the rule engine. The XU retrieves and loads rulesets from persistence layer, passes data between the application and the rule engine, and manages the rule engines that are attached to loaded rulesets. The XU can create several rule engine instances. It also manages hot deployment. For hot deployment, the XU receives notification from the JMX layer when new versions of rulesets are available, and responds appropriately.

As you saw earlier during the configuration of Rule Execution Server, the Execution Unit (XU) is a stand-alone deployable unit. You installed the XU RAR file on the WebSphere Application Server. The RAR file contains the XU and the persistence layer.



JMX management and execution model

- JMX API underlies Rule Execution Server architecture
 - MBeans model system administration functions
 - Access MBeans through Rule Execution Server console
- Management model
 - Provides access to runtime JMX MBeans for the Rule Execution Server model
 - Responsible for hot update and deployment
 - When a new version of a ruleset is deployed, the JMX layer informs all XUs in the cluster that a new version of the ruleset is available in database
- Execution model
 - Provides access to runtime JMX MBeans for the Execution Unit (XU) to notify of changes and retrieve statistics
 - Exposes execution statistics as MBeans that can then be monitored by using JMX tools (like Tivoli)
 - Execution Unit (XU) instances run a local JMX MBean server



© Copyright IBM Corporation 2015

Figure 6-11. JMX management and execution model

WB393 / ZB3931.0

Notes:

The management and monitoring stack comprises:

- Management console
- JMX Management Model components

Rule Execution Server uses the Java Management Extension (JMX) API as its underlying architecture.

The JMX API uses Java objects that are called MBeans to model system administration functions. Each MBean contains a set of attributes that define parameters for various management functions and operations that define administrative actions.

The management and monitoring model components constitute an interface that depends on your application server and that manages business logic, including remote updating and browsing. They are based on JMX, a part of Java SE.

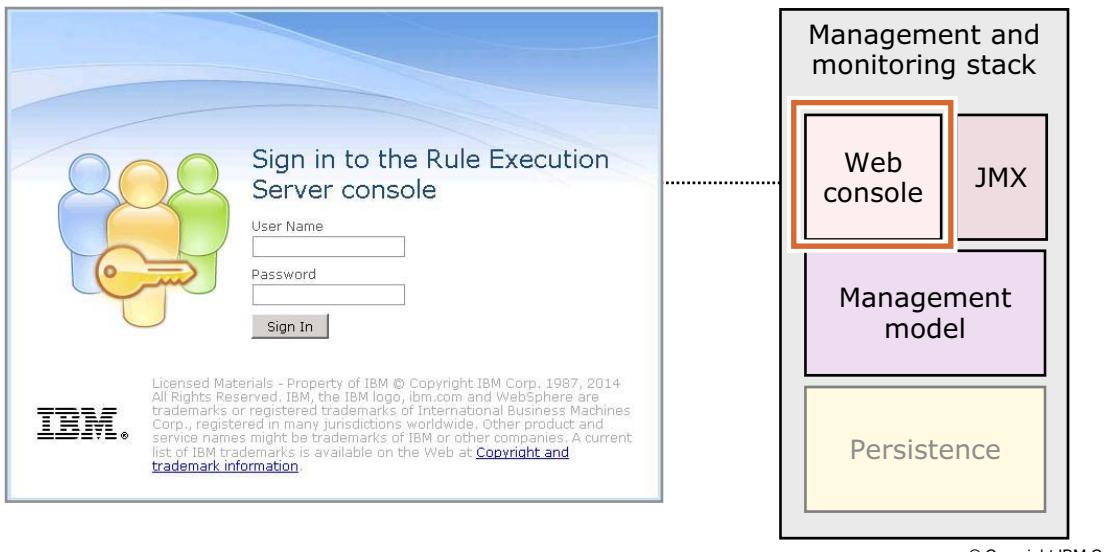
The JMX components are responsible for:

- Hot update and deployment. When a new version of a ruleset is deployed, the JMX layer informs all XUs in the cluster that a new version of the ruleset is available in the database.
- Displaying execution statistics as MBeans Execution statistics so they can then be monitored by using JMX tools like Tivoli.

From the Rule Execution Server console, you can access these MBean attributes and operations through a convenient graphical user interface.

Management and monitoring stack: Console (1 of 2)

- Web-based administration interface
- Application-specific interface to manage business logic, including remote browsing, updating, and deployment of RuleApps
- Central point of the Rule Execution Server architecture



© Copyright IBM Corporation 2015

Figure 6-12. Management and monitoring stack: Console (1 of 2)

WB393 / ZB3931.0

Notes:

Rule Execution Server is accessed through the Rule Execution Server console, which is a web-based administration interface. The Rule Execution Server console is a graphical user interface that depends on your application server. Use it to manage business logic, including remote browsing, updating, and deployment of RuleApps.

The Rule Execution Server console is the central point of the Rule Execution Server architecture. Many features of Rule Execution Server do not work without the Rule Execution Server console.

Management and monitoring stack: Console (2 of 2)

- By default, Rule Execution Server console manages XU instances only if they are running on the same cluster or the same JVM as the console
- To change default behavior, use TCP/IP management mode
 - You can manage XU instances that are running on a different JVM or JMX bean server
 - Example: Managing an XU that handles a desktop application
- Configure the XU to connect to a TCP/IP management server that the Rule Execution Server management EAR file starts
 - Configure the Rule Execution Server EAR to start a TCP/IP management server
 - Configure Java SE or Java EE XU instances so that they connect to the TCP/IP management server that is started

© Copyright IBM Corporation 2015

Figure 6-13. Management and monitoring stack: Console (2 of 2)

WB393 / ZB3931.0

Notes:

In the Rule Execution Server console, you can use the TCP/IP management mode to monitor and manage XU instances, both Java EE and Java SE, outside the JVM or cluster where the console is running.

By setting the TCP/IP management mode, you can enable Rule Execution Server to manage XU instances that are running on a different JVM or JMX MBean server than the Rule Execution Server console.

By default, the Rule Execution Server console cannot manage XUs that handle the execution of Java SE or Java EE rule sessions in the following situations:

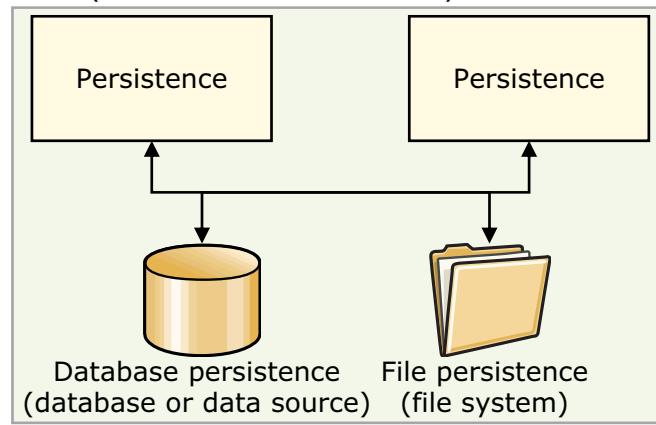
- The XUs are not running in the same cluster as the Rule Execution Server console.
- The XUs are not running in the same Java virtual machine as the Rule Execution Server console.

You can change this behavior by configuring the XU to connect to a TCP/IP management server that is started by the Rule Execution Server management EAR file. A typical use case is when you want to manage a XU that handles a desktop application.

Persistence layer

- Packaged in the management stack and the execution stack
 - Both the management and execution stacks can access the ruleset storage.
 - When a new version of a ruleset is added to the management model, a notification of the update is sent through JMX to the Execution Unit (XU). The XU receives the new resource.
- The persistence layer provides a solution for file persistence (file system) or for database persistence (JDBC or data source).

- Possible values for persistence type:
 - **file** for a file persistence in Java SE.
 - **datasource** for a database persistence in Java EE.
 - **jdbc** for a database persistence in Java SE.



© Copyright IBM Corporation 2015

Figure 6-14. Persistence layer

WB393 / ZB3931.0

Notes:

The persistence layer is packaged in the management stack *and* in the execution stack so that ruleset storage is accessible from both stacks.



Important

The ruleset persistence settings must be the same in the Execution Unit as in the Rule Execution Server console.

The persistence layer includes database persistence components that provide a solution for database persistence that is based on JDBC or a data source. Database persistence is the default solution, as it works in any architecture.

The persistence layer also includes file persistence components to provide a solution for a file persistence, by using the file system. Use this solution on the Java SE platform if you cannot set up a database. You might also use the file persistence solution when you do not want to set up a database, such as when you develop in Rule Designer.

**Warning**

Use database persistence when you deploy to a Java EE cluster environment. If you use file persistence, there is a risk of inconsistency. If, nevertheless, you choose to use file persistence with a clustered Rule Execution Server, you must make sure that all instances have access to a common network file system.



Ant tasks

- Rule Execution Server provides a series of Management Ant tasks and Persistence Ant tasks for automation purposes
- Management Ant tasks:
 - To automate RuleApp deployment and removal in Rule Execution Server
 - To automate Rule Execution Server configuration backup and restoration
- Persistence Ant tasks:
 - To directly access the persistence layer to store or remove rulesets and RuleApps directly in the database, without passing through the Management stack (JMX)

© Copyright IBM Corporation 2015

Figure 6-15. Ant tasks

WB393 / ZB3931.0

Notes:

Rule Execution Server also supports Management Ant tasks and Persistence Ant tasks that you use for automation purposes. You can use Management Ant tasks to automate RuleApp deployment and removal in Rule Execution Server, and Rule Execution Server configuration backup and restoration.

You can use Persistence Ant tasks to access the persistence layer, and store or remove rulesets and RuleApps directly in the database without using the Management stack (that is, the JMX layer). Because of this direct access, if you use Persistence Ant tasks to deploy rulesets or RuleApps, there is no notification, no hot deployment, and the Rule Execution Server console is not aware of this deployment. As a consequence, use these Persistence Ant tasks only if you do not use a Rule Execution Server console or do not care about hot deployment.

6.3. Managed execution in action

Managed execution in action



© Copyright IBM Corporation 2015

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

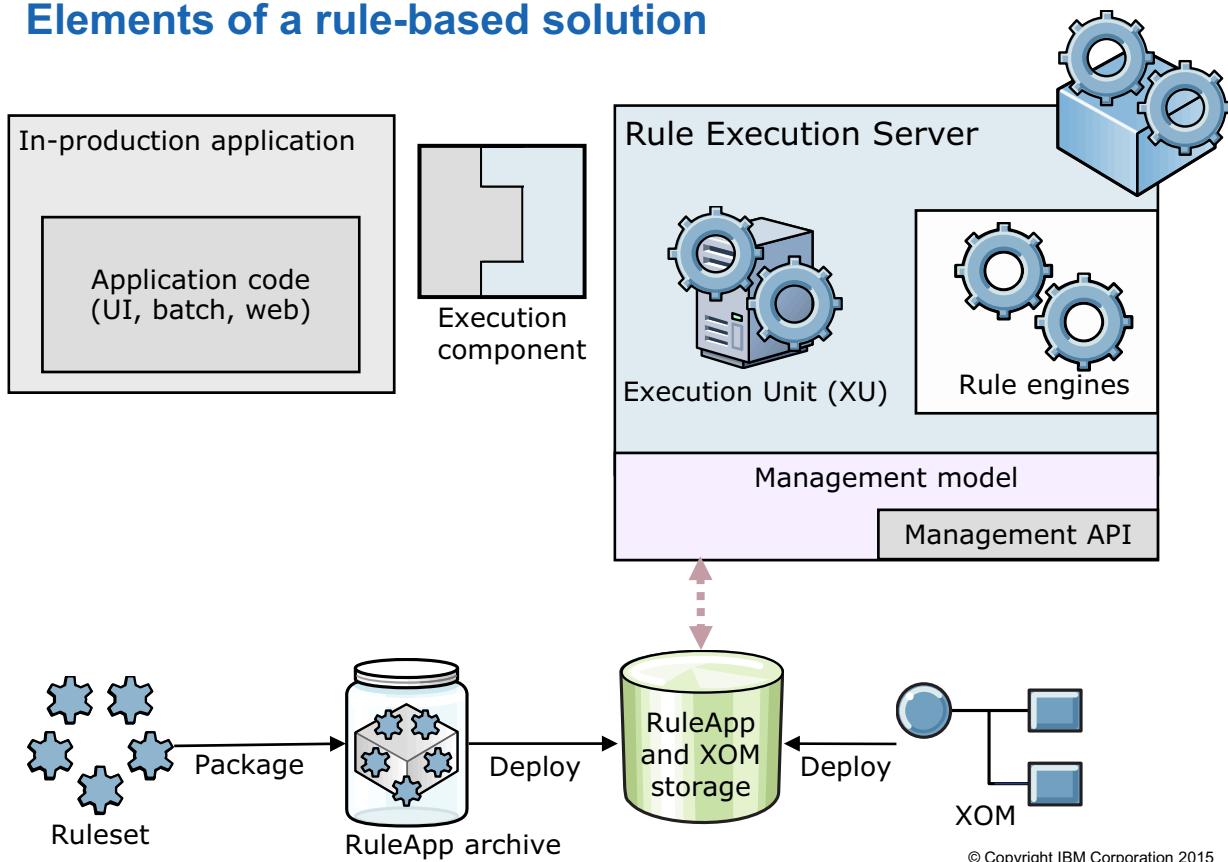
10.1

Figure 6-16. Managed execution in action

WB393 / ZB3931.0

Notes:

Elements of a rule-based solution



© Copyright IBM Corporation 2015

Figure 6-17. Elements of a rule-based solution

WB393 / ZB3931.0

Notes:

The different elements of your business rule solution (XOM, ruleset, RuleApp, rule engine, API, client application, and execution components of Rule Execution Server) are organized in your enterprise as shown here. In the following sections, you learn how these elements dynamically interact in different use cases.



Ruleset parsing

- When requested to execute a ruleset, the XU first retrieves the ruleset from the persistence layer according to the ruleset path, and then parses it
- Ruleset parsing can be *synchronous* or *asynchronous* (default)
- **Asynchronous parsing:** Executes the ruleset in a timely manner
- **Synchronous parsing:** Forces the execution to wait for the latest deployed ruleset version
 - When you deploy a new ruleset version, all executions are suspended until that latest version is parsed

© Copyright IBM Corporation 2015

Figure 6-18. Ruleset parsing

WB393 / ZB3931.0

Notes:

When the client application asks for a ruleset execution, the XU reads the ruleset from the persistence layer by using its ruleset path, loads it, and parses it.

By default, rulesets are parsed asynchronously: If a ruleset is not yet parsed, it can nevertheless be run based on its previous version if it exists in the cache.

- Keep the default asynchronous parsing when the timely execution of a ruleset is paramount and waiting for parsing of a new ruleset is not an option.
- Change the default behavior if you must force ruleset executions to use the latest deployed version of the ruleset. With this action, all executions are suspended when a new version of the ruleset is deployed until that newer version is parsed. Requests to execute an updated ruleset are therefore done only when the new ruleset is parsed and no execution request uses the old ruleset.

You can change the default behavior either by changing the XU deployment descriptor, or developers can use the Rule Execution Server API (method `IlrSessionRequest.setForceUptodate`).



Ruleset execution

- The XU retrieves a rule engine from the pool of available rule engines, and attaches it to the rule session
- The XU feeds the rule engine with the loaded rulesets, ruleset parameters, or objects in working memory
- The XU then requests the ruleset execution by the rule engine
- After ruleset execution, the XU either:
 - Releases the rule engine that is attached to the rule session back to the rule engine pool if the rule session is stateless
 - Keeps the rule engine that is attached to the rule session if the rule session is stateful

© Copyright IBM Corporation 2015

Figure 6-19. Ruleset execution

WB393 / ZB3931.0

Notes:

6.4. Platforms for Rule Execution Server

Platforms for Rule Execution Server



© Copyright IBM Corporation 2015

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

10.1

Figure 6-20. Platforms for Rule Execution Server

WB393 / ZB3931.0

Notes:



Rule execution environment

- Your deployment platform depends on the context of your enterprise:
 - Java SE
 - Java EE
 - Service-oriented architecture (SOA)
- This environment determines:
 - How Rule Execution Server is integrated
 - How the client application requests rule execution

© Copyright IBM Corporation 2015

Figure 6-21. Rule execution environment

WB393 / ZB3931.0

Notes:



Possible choices: Introduction

Rule Execution Server deployment	Server for deployment	Guidelines for client application development
Not deployed in a web server or an application server	(not applicable)	<ul style="list-style-type: none"> • Base client application on Java • Use Rule Execution Server Java SE Rule Session API
Deployed in a web server or an application server	(any)	<ul style="list-style-type: none"> • Request execution of business rules as a decision service • Use hosted transparent decision service or monitored transparent decision service
	Supported application servers	<ul style="list-style-type: none"> • Use Rule Execution Server Java EE API • Can be synchronous or not, local or remote, with or without EJB
	Non-supported application servers, or web servers	<ul style="list-style-type: none"> • Use Rule Execution Server Java SE API

© Copyright IBM Corporation 2015

Figure 6-22. Possible choices: Introduction

WB393 / ZB3931.0

Notes:

This table is a simplified guideline to help determine which platform to use.

In a simple case, such as during testing, you do not want to deploy Rule Execution Server in a web server or in an application server. Base your application on Java, use the Rule Execution Server Java SE Rule Session API, and package Rule Execution Server stacks within your client application. Then, run the client application and Rule Execution Server stacks within the same JVM.

However, in many cases, you deploy Rule Execution Server either in a *web server* (for example, Apache HTTP Server) or in an *application server* (for example, WebSphere Application Server). In that case, your business rule application considers the deployed business rules either as "services", if your approach is SOA-based, or not.

- If the business rule application is *SOA-based*, run your business rules as *decision services*.
- If your business rule application is not SOA-based, and rules are not used as a decision service, you can base your business rule application on Java.

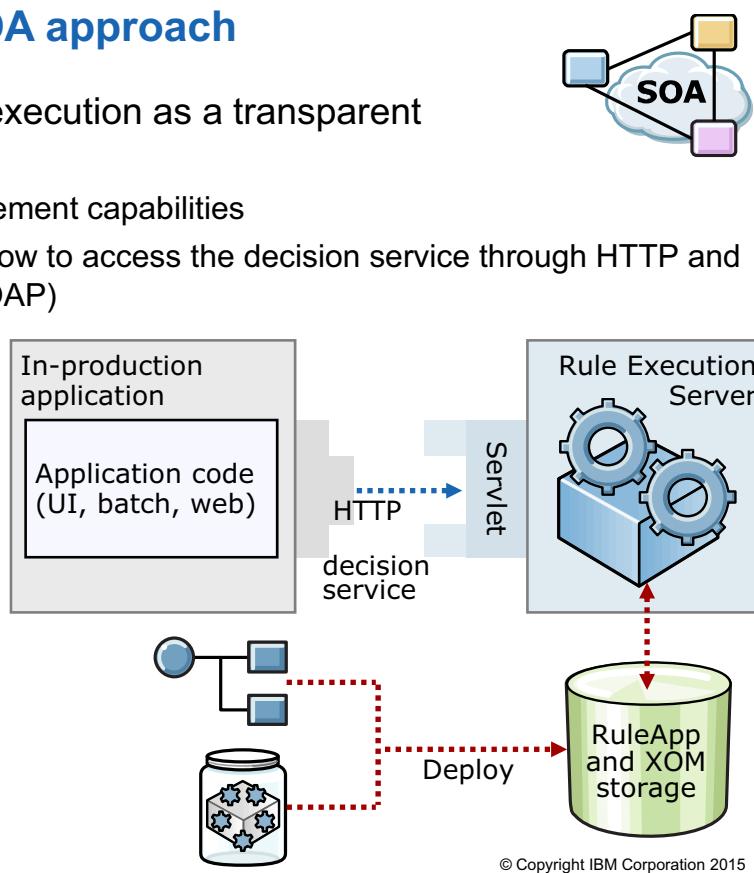
The client application also depends on where Rule Execution Server is deployed.

- To deploy Rule Execution Server on a supported application server (for example WebSphere Application Server), use the appropriate Rule Execution Server *Java EE API* (synchronous or not, local or remote, with or without EJB) in the client application.
- If you want to deploy on a non-supported application server or in a web server (for example, Apache HTTP Server), use Java SE instead. In that case, the client application must use the *Java SE Rule Session API*.

Decision services: SOA approach

- You can expose ruleset execution as a transparent decision service
 - A web service with management capabilities
 - Users need to know only how to access the decision service through HTTP and with XML data formats (SOAP)

- Two types of decision services:
 - Hosted transparent decision service* (HTDS)
 - Monitored transparent decision service* (MTDS)



© Copyright IBM Corporation 2015

Figure 6-23. Decision services: SOA approach

WB393 / ZB3931.0

Notes:

A decision service is technically a web service with management capabilities. It enables users to access Rule Execution Server through a web service, rather than accessing it directly. It is said to be *transparent* because users do not have to know how it is implemented. Users must know how to access the decision service through HTTP and XML data formats (SOAP).

WebSphere Operational Decision Management provides two types of transparent decision services:

- Hosted transparent decision service (HTDS):** An HTDS is essentially a ruleset that is deployed as a web service, which is installed on the same application server as Rule Execution Server.
- Monitored transparent decision service (MTDS):** An MTDS is a service that is generated in Rule Designer and installed on the same application server as Rule Execution Server.



Hosted transparent decision service

- A ruleset that is deployed as a web service, which is installed on the same application server as Rule Execution Server
- An execution component that is linked to a ruleset path with a JMX management bean (MBean)
- Rule Execution Server automatically exposes deployed rulesets as a web (decision) service
- To create an HTDS:
 - Install Rule Execution Server on your web or application server
 - Deploy the HTDS ruleset archive to the same server

© Copyright IBM Corporation 2015

Figure 6-24. Hosted transparent decision service

WB393 / ZB3931.0

Notes:

A hosted transparent decision service is an execution component that is linked to a ruleset path with a JMX managed bean (MBean). To use a hosted transparent decision service, install Rule Execution Server on your web server or application server, and deploy the hosted transparent decision service archive to the same server.

Rule Execution Server automatically exposes any deployed rulesets as a web service, regardless of the underlying type of XOM (Java or dynamic). The decision service automatically generates a *Web Services Description Language (WSDL)* file for each deployed ruleset archive, and the decision service MBean is able to retrieve execution statistics. These rulesets can be exposed as a web service without any code deployment.



Monitored transparent decision service

- Generated with Rule Designer and installed on the same application server as Rule Execution Server
- Exposes management and runtime information about the web service to Rule Execution Server
- To create an MTDS:
 - Create it as a project, by using the Rule Designer Client Project for RuleApps wizard
 - Install it on the web or application server

© Copyright IBM Corporation 2015

Figure 6-25. Monitored transparent decision service

WB393 / ZB3931.0

Notes:

A monitored transparent decision service exposes management and runtime information about the web service to Rule Execution Server. You create a monitored transparent decision as a project by using the Rule Designer Client Project for RuleApps wizard. You then install it on the web or application server.

Monitored transparent decision services can manage rulesets that use any type of XOM. However, with XML, notice that the XML ruleset parameters are handled as String, not as XML objects.

6.5. Managing and monitoring through the Rule Execution Server console

Managing and monitoring through the Rule Execution Server console



© Copyright IBM Corporation 2015

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

10.1

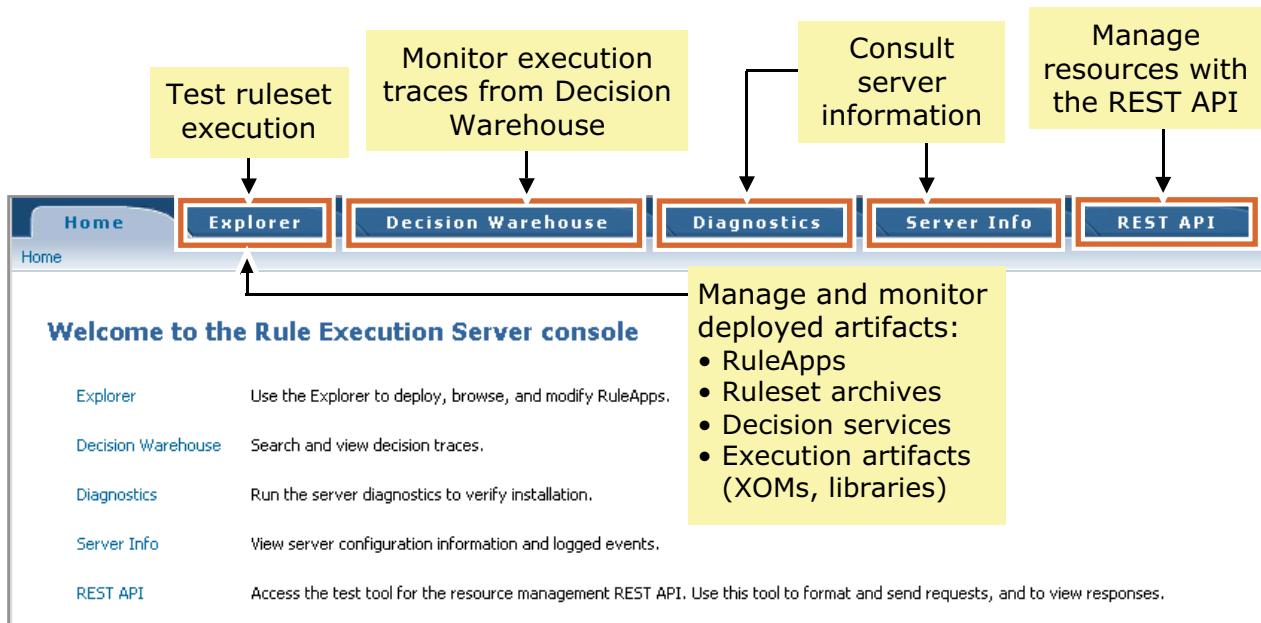
Figure 6-26. Managing and monitoring through the Rule Execution Server console

WB393 / ZB3931.0

Notes:



Rule Execution Server console features



© Copyright IBM Corporation 2015

Figure 6-27. Rule Execution Server console features

WB393 / ZB3931.0

Notes:

The screen capture on this slide shows the various features that you can access through the Rule Execution Server console.

The REST API is covered later in this unit.



System administration through the console

- Rule Execution Server console provides an interface to access MBean attributes and operations:
 - Determine what rulesets are deployed
 - Enable or disable rulesets
 - View event logs
 - Manage the storage and replication of server configuration data
- Make real-time changes to business rules
 - Deploy, change, and manage business rules while the application is running
- The Rule Execution Server database stores all the changes that are made to a ruleset
 - Information about the user who made changes
 - Time of modification
 - Any comments that were added

© Copyright IBM Corporation 2015

Figure 6-28. System administration through the console

WB393 / ZB3931.0

Notes:

Recall that you can use the MBeans accessor API for most of the same management tasks that you can do in the Rule Execution Server console. The server-side API is useful for you to write scripts of system administration operations or if you prefer to work in a command-line environment.

Any action that can be done through the Rule Execution Server console can also be handled as JMX bean methods. However, Rule Execution Server console also provides management access to the Java XOMs, which you cannot manage through the JMX API.



Rule Execution Server administration artifacts: RuleApps

- RuleApps
 - A RuleApp is a deployable management unit that contains rulesets
 - A RuleApp can contain only one ruleset
- A RuleApp keeps a record of the following details:
 - RuleApp name
 - RuleApp version
 - Number of rulesets that the RuleApp manages
 - RuleApp creation date
- You can create RuleApps directly in the Rule Execution Server console and specify the associated rulesets later

© Copyright IBM Corporation 2015

Figure 6-29. Rule Execution Server administration artifacts: RuleApps

WB393 / ZB3931.0

Notes:

Rule Execution Server administration artifacts: RuleApp archives

- A RuleApp archive is an archive that stores RuleApps in a file system
 - The RuleApp descriptor file is named `archive.xml`
 - The descriptor file is stored in the `META-INF` directory of the archive
- The descriptor files contain the basic metadata of a RuleApp:
 - Name, version, and status of the RuleApp
 - Name, version, and status of each ruleset
- Search for resource in a path that begins with parent element in descriptor

© Copyright IBM Corporation 2015

Figure 6-30. Rule Execution Server administration artifacts: RuleApp archives

WB393 / ZB3931.0

Notes:

You search for a resource in a path that begins with the parent element in the descriptor.

- The RuleApp path is defined by the name and the version number of the RuleApp.
 - No resources are associated with a RuleApp.
- The Ruleset path is defined by the name and the version number of the RuleApp, followed by the name and the version number of the ruleset.
 - This path is called the *canonical ruleset path*.

Here are some examples of valid ruleset paths:

- `/rule8_8app/6_ruleset_6/1.0`
- `/rule8_8app/1.0/6_ruleset_6`
- `/rule8_8app/6_ruleset_6`
- `/rule8_8app/1.0/6_ruleset_6/1.0`

**Important**

RuleApp archives are saved in a strict directory structure. If you change that directory structure, you invalidate the RuleApp.



Rule Execution Server administration artifacts: Rulesets

- You can add a ruleset to a RuleApp in one of the following ways:
 - Package the ruleset within a RuleApp through the Rule Execution Server console
 - Deploy the ruleset within a RuleApp file
- A ruleset is deployed to Rule Execution Server, along with a ruleset archive

© Copyright IBM Corporation 2015

Figure 6-31. Rule Execution Server administration artifacts: Rulesets

WB393 / ZB3931.0

Notes:

A classic rule engine archive contains the following directories and files:

- A directory for each package in the ruleset, with a `.irl` file for each rule in the package
- An `.irl` file for each rule task in the ruleset
- A `.rf1` file for each ruleflow in the ruleset
- A **META-INF** directory that contains:
 - An `archive.xml` file
 - The ruleset signature
- A **RESOURCES** directory, which can contain:
 - A `metadata.xml` file, which includes the hierarchical properties and the property type declaration in XML format
 - A `bominfo.xml` file, which includes the BOM paths
 - BOM-to-XOM mapping files

- XOM files
- BOM files
- The `engine.conf` configuration file

Decision engine archives contain the result of a further compilation of the rules:

- Java bytecode when you compile from Rule Designer and select the **Optimize ruleset loading (Java bytecode generation)** option on the “Export a Ruleset Archive” page.
- Rules and ruleflows in an intermediate representation when you clear the **Optimize ruleset loading (Java bytecode generation)** option or when you generate rulesets from Decision Center.



Back up and restore

- Backing up deployed RuleApps
 - Save deployed RuleApps for security purposes or when your test server and your production server are not connected to the same network
 - A backup saves all the RuleApps that are deployed on Rule Execution Server
- Restoring a previous server configuration
 - You can use the Rule Execution Server console to restore a server configuration to an installed Rule Execution Server
 - If you restore previously downloaded RuleApps, the restored RuleApps erase the currently deployed RuleApps

© Copyright IBM Corporation 2015

Figure 6-32. Back up and restore

WB393 / ZB3931.0

Notes:

6.6. Managing resources with the REST API

Managing resources with the REST API



© Copyright IBM Corporation 2015

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

10.1

Figure 6-33. Managing resources with the REST API

WB393 / ZB3931.0

Notes:



REST API for ruleset execution

- Rule Execution Server provides a Representational State Transfer (REST) API for ruleset execution
 - Provides XML and JSON generation, XSD validation, and execution services
 - Execute rulesets with XML or JSON format through the HTTP protocol
- The REST service for ruleset execution provides these benefits:
 - Do not need a client library or a complex configuration to interact with a remote Rule Execution Server instance
 - Work across platforms or from various client applications
 - Easy to switch from local to remote Rule Execution Server execution
- You can test REST API ruleset execution requests in Rule Execution Server console

© Copyright IBM Corporation 2015

Figure 6-34. REST API for ruleset execution

WB393 / ZB3931.0

Notes:

You can use the REST API to execute rulesets through the HTTP protocol by using the XML format.

The REST services include:

- XML payload sample generation
- XML and XSD validation
- Ruleset execution services

With these REST services, you do not need to add any client libraries or perform complex configuration, which is typically required with the JMX API.

By using the REST API, you can work across environments or from other client applications, such as JavaScript clients. Also, you can easily switch from local execution to a remote Rule Execution Server instance.

To verify that your ruleset execution requests use well-formatted XML, you can test the REST API requests in the Rule Execution Server console. You can also use REST API to integrate ruleset execution with other BPM products.



Endpoint URLs

- The REST API includes URIs that represent the rulesets

- URI format:

`http://host:port/DecisionService/rest/v1/rulesetPath?options`

- `host:port/DecisionService`: Host address and port for the HTDS application
- `/rest/v1`: Context root and version of the REST service
- `rulesetpath`: Can be either canonical or non-canonical
- `options`: Options for generation of WADL file
 - `inline`: Generates and displays a WADL file with the XSD schema included
 - `zip`: Generates a compressed file for download

© Copyright IBM Corporation 2015

Figure 6-35. Endpoint URIs

WB393 / ZB3931.0

Notes:

The endpoint URLs represent the rulesets, which are the Rule Execution Server resources for the REST execution service.

The first part of the URI consists of the host address and port of the hosted transparent decision service (HTDS) application.

The next part of the URI is `/rest/v1`. The `rest` represents the REST service context root. The `v1` indicates the version number of the REST service.

The third part of the URI is `rulesetPath`, where you should see the short ruleset path or the canonical ruleset path. A canonical ruleset path includes the RuleApp name and version number, followed by the name and version number of the ruleset. Here is an example of a canonical ruleset path: `myRuleApp/1.0/myRuleset/1.0`

A short ruleset path leaves out one or both version numbers, and includes only the names of the rule application and ruleset.

The last part of the URI is `options`, which generates a WADL file. Valid values for `options` are either the `inline` parameter or the `zip` parameter. If you use the `inline` option, the XSD schema

is included in the WADL file, and you can see it right away. Otherwise, you can use the `zip` option to download the compressed file and view it locally.

HTTP methods

- **GET** method
 - Generate a sample XML or JSON payload
 - Retrieve the WADL for a specified ruleset
- **POST** method
 - Create an execution request
 - Validate an XML payload
- HTTP header fields
 - The **Accept-Language** header field is equivalent to the **accept-language** URI parameter for a list of languages that are valid in response message
 - The **Content-Type** field accepts both **application/xml** and **application/json** content types

© Copyright IBM Corporation 2015

Figure 6-36. HTTP methods

WB393 / ZB3931.0

Notes:

The REST API supports HTTP GET and POST methods. You use the GET method to generate a sample XML or JSON payload or retrieve the WADL representation for the ruleset.

You use the POST method to validate an XML payload and to execute the ruleset. The request body for the execution request contains the XML or JSON payload. If the request body contains XML, it is a good practice to validate the XML first before sending the request.

For the HTTP header, the **Accept-Language** field is the same as the **accept-language** URI parameter. It takes a list of languages as valid value.

The **Content-Type** field accepts **application/xml** and **application/json**. XML is the default content type of the response. You can also find the schema definitions by retrieving the WADL file.



Request and response schema

- Request contains:
 - The ruleset IN and INOUT parameters in alphabetical order
 - A decision ID (optional)
 - A trace filter (optional)
- Execution response contains:
 - The ruleset INOUT and OUT parameters, in alphabetical order
 - The decision ID
 - The trace, if specified in the request
- Validation responses are returned in JSON format
 - Responses to valid requests contains an empty JSON list []
 - Responses to invalid requests describe the type and location of the error
- Example of validation response to invalid request:


```
{"type": "Error", "line": 9, "column": 22, "message": "cvc-datatype-valid.1.2.1: '5d' is not a valid value for 'integer'.")}
```

© Copyright IBM Corporation 2015

Figure 6-37. Request and response schema

WB393 / ZB3931.0

Notes:

In the ruleset execution REST service, requests and responses follow different schemas, depending on whether the ruleset XOM is based on XML classes or on Java classes. The schema determines how the types are serialized.

With the ruleset signature, the request part is composed of the following elements:

- The IN and INOUT parameters of the ruleset, in alphabetical order.
- An optional decision ID if you want to set it to a specific value and an optional trace filter.

The request can be sent in either the XML or JSON format.

The execution response consists of:

- The INOUT and OUT parameters of the ruleset, in alphabetical order.
- The decision ID, either the default value or the value that you set in the request.
- If you set trace filter in request, the trace is returned in response.

Notice that the execution response is sent in the same format as the execution request (either XML or JSON). The XML payload is analyzed against the generated XSD files.

The validation response is returned in JSON format.

- If the request is valid, the response is an empty JSON list [].
- If the request is invalid, the tool returns the list of errors.
- Each response to an invalid request includes:
 - Type: Error, Fatal, or Warning
 - Line: The number of the line that contains the error in the XML file
 - Column: The column number that contains the error in the XML file
 - Message: Description of error

The example on the slide shows you that there is an error in line 9, column 22. Based on the XML schema data type validation rule, value “5d” is not a valid integer.

WADL representation

- You generate WADL representation to write the XML payload of a specific ruleset execution request
- URL: `http://{host}:{port}/DecisionService/rest/v1/{rulesetPath}/wadl`
- A set of XSD files are also generated, but separated from WADL file
- Optional parameters:
 - `inline`: the `.xsd` files are included in the `.wadl` file
 - `zip`: the `.xsd` files and `.wadl` file are generated in a compressed file

Response result:

- The WADL file
 - One GET method function: XML payload sample generation
 - Two POST method functions: XML payload validation and ruleset execution
- XSD files to describe XML representation of the input and output objects

© Copyright IBM Corporation 2015

Figure 6-38. WADL representation

WB393 / ZB3931.0

Notes:

The Web Application Description Language (WADL) representation contains all the information of the request and response elements. You can use the WADL representation as a reference while you develop your client application.

You can generate the WADL file through a REST request. By default, a set of XSD files are generated and they are separated from WADL file. If you specify the `inline` option, the XSD file contents are included in the WADL file. Alternatively, to get WADL code and its XSD files to a compressed file, add the `zip` parameter.

If it is a valid request, you get a WADL representation of the request and response documentation. It contains one GET method function for XML payload sample generation, and two POST method functions for XML payload validation and ruleset execution. The attached XSD files describe the XML representation of the input and output objects.



REST API management services

- Use the **REST API** tab in the Rule Execution Server console to manage REST resources through HTTP methods

The screenshot shows the REST API tab selected in the top navigation bar. Below it, the title "REST API tool" is displayed. A note says: "You can use this test tool to test the REST API for resource management. To test the REST API for remote ruleset execution, click the Explorer tab." It also mentions the REST API WADL file: </res/api/auth/v1/DecisionServer.wadl>. The main area lists four endpoints under the "/ruleapps" path, each with a method (GET or POST), URL, and a brief description:

- GET /ruleapps**: getRuleApps Returns all the RuleApps contained in the repository.
- GET /ruleapps?count=true**: getCountOfRuleApps counts the number of elements in this list.
- POST /ruleapps**: deployRuleAppArchive Deploys a RuleApp archive in the repository, based on the merging and versioning policies passed as parameters. The RuleApp archive
- POST /ruleapps**: addRuleApp Adds a new RuleApp in the repository. The RuleApp representation is passed in the request body in JSON or XML format. If the RuleApp already exists, specific error description and the HTTP status 202 is returned.

© Copyright IBM Corporation 2015

Figure 6-39. REST API management services

WB393 / ZB3931.0

Notes:

From the **REST API** tab, you can manage resources through HTTP methods.

During the exercise, you learn how to create deployment requests and how to access and manage REST resources.



Unit summary

Having completed this unit, you should be able to:

- Describe the Rule Execution Server architecture
- Use the Rule Execution Server console
- Explain how to access and manage resources by using the REST API

© Copyright IBM Corporation 2015

Figure 6-40. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** Rule Execution Server architecture is composed of a management stack and an execution stack.
2. **True or False:** The Execution Unit (XU) uses the execution components to execute a ruleset.
3. **True or False:** You can manage RuleApp and XOM deployment through the Rule Execution Server console.

© Copyright IBM Corporation 2015

Figure 6-41. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. True.

2. True.

3. True.

© Copyright IBM Corporation 2015

Figure 6-42. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 8



Exploring the Rule Execution Server
console

© Copyright IBM Corporation 2015

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

10.1

Figure 6-43. Exercise 8

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Work with Rule Execution Server console tools
- Manage RuleApps and rulesets through the Rule Execution Server console

© Copyright IBM Corporation 2015

Figure 6-44. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 7. Using Decision Warehouse to audit ruleset execution

What this unit is about

In this unit, you learn how to monitor and audit ruleset execution through Decision Warehouse.

What you should be able to do

After completing this unit, you should be able to:

- Enable ruleset monitoring
- Audit the execution of rulesets with Decision Warehouse
- Optimize Decision Warehouse

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Enable ruleset monitoring
- Audit the execution of rulesets with Decision Warehouse
- Optimize Decision Warehouse

© Copyright IBM Corporation 2015

Figure 7-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Traces
- Auditing ruleset execution through Decision Warehouse
- Optimizing Decision Warehouse
- Customizing Decision Warehouse
- Troubleshooting

© Copyright IBM Corporation 2015

Figure 7-2. Topics

WB393 / ZB3931.0

Notes:

7.1. Traces

Traces



© Copyright IBM Corporation 2015

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

10.1

Figure 7-3. Traces

WB393 / ZB3931.0

Notes:

Recall: ODM administrator

- Responsibilities:
 - Deploying and configuring the server and database for Decision Center and Rule Execution Server
 - Managing user access to Decision Center and Rule Execution Server
 - Configuring trace data sources for testing purposes
 - Deploying applications
 - Redeploying rulesets and event assets as changes are made
 - Generating detailed execution reports
 - Tracking and monitoring rule execution
 - Restoring a particular application state
- Tools: Servers for Decision Center or runtime environments



Administrator

© Copyright IBM Corporation 2015

Figure 7-4. Recall: ODM administrator

WB393 / ZB3931.0

Notes:



Traces

- When you execute your business rule application, your application generates different types of traces
 - **Log traces:** Traces that are generated in the log file
 - **Output traces:** Traces that business rules print in the standard output file
 - **Decision traces:** Data that the rule engine generates to trace how the ruleset executes
- To capture all **decision traces** for rule execution, use Decision Warehouse
- To capture all decision traces for rules that are executed with an execution mode other than RetePlus:
 - Set the `ruleset.sequential.trace.enabled` property to true on your ruleset

© Copyright IBM Corporation 2015

Figure 7-5. Traces

WB393 / ZB3931.0

Notes:

When you run your business rule application, your application generates different types of traces:

Log traces

Ruleset log traces are generated in the log file, for example, by running instructions like `System.out.println` in your client application.

You cannot retrieve the log traces programmatically.

Output traces

Ruleset output traces are generated when actions of business rules are executed that print text in the standard output file. For example, an action of a business rule can print a text in the standard output file by running the BAL keyword `print`.

Developers can retrieve the ruleset output traces programmatically, by using `IlrSessionResponse.getRulesetExecutionOutput`. This call returns a `String`.

Decision traces

Ruleset decision traces are data that the rule engine generates to describe how the ruleset executes.

With decision traces, your client application can retrieve the duration of the ruleset execution, the number of rules that are fired or not fired, and other information.

You can retrieve the ruleset output traces programmatically, as you see in the slide.

7.2. Auditing ruleset execution through Decision Warehouse

Auditing ruleset execution through Decision Warehouse



© Copyright IBM Corporation 2015

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

10.1

Figure 7-6. Auditing ruleset execution through Decision Warehouse

WB393 / ZB3931.0

Notes:

Auditing ruleset execution

- Audit decision traces to get details of rule execution
 - Troubleshooting rule execution behavior
 - Governance
- Decision traces contain information about how a decision was made
 - The executed ruleflow and the path that is taken through the ruleflow tasks
 - List of rules that were executed
 - These details are intended to help users, such as an auditor, understand what happened as a result of executing a ruleset
- To generate decision traces, you must enable ruleset monitoring
 - When the ruleset executes, Rule Execution Server generates a decision trace and saves it through Decision Warehouse
 - Decision Warehouse stores decision traces in a database

© Copyright IBM Corporation 2015

Figure 7-7. Auditing ruleset execution

WB393 / ZB3931.0

Notes:

When rules are used in applications to take decisions that are based on real data, the details about the executed rules can help auditors understand what happens during ruleset execution.

To audit rule execution, you use Decision Warehouse.



Rule Execution Server console: Decision Warehouse

- Tool for monitoring ruleset execution
 - Used to store, filter, and view the results of rule execution
- Accessed through the Rule Execution Server console

A screenshot of the Rule Execution Server console interface. At the top, there is a navigation bar with tabs: Home, Explorer, Decision Warehouse (which is highlighted with an orange border), Diagnostics, Server Info, and REST API. Below the navigation bar, the main content area has a title "Decision Warehouse". On the left, a sidebar titled "Select a task:" lists options: Search Decisions, Persistence Properties, and Clear Decisions. The main content area is titled "Search Decisions" and contains a form with fields for "Executed ruleset path:", "Decision ID:", "Rules Fired and Tasks Executed:", "Input parameters:", and "Output parameters:". Each input field has a small ellipsis button (...) to its right. At the bottom right of the content area, it says "© Copyright IBM Corporation 2015".

Figure 7-8. Rule Execution Server console: Decision Warehouse

WB393 / ZB3931.0

Notes:

Decision Warehouse is available as a tab in the Rule Execution Server console.

Decision Warehouse (1 of 2)

- Decision Warehouse provides a means to store, filter, and view rule execution activity
- When you enable ruleset monitoring, Rule Execution Server generates ruleset decision traces behind the scene, and saves them in Decision Warehouse
- Decision Warehouse stores decision traces in a database
- A trace contains information about how a decision was made: the executed ruleflow, the path of executed ruleflow tasks, and the rules that were fired

© Copyright IBM Corporation 2015

Figure 7-9. Decision Warehouse (1 of 2)

WB393 / ZB3931.0

Notes:

Through Decision Warehouse, users can view stored ruleset decision traces.

When you enable ruleset monitoring, Rule Execution Server generates ruleset decision traces behind the scene, and saves them in Decision Warehouse.

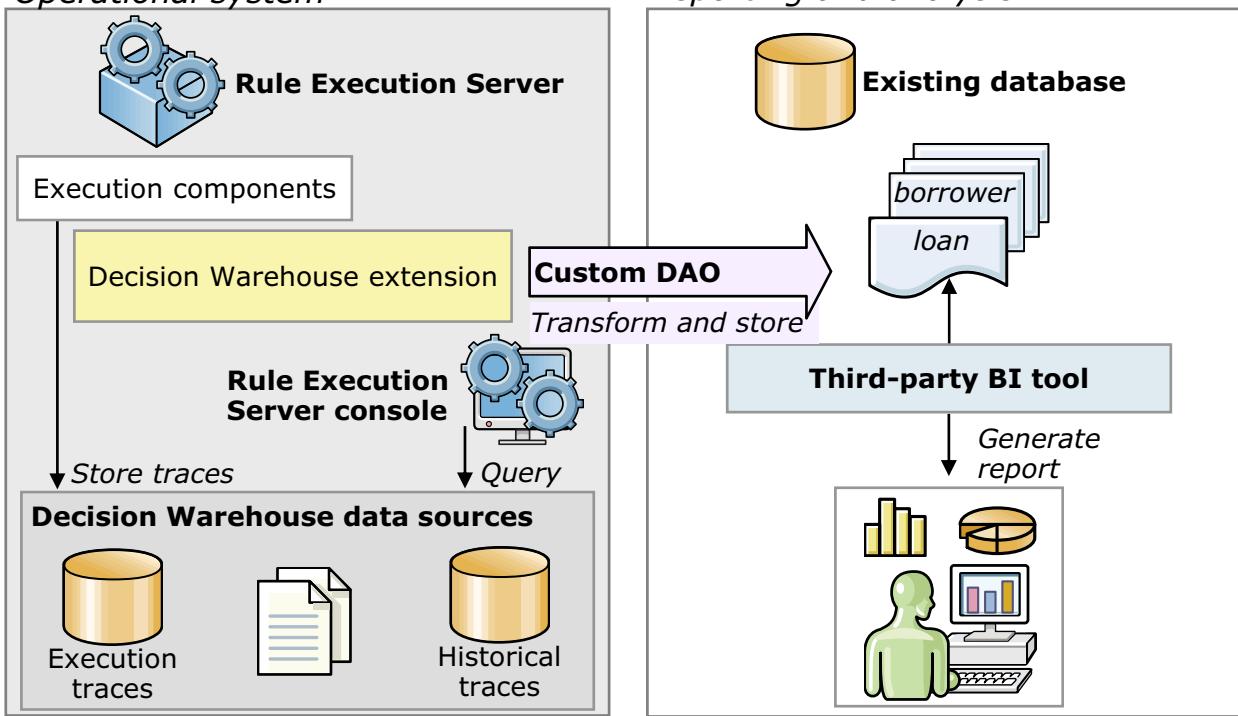
Decision Warehouse stores decision traces in a database. The trace contains information about how a decision was made. It records the executed ruleflow, the path of executed ruleflow tasks, and the rules fired. These details are intended to help users understand what happened as a result of executing a ruleset.

For more information about limitations and the use of Decision Warehouse in the Rule Execution Server console, see these sections in the product documentation:

- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.dserver.rules.res.managing/topics/con_res_dw_overview.html
- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.dserver.rules.res.console/topics/tpc_rescons_dw_intro.html

Decision Warehouse (2 of 2)

*Enterprise applications
Operational system*



© Copyright IBM Corporation 2015

Figure 7-10. Decision Warehouse (2 of 2)

WB393 / ZB3931.0

Notes:

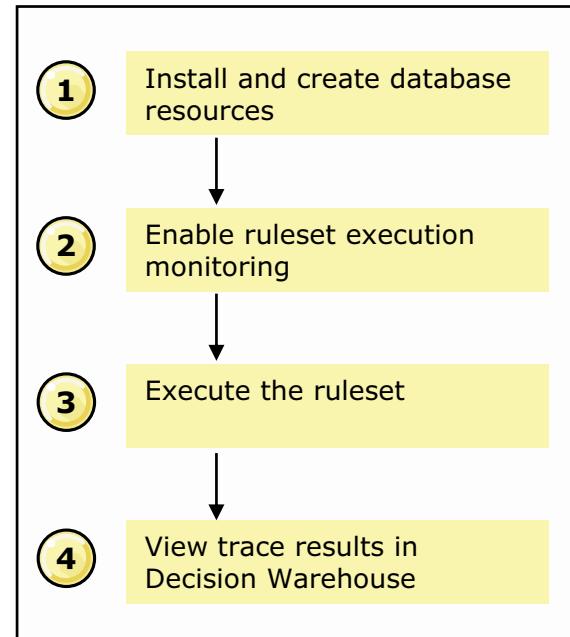
This figure shows Decision Warehouse in an operational system that uses an extension point to customize how the trace data is stored.

Information is sent through a Data Access Object (DAO) to an existing database. Business users can then generate reports from that data and use the business reports to measure business performance.

By default, Decision Warehouse stores trace data in the Rule Execution Server database. Later, you learn more about customizing these data sources and optimizing the data that Decision Warehouse stores.

Step 1: Default use of Decision Warehouse

- Creation of database resources is completed during installation
 - Default data source for Rule Execution Server is used for Decision Warehouse
 - Database schemas are created through Installation Settings wizard



© Copyright IBM Corporation 2015

Figure 7-11. Step 1: Default use of Decision Warehouse

WB393 / ZB3931.0

Notes:

You saw how to install and create database resources during exercise Unit 2, "Configuring Operational Decision Manager". You complete the installation of Rule Execution Server and Decision Warehouse through the Installation Settings wizard in the Rule Execution console.



Step 2: Enable rule execution monitoring

- To capture traces, enable monitoring at the ruleset level
 - After traces are enabled, Rule Execution Server generates ruleset decision traces behind the scene and stores them through Decision Warehouse
- Enable monitoring with these ruleset properties:
 - `monitoring.enabled`
 - `ruleset.trace.enabled` (if the ruleset contains rule tasks, use the RetePlus execution mode)
 - `ruleset.sequential.trace.enabled` (if the ruleset contains rule tasks, use the sequential or Fastpath execution modes)
 - `ruleset.bom.enabled` (optional)

© Copyright IBM Corporation 2015

Figure 7-12. Step 2: Enable rule execution monitoring

WB393 / ZB3931.0

Notes:

Enabling decision traces is done at the ruleset level. The monitoring properties can be added to the ruleset either before or after deployment. As an administrator, you can set these properties through the Rule Execution Server console.

To configure a ruleset for monitoring, you add the following ruleset properties to the ruleset, and set their values to true:

- `monitoring.enabled`
- `ruleset.bom.enabled`
 - This property is optional. If it is not defined, Decision Warehouse acts as if it is defined and its value is `false`.
 - If this property is defined and its value is `true`, the input and output data is serialized and stored in a BOM representation of the data in XML format.

When this property is defined and its value is `false`:

- If the ruleset is based on a dynamic XOM, the input and output parameters are stored as XML code.

- If the ruleset is based on a Java XOM, the `toString` method of the ruleset parameter type stores the content.
- `ruleset.sequential.trace.enabled`
 - This property is required only if the ruleset contains rule tasks that use sequential or Fastpath execution modes.

You can set these ruleset properties in:

- Rule Designer and Decision Center Enterprise console before you deploy a ruleset.
- The Rule Execution Server console after you deploy the ruleset.

These monitoring properties can be added to the ruleset either before or after deployment. As an administrator, you can set these properties through the Rule Execution Server console. You saw how to manage (add, edit, remove) ruleset properties in Rule Designer and in the Decision Center Enterprise console.



Step 3: Execute the ruleset

- Execute your ruleset by using:
 - Decision Validation Services
 - A client application
 - A hosted transparent decision service

© Copyright IBM Corporation 2015

Figure 7-13. Step 3: Execute the ruleset

WB393 / ZB3931.0

Notes:

To generate trace data, you must execute the ruleset with a client application, or through testing with Decision Validation Services, or by using a decision service. The generated trace data is written to the default Decision Warehouse database.



Step 4a: View stored decision traces

- Open the **Decision Warehouse** tab in the Rule Execution Server console
 - You must be logged in with the `resMonitor` or `resAdmin` role
- Search for the latest ruleset execution traces

65 Decision(s) found						Display by <input type="button" value="10"/>
Decision ID	Date	Ruleset Version	Number of rules fired	Decision Trace	Processing Time (ms)	
5ab23322-97ef-4787-b892-58166e39cebc	2009-03-26 18:02:58	/prodra1238086969350_90b385fe_4e4c_4842_b408_4d1e9dcadfd6 /1.0/prodr1238086969350_90b385fe_4e4c_4842_b408_4d1e9dcadfd6/1.0	1	View Decision details	416	
50a3601b-20fa-4f78-abf7-6935f6d4516c	2009-03-19 15:30:10	/prodra1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945 /1.0/prodr1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945/1.0	1	View Decision details	35	
6c44f221-5d0e-486f-89d9-ceab48042d84	2009-03-19 15:30:10	/prodra1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945 /1.0/prodr1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945/1.0	0	View Decision details	37	
a18b5329-c990-4ea5-9a37-fb01e0b1e3e6	2009-03-19 15:30:08	/prodra1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945 /1.0/prodr1237473006170_519ba66c_0ad6_4142_a3e9_3d7df42c7945/1.0	1	View Decision details	205	
385ee32d-6bc8-49b1-a648-cd51c0cf07d	2009-03-19 15:27:48	/prodra1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e /1.0/prodr1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e/1.0	1	View Decision details	172	
6602fed-6c46-4fe5-8bc8-f10839340a28	2009-03-19 15:27:48	/prodra1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e /1.0/prodr1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e/1.0	0	View Decision details	47	
41565ac8-2a56-47c4-b5ce-e8d925c9fa	2009-03-19 15:27:47	/prodra1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e /1.0/prodr1237472858960_f57cfa85_481b_4e0e_a50a_9bd94a70e73e/1.0	1	View Decision details	505	
57705e85-ff38-4b53-a716-eb0a45fcbe76	2009-03-13 11:57:58	/prodra1236941862805_eee830e0_cefa_4105_a0be_7851d0bca552 /1.0/prodr1236941862805_eee830e0_cefa_4105_a0be_7851d0bca552/1.0	4	View Decision details	1138	
64efc105-58cd-4e71-8350-e550b957a610	2009-03-06 19:22:52	/prodra1236363766960_fad3c26c_ab69_46d9_9763_8d4ca0a422fc /1.0/prodr1236363766960_fad3c26c_ab69_46d9_9763_8d4ca0a422fc/1.0	3	View Decision details	18	
f91fe910-2585-4d18-987a-dbc9e06b4f77	2009-03-06 19:22:52	/prodra1236363766960_fad3c26c_ab69_46d9_9763_8d4ca0a422fc /1.0/prodr1236363766960_fad3c26c_ab69_46d9_9763_8d4ca0a422fc/1.0	3	View Decision details	13	

1 - 10 out of 65 results

© Copyright IBM Corporation 2015

Figure 7-14. Step 4a: View stored decision traces

WB393 / ZB3931.0

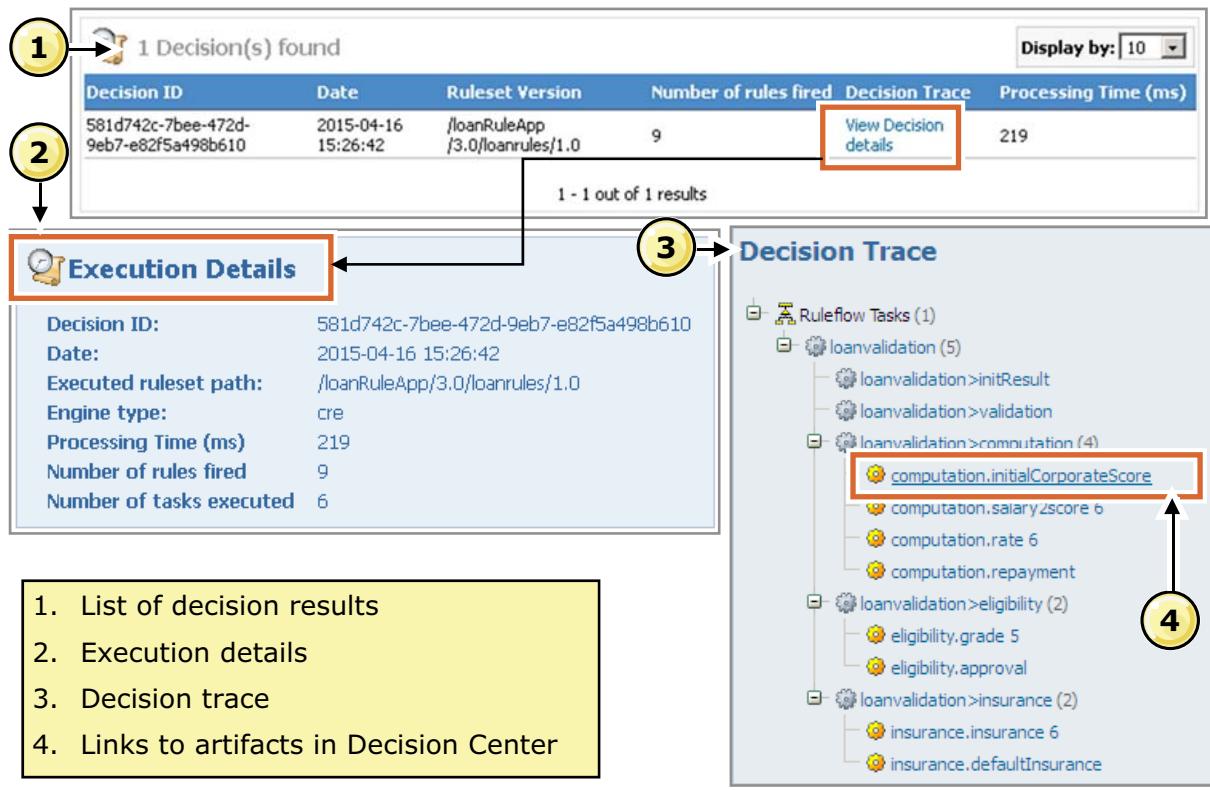
Notes:

After your decision traces are stored in Decision Warehouse, you can use the **Decision Warehouse** tab of the Rule Execution Server console to query Decision Warehouse and view the stored decision traces.

With the query feature, you can define filters on the specified data source so that only the events or decisions you are interested in are shown. You can use the Decision ID to locate a specific transaction and view the executed rules. A Decision ID is automatically generated by default, and is equal to the ID of the Execution Unit (XU) connection.

 WebSphere Education 

Step 4b: View decision details and fired rules



1 Decision(s) found

Decision ID	Date	Ruleset Version	Number of rules fired	Decision Trace	Processing Time (ms)
581d742c-7bee-472d-9eb7-e82f5a498b610	2015-04-16 15:26:42	/loanRuleApp/3.0/loanrules/1.0	9	View Decision details	219

1 - 1 out of 1 results

Execution Details

Decision ID: 581d742c-7bee-472d-9eb7-e82f5a498b610
 Date: 2015-04-16 15:26:42
 Executed ruleset path: /loanRuleApp/3.0/loanrules/1.0
 Engine type: cre
 Processing Time (ms): 219
 Number of rules fired: 9
 Number of tasks executed: 6

Decision Trace

- Ruleflow Tasks (1)
 - loanvalidation (5)
 - loanvalidation>initResult
 - loanvalidation>validation
 - loanvalidation>commutation (4)
 - [computation.initialCorporateScore](#)
 - computation.salary_zscore 6
 - computation.rate 6
 - computation.repayment
 - loanvalidation>eligibility (2)
 - eligibility.grade 5
 - eligibility.approval
 - loanvalidation>insurance (2)
 - insurance.insurance 6
 - insurance.defaultInsurance

1. List of decision results
2. Execution details
3. Decision trace
4. Links to artifacts in Decision Center

© Copyright IBM Corporation 2015

Figure 7-15. Step 4b: View decision details and fired rules

WB393 / ZB3931.0

Notes:

Explanation of legend:

1. For each decision trace, you can click the **View Decision details** link to access the information details such as which rule tasks and rules were fired, and the input and output parameter values.
2. The Execution Details pane gives overview information about the taken decision.
3. The Decision Trace pane gives the list of ruleflow tasks and rule artifacts that were executed.
4. If you deploy the RuleApp from Decision Center, the Decision Trace section contains links to the rule artifacts in Decision Center repository. You can click these links to open the Decision Center Enterprise console and view the corresponding rule artifact.

7.3. Optimizing Decision Warehouse

Optimizing Decision Warehouse



© Copyright IBM Corporation 2015

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

10.1

Figure 7-16. Optimizing Decision Warehouse

WB393 / ZB3931.0

Notes:

Ruleset monitoring options

- To enable ruleset monitoring, you set predefined properties on the ruleset
 - You can define properties on a ruleset in Rule Designer, Decision Center, or Rule Execution Server console
 - Example ruleset properties on RuleApp deployed from Decision Center

<input type="checkbox"/> Select All	Name	Value
<input type="checkbox"/>	ilog.rules.teamserver.baseline	current
<input type="checkbox"/>	ilog.rules.teamserver.permalink.project	http://localhost:9080/teamserver/faces/home.jsp?project=loan-rules&baseline=current&datasource=jdbc%2FilogDataSource
<input type="checkbox"/>	ilog.rules.teamserver.permalink.report	http://localhost:9080/teamserver/faces/servlet/ReportingServlet?project=loan-rules&baseline=true&datasource=jdbc%2FilogDataSource&locale=en_US
<input checked="" type="checkbox"/>	monitoring.enabled	true
<input type="checkbox"/>	ruleset.bom.enabled	true
<input type="checkbox"/>	rulesetSEQUENTIAL.trace.enabled	true

© Copyright IBM Corporation 2015

Figure 7-17. Ruleset monitoring options

WB393 / ZB3931.0

Notes:

You can manage ruleset properties through the Rule Execution Server console. As shown here, you can view which properties are defined on a ruleset in the **Show-Hide Properties** pane. In the **monitoring options** section, you can modify the definition of the monitoring properties.

To manage monitoring properties in the Rule Execution Server console:

- Click the **Explorer** tab.
- In the Navigator pane, click the relevant ruleset.
- In the Ruleset View, click **Show Monitoring Options**.
- At the bottom of the monitoring options box, click the **Edit** icon.
- Select the **Enable tracing in the Decision Warehouse** option.
- Select the appropriate options (that you see next)

The options that you select modify the ruleset properties, as you see next.

WebSphere Education

Filtering trace data (1 of 2)

Hide Monitoring Options

monitoring options

Enable tracing in the Decision Warehouse

Store the values of the ruleset parameters to:

BOM format with optional filter:

Native format

Select the execution traces to store in the Decision Warehouse:

- Execution Date
- Execution Duration
- Total Number of Tasks Executed
- Total Number of Tasks Not Executed
- Total Number of Rules Fired
- Total Number of Rules Not Fired
- Execution Events
- List of All Tasks
- List of Tasks Not Executed
- List of All Rules
- List of Rules Not Fired
- Bound Object by Rule
- System Properties
- Working Memory with optional filter:

Save **Cancel**

© Copyright IBM Corporation 2015

Figure 7-18. Filtering trace data (1 of 2)

WB393 / ZB3931.0

Notes:

The **monitoring options** section of the Rule Execution Server console summarizes the trace options that are stored in Decision Warehouse.

When you select filtering options, the monitoring.filters property is enabled and the value of this property uses the filters as values.

The possible trace options include:

- **Execution Date:** The execution start date.
- **Execution Duration:** A measure of time that the engine took to execute the ruleset.
- **Total Number of Tasks Executed:** The number of task instances that the engine executed.
- **Total Number of Tasks Not Executed:** The number of ruleflow tasks that the engine did not execute.
- **Total Number of Rules Fired:** The number of rule instances that the engine executed.
- **Total Number of Rules Not Fired:** The number of rules that the engine did not execute.
- **Execution Events:** The full execution tree for the executed ruleset.

The tree is composed of executed task instances and executed rule instances.

- **List of All Tasks:** The ruleflow tasks in the ruleset.
- **List of Tasks Not Executed:** The ruleflow tasks that the engine did not execute.

To get the list of rule instances that were executed, use the Execution Events trace option.

- **List of All Rules:** All the rules in the ruleset.
- **List of Rules Not Fired:** All the rules that the engine did not execute.

To get the list of task instances that were executed, use the Execution Events trace option.

- **Bound Object by Rule:** Used with the Execution Events trace option.

When set, the rules that are executed in the execution tree are bound to the objects to which they were applied.

- **System Properties:** The JRE system properties that Rule Execution Server uses; for example, JAVA_HOME
- **Working Memory with optional filter:** The contents of the working memory.

Use the optional filter to define specific classes that you want to record.

The following filter records the objects that are specified in the classes MyClass and MyOtherClass of the package com.xyz:

```
com.xyz.MyClass,com.xyz.MyOtherClass
```

Select All	Name	Value
<input type="checkbox"/>	monitoring.enabled	true
<input type="checkbox"/>	monitoring.filters	INFO_EXECUTION_DATE=true,INFO_EXECUTION_DURATION=true
<input type="checkbox"/>	monitoring.inout.filters	borrower.some.heavy.property,loan.longfield
<input type="checkbox"/>	ruleset.bom.enabled	true
<input type="checkbox"/>	ruleset.sequential.trace.enabled	true

properties 1 - 5 of 5

Upload properties from file

Choose file: [Browse...](#) [Proceed to update](#) [Preview update](#) Override existing properties

Hide Monitoring Options

monitoring options

✓ Tracing in the Decision Warehouse is currently enabled
Ruleset parameters will be stored as BOM XML (with filter borrower.some.heavy.property,loan.longfield)

The following execution traces will be stored:
Execution Date
Execution Duration

[Edit](#)

© Copyright IBM Corporation 2015

Figure 7-19. Filtering trace data (2 of 2)

WB393 / ZB3931.0

Notes:

Here you see the `monitoring.filters` property that is set with the Execution Date and Execution Duration filters.

By limiting the filters that you select, you minimize the amount of data that is stored in Decision Warehouse. Choose only filters for information that you require. For example, by selecting the Execution Date filter, you can track ruleset executions by their time stamp. The Execution Duration filter can indicate whether there is a performance issue if the duration seems to take too long.

How filters work

- If you do *not* set the `monitoring.filters` property, Decision Warehouse stores all traces
- If you set the `monitoring.filters` property, the Decision Warehouse stores *only* the trace information for the filters that are set to `true`
 - By default, all the filters that are controlled by the `monitoring.filters` property are set to `false` when you define the property
- If you set one filter to `true`, the Decision Warehouse excludes all other execution information from the trace
 - Only the information for which you set a filter to `true` is included in the trace

© Copyright IBM Corporation 2015

Figure 7-20. How filters work

WB393 / ZB3931.0

Notes:

If you do not select all the available options for the decision traces, you filter some of the decision traces. The `monitoring.filters` property is automatically added to the ruleset, with the list of decision traces that you select.

Removing the `monitoring.filters` property from the ruleset has the same effect as selecting all the possible decision traces, that is, setting no filters.



Removing BOM serialization

- When the property `ruleset.bom.enabled` is set to `true`, the list of objects in the ruleset parameters are converted into a memory buffer by using BOM serialization
- For large amounts of input and output data, BOM serialization can result in poor performance
- To optimize performance, either:
 - Turn off BOM serialization by setting `ruleset.bom.enabled` to `false`
 - Keep `ruleset.bom.enabled` set to `true` but use a filter to reduce the effect on performance

© Copyright IBM Corporation 2015

Figure 7-21. Removing BOM serialization

WB393 / ZB3931.0

Notes:

You can specify how to store the ruleset parameter values:

- **Native format:** To turn off the BOM serialization of the objects in the ruleset parameters
Native format is the default value that corresponds to the absence of the `ruleset.bom.enabled` property.
In native format, the ruleset parameters are stored as follows:
 - Java XOM: The ruleset parameters are stored as string representations that are obtained by calling `java.lang.Object.toString`
 - Dynamic XOM: The ruleset parameters are stored as XML
- **BOM format with an optional filter:** To convert the list of objects in the ruleset parameters into a memory buffer by using BOM serialization
BOM format with an optional filter is set when you define the `ruleset.bom.enabled` property, and set its value to `true`.
With this format, the ruleset parameters are stored as BOM XML, with or without a filter.



Removing BOM serialization

ruleset.bom.enabled is set to true when BOM format with optional filter is selected

Select All	Name	Value
<input type="checkbox"/>	monitoring.enabled	true
<input type="checkbox"/>	monitoring.filters	
<input type="checkbox"/>	ruleset.bom.enabled	true
<input type="checkbox"/>	ruleset.sequential.trace.enabled	true

properties 1 - 4 of 4 prev 10 next 10

Upload properties from file

Choose file:

Hide Monitoring Options

monitoring options

Enable tracing in Decision Warehouse

Store the values of the ruleset parameters to:

BOM format with optional filter

Native format

ruleset.bom.enabled is set to false when Native format is selected

Select All	Name	Value
<input type="checkbox"/>	monitoring.enabled	true
<input type="checkbox"/>	monitoring.filters	
<input type="checkbox"/>	ruleset.bom.enabled	false
<input type="checkbox"/>	ruleset.sequential.trace.enabled	true

properties 1 - 4 of 4 prev 10 next 10

Upload properties to BOM file

Choose file:

Hide Monitoring Options

monitoring options

Enable tracing in Decision Warehouse

Store the values of the ruleset parameters to:

BOM format with optional filter

Native format

© Copyright IBM Corporation 2015

Figure 7-22. Removing BOM serialization

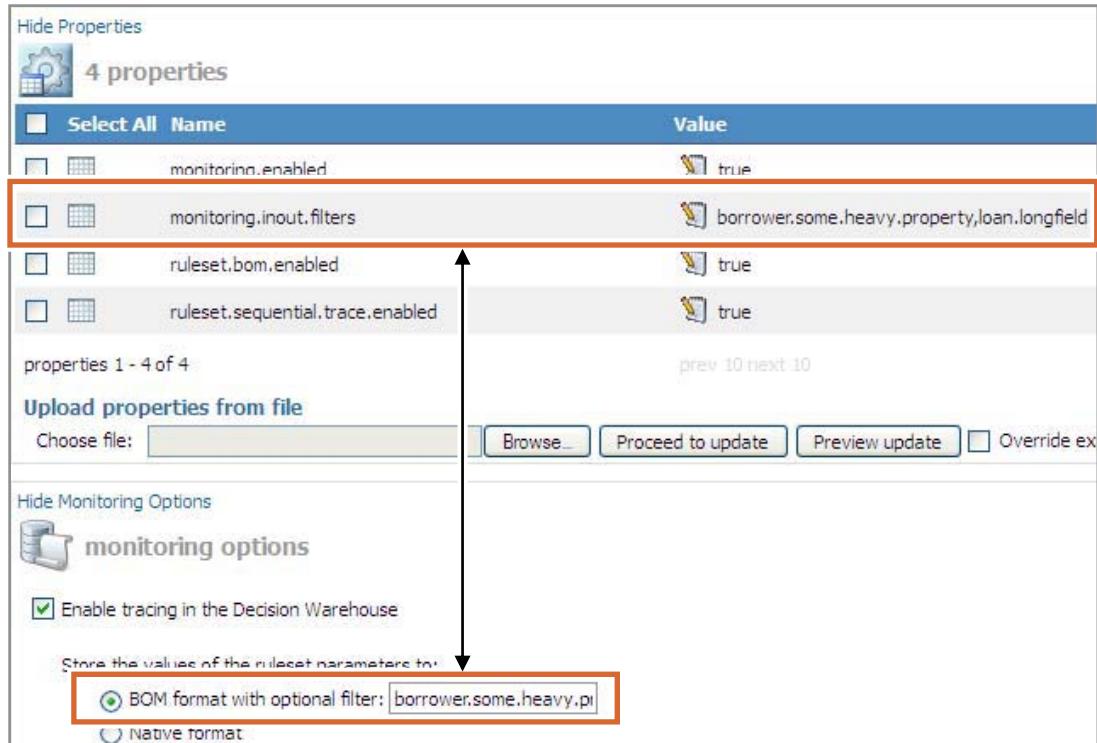
WB393 / ZB3931.0

Notes:

When you set `ruleset.bom.enabled` to `true`, BOM serialization is turned on. When you turn it off, the **Native format** monitoring option is automatically selected.

WebSphere Education 

Reducing performance impact during BOM serialization



The screenshot shows the 'Properties' section of the IBM ODM interface. It displays four properties:

Name	Value
monitoring.enabled	true
monitoring.inout.filters	borrower.some.heavy.property,loan.longfield
ruleset.bom.enabled	true
rulesetSEQUENTIAL.trace.enabled	true

Below the properties, there is an 'Upload properties from file' section and a 'Hide Monitoring Options' section. The 'monitoring options' section contains a checked checkbox for 'Enable tracing in the Decision Warehouse'. At the bottom, there is a dropdown menu for 'Store the values of the ruleset parameters to:' with two options: 'BOM format with optional filter:' (selected) and 'Native format'. An arrow points from the 'monitoring.inout.filters' property in the main table down to the selected 'BOM format with optional filter:' option in the dropdown.

© Copyright IBM Corporation 2015

Figure 7-23. Reducing performance impact during BOM serialization

WB393 / ZB3931.0

Notes:

If you want to capture BOM information, for example, during testing, you can add a filter to the **BOM** format with optional filter to filter the contents of serialized in-out parameters. For example, if you know that your ruleset parameters include classes that you do not want captured in the trace, you specify those classes as a filter. In the example here, consider these fictitious classes:

- borrower.some.heavy.property
- loan.longfield

By adding these classes as filters, you set another ruleset property called **monitoring.inout.filters**:

monitoring.inout.filters=borrower.some.heavy.property,loan.longfield

When the trace information is retrieved, this filter removes these objects from the trace.

7.4. Customizing Decision Warehouse

Customizing Decision Warehouse



© Copyright IBM Corporation 2015

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

10.1

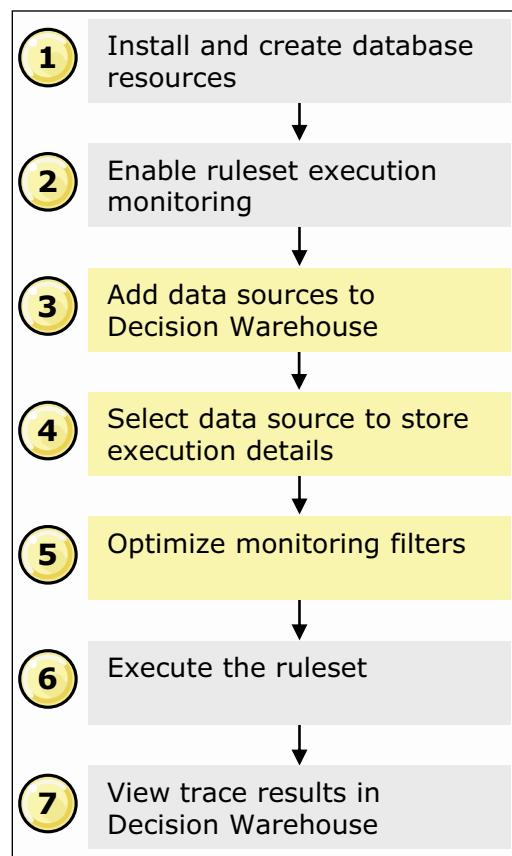
Figure 7-24. Customizing Decision Warehouse

WB393 / ZB3931.0

Notes:

Customizing Decision Warehouse

- Customizing Decision Warehouse includes:
 - Customizing the data sources
 - Optimizing monitoring filters
- Data sources
 - Customize how and where trace data is stored
 - Use multiple data sources
- Use Rule Execution Server console to query any Decision Warehouse data source



© Copyright IBM Corporation 2015

Figure 7-25. Customizing Decision Warehouse

WB393 / ZB3931.0

Notes:

Using the Rule Execution Server console, you can query any Decision Warehouse data source.

A typical example of using multiple data sources includes:

- One data source is configured to store the production data in real time and the client application references this data source to store decision traces.
- Another data source references a historical database on which users can query data through the Rule Execution Server console.



Persistence properties

- Decision Warehouse can use multiple data sources that you can query by using the Rule Execution Server console
 - By default, Decision Warehouse uses the data source that is defined for Rule Execution Server

The screenshot shows the 'Decision Warehouse' tab selected in the top navigation bar. The main content area is titled 'Persistence Properties'. A red box highlights this title. Below it, a sub-section titled 'Choose one of the persistence configurations:' contains two tabs: 'Configuration' (selected) and 'Properties'. Under 'Configuration', there is a table with one row. The row shows 'JNDI_NAME=jdbc/resdatasource' and 'defaultDWConfiguration' (with a checked radio button). The 'Properties' tab is also visible. At the bottom of the configuration section is a 'Select' button.

© Copyright IBM Corporation 2015

Figure 7-26. Persistence properties

WB393 / ZB3931.0

Notes:

The JAR files of the product distribution include a default data source that you can customize. Each application server has a default data source that stores the execution traces of the rulesets that are enabled for Decision Warehouse.

Developers can customize how and where the trace data is stored.

To use a new data source, developers must:

- Configure the calling client application so that it uses this new data source to store information in the Decision Warehouse.
- Configure the Rule Execution Server console so that users can select the new data source from the list of configured sources in the Decision Warehouse tab and query its content.

7.5. Troubleshooting

Troubleshooting



© Copyright IBM Corporation 2015

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

10.1

Figure 7-27. Troubleshooting

WB393 / ZB3931.0

Notes:

Decision Warehouse does not show stored trace data after ruleset execution

Symptom

- No execution trace seems to be captured or the Decision Warehouse queries through the Rule Execution Server console return empty results

Possible resolution

- Ensure that the following properties are set correctly on your ruleset:
 - monitoring.enabled=true
 - rulesetSEQUENTIAL.trace.enabled=true (if the ruleset contains tasks that use the Sequential or Fastpath execution modes)
- If these properties are not set correctly, the trace of a ruleset execution is not stored in Decision Warehouse
- You can set these properties on a ruleset through the Rule Execution Server console

© Copyright IBM Corporation 2015

Figure 7-28. Decision Warehouse does not show stored trace data after ruleset execution

WB393 / ZB3931.0

Notes:

For more information, see:

<http://www.ibm.com/support/docview.wss?uid=swg21424948>



Reducing stored execution traces in Decision Warehouse

Symptom

- Execution traces use too much space
 - If you select all the filters, a large amount of information is stored in the field `FULL_EXECUTION_TRACE`
 - Depending on the size of the ruleset, a full execution trace can use a large amount of space
 - Over time, after running thousands of executions, the execution trace data takes up too much space in the database

Possible resolution

- Apply filters on rulesets to optimize the storage of the execution trace data
 - Deciding on which filter to apply is subject to the requirements of each application
 - Consider setting `INFO_TOTAL_RULES_NOT_FIRED=false` and `INFO_TASKS_NOT_EXECUTED=false` to reduce the data on the rules that were not fired

© Copyright IBM Corporation 2015

Figure 7-29. Reducing stored execution traces in Decision Warehouse

WB393 / ZB3931.0

Notes:

For more suggestions on reducing the amount of trace information that is stored in Decision Warehouse, see:

<http://www.ibm.com/support/docview.wss?uid=swg21586717>

Unit summary

Having completed this unit, you should be able to:

- Enable ruleset monitoring
- Audit the execution of rulesets with Decision Warehouse
- Optimize Decision Warehouse

© Copyright IBM Corporation 2015

Figure 7-30. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** To have ruleset execution trace data, you must first enable monitoring at the ruleset level.
2. **True or False:** You cannot monitor ruleset execution for RuleApps that are deployed from Decision Center.
3. **True or False:** You can optimize Decision Warehouse by enabling BOM serialization.

© Copyright IBM Corporation 2015

Figure 7-31. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. **True.** *If you do not enable ruleset monitoring, no traces are generated.*
2. **False.** *You can enable monitoring on any RuleApp that is deployed to Rule Execution Server. You can set the monitoring properties in Rule Designer or Decision Center or in Rule Execution Server console.*
3. **False.** *BOM serialization can severely affect performance, so you can turn it off and use **Native Format** for the BOM.*

© Copyright IBM Corporation 2015

Figure 7-32. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 9



Auditing ruleset execution through
Decision Warehouse

© Copyright IBM Corporation 2015

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

10.1

Figure 7-33. Exercise 9

WB393 / ZB3931.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Enable monitoring for ruleset execution
- Retrieve decision traces through Decision Warehouse
- Optimize Decision Warehouse
- Delete trace data from Decision Warehouse

© Copyright IBM Corporation 2015

Figure 7-34. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 8. Monitoring execution and performance

What this unit is about

In this unit, you learn how to review logs to troubleshoot and improve performance.

What you should be able to do

After completing this unit, you should be able to:

- Monitor ruleset execution through the Rule Execution Server console
- Manage logging properties for the Execution Unit (XU)
- Describe how to improve Rule Execution Server performance and troubleshoot issues

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Monitor ruleset execution through the Rule Execution Server console
- Manage logging properties for the Execution Unit (XU)
- Describe how to improve Rule Execution Server performance and troubleshoot issues

© Copyright IBM Corporation 2015

Figure 8-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Monitoring ruleset execution
- Monitoring the Execution Unit (XU) on WebSphere Application Server
- Tuning performance
- Troubleshooting Rule Execution Server

© Copyright IBM Corporation 2015

Figure 8-2. Topics

WB393 / ZB3931.0

Notes:

8.1. Monitoring ruleset execution

Monitoring ruleset execution



© Copyright IBM Corporation 2015

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

10.1

Figure 8-3. Monitoring ruleset execution

WB393 / ZB3931.0

Notes:



Monitoring rulesets in Rule Execution Server console

- Enable ruleset monitoring
- Generate ruleset execution statistics
- Test ruleset execution
- View execution-related events in the XU log

© Copyright IBM Corporation 2015

Figure 8-4. Monitoring rulesets in Rule Execution Server console

WB393 / ZB3931.0

Notes:

In addition to Decision Warehouse, the Rule Execution Server console provides tools to monitor ruleset execution.



Monitoring ruleset execution statistics (1 of 2)

- Rule Execution Server provides monitoring information about the execution of a ruleset and the status of logged events during the session
- Ruleset metrics include:

Name	Description
Count	The number of executions, for all execution units
TotalTime	The total time that is spent in the execution, for all execution units, in milliseconds
AverageTime	The average time of an execution, for all execution units, in milliseconds
MaxTime	The time of the longest execution, for all execution units, in milliseconds
MinTime	The time of the shortest execution, for all execution units, in milliseconds
FirstTime	The date of the first execution, for all execution units
LastTime	The date of the last execution, for all execution units

© Copyright IBM Corporation 2015

Figure 8-5. Monitoring ruleset execution statistics (1 of 2)

WB393 / ZB3931.0

Notes:

The ruleset metrics are available for each executed ruleset through an MBean in the JMX infrastructure of the Java EE server.

When Rule Execution Server is deployed in a Java EE cluster, the metrics are computed by consolidating the statistics of the Execution Units (XUs) deployed in each node of the server.

Ruleset statistics include:

- **Count:** The number of times the ruleset was executed during this session.
- **Total time (ms):** The total time to execute the ruleset.
- **Average time (ms):** The average time to execute the ruleset.
- **Max / Min time (ms):** The longest and shortest ruleset execution times.
- **First / Last Execution Date:** The dates and times of the first and last ruleset executions.
- **Last Execution Time (ms):** The time of the last ruleset execution.

The average time (Average time (ms)) is derived from the total execution time (Total time (ms)) and the number of executions completed (Count).



Monitoring ruleset execution statistics (2 of 2)

- To access execution statistics, go to the Ruleset View for a particular ruleset and click **View Statistics**

The screenshot shows the 'Ruleset View' interface. At the top, there is a toolbar with several buttons: 'Test Ruleset', 'View Statistics' (which is highlighted with an orange box and has an arrow pointing down to its details), 'View Execution Units', 'Upload Ruleset Archive', 'Add Managed URI', 'Add Property', and 'Edit'. Below the toolbar, there is a table with two columns: 'Server' and 'Execution Unit Name'. Under 'Server', it lists 'SamplesCell - SamplesNode - SamplesServer'. Under 'Execution Unit Name', it lists 'default'. To the right of this table is a 'Statistics' section with a table showing various metrics for rule execution. This 'Statistics' section is also highlighted with an orange box.

Statistics	
Metric	Ruleset Execution
Count	16
Total Time (ms)	1018
Average Time (ms)	63.625
Min. Time (ms)	6
Max. Time (ms)	462
Last Execution Time (ms)	24
First Execution Date	Apr 17, 2015 1:20:11 PM GMT-07:00
Last Execution Date	Apr 23, 2015 4:08:27 PM GMT-07:00

© Copyright IBM Corporation 2015

Figure 8-6. Monitoring ruleset execution statistics (2 of 2)

WB393 / ZB3931.0

Notes:

To generate statistics on the previous executions of a ruleset:

- In the Navigator pane, click the relevant ruleset.
- In the Ruleset View, click **View Statistics**.

The Ruleset Statistics View opens and provides a ruleset execution statistics table for each Execution Unit (XU) in the configuration, and consolidated statistics on the entire cluster.



Testing ruleset execution

- The Rule Execution Server console provides a web testing interface for you to test ruleset execution
 - Ruleset should be associated with a XOM for this test

Ruleset View

Test Ruleset (highlighted with a red box) | View Statistics | View Execution Units | Upload Ruleset Archive | Add Managed URI | Add Property | Edit

Input Parameters

Direction	Name	Kind	XOM Type	Value
<input type="checkbox"/>	<input checked="" type="checkbox"/> borrower	native	training_loan.Borrower	<pre>import training_loan.*; Borrower borrower = null;</pre>
<input type="checkbox"/>	<input checked="" type="checkbox"/> loan	native	training_loan.Loan	<pre>import training_loan.*; Loan loan = null;</pre>

Execute Task **Task Name** **Execute**

Call the deployed ruleset (highlighted with a yellow box)

Enter ruleset parameter values (highlighted with a yellow box)

© Copyright IBM Corporation 2015

Figure 8-7. Testing ruleset execution

WB393 / ZB3931.0

Notes:

You can use this test tool to test rulesets that are associated with either dynamic or Java XOMs. If your ruleset uses a Java XOM, you can set the `ruleset.managedxom.uris` property to the location of the Java XOM files.

You can use this web testing interface in the Rule Execution Server console only when the console is deployed in a Java EE environment and where an Execution Unit (XU) is reachable through a JNDI lookup. You can verify that your deployment conforms to this configuration by checking that the result of the XU Lookup diagnostic test is green.



Warning

If you add the `ruleset.managedxom.uris` ruleset property to a ruleset, all execution requests use this same property value. In a production environment, this behavior is unlikely to be the intended one.

You construct the input parameters by using the options that are provided in the interface. You test the deployed ruleset by clicking **Execute**.



Troubleshooting

A common cause for the ruleset test to fail is due to a parsing error where the BOM and B2X do not match the XOM. Make sure to check that the `ruleset.managedxom.uris` property is pointing to the correct JAR files if your ruleset uses a Java XOM.



Logged events on execution units (1 of 2)

- In the Rule Execution Server console, you can view the events that are logged on the execution units (XU) that were used to execute a ruleset

The screenshot shows the 'Ruleset View' interface. At the top, there is a toolbar with several buttons: 'Test Ruleset', 'View Statistics', 'View Execution Units' (which is highlighted with a red box), 'Upload Ruleset Archive', 'Add Managed URI', 'Add Property', and 'Edit'. Below the toolbar, the main area displays a summary of execution units. A large red box highlights a section showing '1 Execution Unit(s)' and a 'Server Name' table. The table has one row with a blue background, showing 'SamplesCell - SamplesNode - SamplesServer'. An arrow points from the 'View Execution Units' button in the toolbar to this table. Another arrow points from the 'SamplesServer' link in the table to a detailed view of the execution unit, which is also enclosed in a red box. This detailed view shows 'Execution Unit 1 - 1 of 1'.

- An icon indicates when there are problems with ruleset execution

© Copyright IBM Corporation 2015

Figure 8-8. Logged events on execution units (1 of 2)

WB393 / ZB3931.0

Notes:

To view logged events on Execution Units (XU):

- Click the **Explorer** tab.
- In the Navigator pane, click the relevant ruleset.
- In the Ruleset View page, click **View Execution Units**.

The Execution Units page is displayed. The Execution Unit (XU) table provides the number of warnings and errors that are logged on each server.

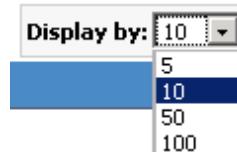
- Click the name of the server in the list of deployed Execution Units.



Logged events on execution units (2 of 2)

- You can modify how the event information is displayed

- To limit the displayed log, use the **Display by** filter



- To sort the log messages by column in ascending order, click the column name: **Level, Creation Date, Message**

1 Last Messages		
Level	Creation Date	Message
[+]	Mar 11, 2013 9:51:54 PM GMT-04:00	The interaction ruleEngine.execute has failed
Error		

- To update the model and the log, click **Refresh** at the top of the page



- To reset the log, click **Reset Execution Unit Messages**

© Copyright IBM Corporation 2015

Figure 8-9. Logged events on execution units (2 of 2)

WB393 / ZB3931.0

Notes:

The events information can be viewed in different ways:

- To limit the number of results that are shown in the log, use the **Display by: 5, 10, 50, 100** filter.
- To sort the log messages by column (**Level, Creation Date, Message**) in ascending order, click the column name. To sort in descending order, click the column name again.

To update the model and the log, click **Refresh** at the top of the page.

To reset the log, click **Reset Execution Unit Messages**.

8.2. Monitoring the Execution Unit (XU) on WebSphere Application Server

Monitoring the Execution Unit (XU) on WebSphere Application Server



© Copyright IBM Corporation 2015

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

10.1

Figure 8-10. Monitoring the Execution Unit (XU) on WebSphere Application Server

WB393 / ZB3931.0

Notes:



Modifying XU configuration properties (1 of 4)

- You can view the Rule Execution Server deployment descriptor (`ra.xml`) through the **Resource Adapter** menu in WebSphere Application Server console
- To see the `ra.xml` configuration properties in the console, open **Resource adapters > View Deployment Descriptor**

The screenshot shows the 'Resources' tree on the left with 'Resource Adapters' expanded. Under 'Resource Adapters', 'Resource adapters' is selected. On the right, there are two panels: 'General Properties' and 'Additional Properties'. In 'General Properties', fields for 'Scope' (cells:xpbaseNode04Cell:nodes:xpbaseNode04) and 'Name' (JRulesXU) are shown. In 'Additional Properties', options include 'J2C connection factories', 'Custom properties', and 'View Deployment Descriptor', with 'View Deployment Descriptor' highlighted by a red box.

Figure 8-11. Modifying XU configuration properties (1 of 4)

WB393 / ZB3931.0

Notes:

Recall during Exercise 2, you configured Rule Execution Server on WebSphere Application Server and deployed the execution unit (XU).



Modifying XU configuration properties (2 of 4)

- To modify `ra.xml` configuration properties, use the **J2C connection factories** menu to access the XU custom properties



General Properties		Additional Properties
* Scope	cells:xpbaseNode04Cell:nodes:xpbaseNode04	Connection pool properties
* Provider	JRulesXU	Advanced connection factory properties
* Name	XU_CF	Custom properties

© Copyright IBM Corporation 2015

Figure 8-12. Modifying XU configuration properties (2 of 4)

WB393 / ZB3931.0

Notes:

Modifying XU configuration properties (3 of 4)

- You can change the values of XU properties in the console and restart the server for new values to take effect

You can monitor the following resources:	
asynchronousRulesetParsing	true
compiledArchivesCacheProperties	
concurrentOpenClose	true
defaultConnectionManagerProperties	pool.maxSize=10, pool.waitTimeout=-1
documentBuilderPoolProperties	pool.maxSize=10, pool.waitTimeout=0
duplicateConnectionEventEnabled	true
idkLoggingEnabled	true
logAutoFlush	false
memoryProfilerIncludedPackageName	com.ibm.rules.,java.,ilog.,org.objectweb.asm.,org.
persistenceProperties	JNDI_NAME=jdbc/resdatasource,XOM_PERSISTENCE
persistenceType	datasource
plugins	{pluginClass=Management,xuName=default,proto {pluginClass=Event,EventDestinationType=Topic,EventDestinationName=}}
profilingEnabled	true
rulesetCacheProperties	ruleset.cache.maintenance.period=300

© Copyright IBM Corporation 2015

Figure 8-13. Modifying XU configuration properties (3 of 4)

WB393 / ZB3931.0

Notes:

You can modify the XU configuration properties directly through the console.

You can also change the `ra.xml` file in the XU RAR, redeploy the XU RAR, and generate a new connection factory. The new connection factory uses the modified `ra.xml` file as a template.

WebSphere Education 

Modifying XU configuration properties (4 of 4)

- The XU manages connections to rule engines and pooling

- You can modify connection pool properties, such as timeouts

- Timeouts can affect performance:
 - Ruleset cache and garbage collection
 - Ruleset parsing

General Properties		Additional Properties
Scope	cells:SamplesCell:nodes:SamplesNode	
* Connection timeout	180	seconds
* Maximum connections	10	connections
* Minimum connections	1	connections
* Reap time	180	seconds
* Unused timeout	1800	seconds
* Aged timeout	0	seconds
Purge policy	EntirePool	

© Copyright IBM Corporation 2015

Figure 8-14. Modifying XU configuration properties (4 of 4)

WB393 / ZB3931.0

Notes:

The XU uses a cache of ruleset instances to avoid parsing rulesets for each execution. By default, a ruleset stays in memory if at least one connection in the JCA connection pool references it. Even if the JCA connection is not active, it references rulesets until the application server deletes them. When a ruleset is no longer referenced by a JCA connection, that ruleset is available for garbage collection. This default behavior ensures that unused rulesets are removed from the memory.

You can modify the JCA connection pool properties on the “Connection pool properties” page in the application server console.

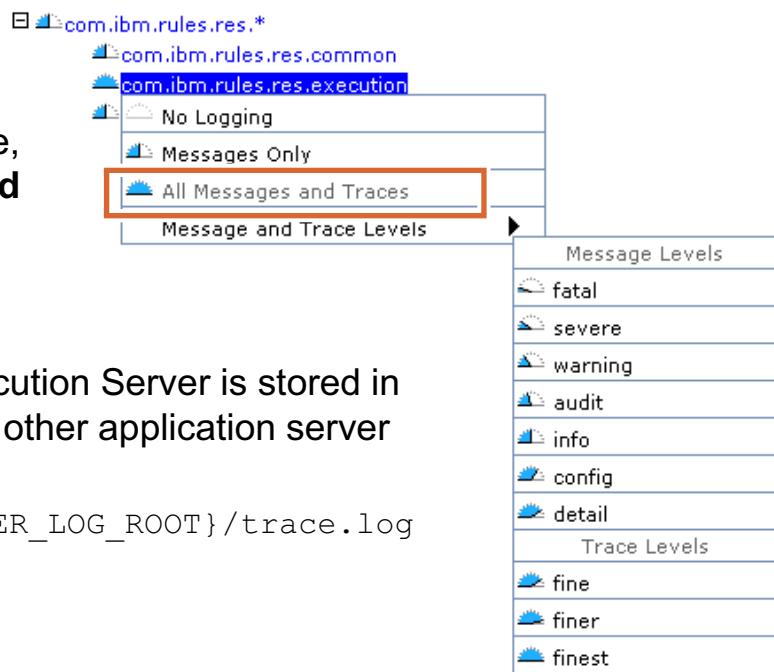
For more information about garbage collection and timeout properties on WebSphere Application Server, see the WebSphere Application Server product documentation:

http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/as_ditamaps/was855_welcome_ndmp.html



Specifying XU trace levels

- You change the XU logging level in the WebSphere Application Server console
- To turn on full trace mode, select: **All Messages and Traces**
- The log for the Rule Execution Server is stored in the same location as the other application server logs
 - Default directory: \${ SERVER_LOG_ROOT }/trace.log



© Copyright IBM Corporation 2015

Figure 8-15. Specifying XU trace levels

WB393 / ZB3931.0

Notes:

On WebSphere Application Server, JDK logging is used by default. You can change the log levels through the console.

On other application servers and for Java SE, you use the `traceEnabled` and `traceLevel` XU properties. To modify those properties, you can change the `ra.xml` file and redeploy the EAR file.

8.3. Tuning performance

Tuning performance



© Copyright IBM Corporation 2015

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

10.1

Figure 8-16. Tuning performance

WB393 / ZB3931.0

Notes:



Common Rule Execution Server performance issues

- Long ruleset execution time
- High memory utilization during ruleset invocation
- High CPU utilization during ruleset invocation
- Slow response time for hosted transparent decision services

© Copyright IBM Corporation 2015

Figure 8-17. Common Rule Execution Server performance issues

WB393 / ZB3931.0

Notes:

Rule Execution Server configuration changes to improve performance

- Rule Execution Server log
 - Verbose logs use resources and reduce performance
 - Set the log level to **Severe** or **Warning** in the production environment
 - If you are not using JDK logging, you can set the `traceLevel` property that is defined in the resource adapter (`ra.xml`) of Rule Execution Server
- Garbage collector
 - On the IBM JVM, the `gencon` garbage collector policy has good results on small and medium applications
 - Tune the garbage collector and memory size
- Ruleset deployment behavior
 - Use asynchronous parsing when you update the ruleset in the application
 - With asynchronous parsing, a separate thread does the parsing, which significantly reduces the impact on ruleset execution threads
 - You can set the `asynchronousRulesetParsing` property to `true` in the `ra.xml` file

© Copyright IBM Corporation 2015

Figure 8-18. Rule Execution Server configuration changes to improve performance

WB393 / ZB3931.0

Notes:

Rule Execution Server performance tuning is very much about testing the application in a load test environment, and determining through experimentation what is a good balance of settings.

For more information about improving performance, see the following links.

- In the product documentation:
 - http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.dserver.rules.res.managing/topics/tpc_res_performance.html
- “Minimizing the impact of ruleset parsing on Rule Execution Server requests” Technote:
 - <http://www.ibm.com/support/docview.wss?uid=swg21427266>

For details on setting the optimal XU pool size, see the white paper *Minimizing the effect of ruleset parsing on the performance of Rule Execution Server*. You can access the white paper at the following web page:

ftp://public.dhe.ibm.com/common/ssi/rep_wh/n/WSW14089USEN/WSW14089USEN.PDF

Improving performance: XU connection pool

- XU connection pooling
 - The optimal XU pool size depends on the number of possible concurrent connections and the number of rulesets to execute
 - Java SE: Set the pool size in the `ra.xml` file
 - Java EE: Configure pool size in the JCA connection factory of the application server
- Memory issues: Reduce pool size

Frequently invoked or updated rulesets	Smaller timeout value: Older ruleset versions flushed out soon
Rarely invoked or updated rulesets	Larger timeout value: Avoid reparsing

- Increase number of JCA connection pools
 - Can configure XU to use multiple JCA connection pools
 - Use a separate connection pool for each ruleset or types of rulesets
 - Good for large rulesets that are not frequently used
 - Diagnose pool status by using XU dump from the Rule Execution Server console

© Copyright IBM Corporation 2015

Figure 8-19. Improving performance: XU connection pool

WB393 / ZB3931.0

Notes:

XU connection pooling

For information about how to calculate the optimum pool size, see the “Rule Execution Server engine pool sizing” technote: <http://www.ibm.com/support/docview.wss?uid=swg21400803>

Adjust Rule Execution Server XU connection pools

If you face memory issues, reduce the connection pool size. Configure the pool size to balance the memory resource limits with the ruleset execution performance requirements. The connection pool size should be large enough so that:

- Connections are available to support concurrent executions on all rulesets without causing eviction of a less-requested ruleset.
- Ruleset parsing is not invoked for less frequently used rulesets.
 - To balance pool size with limited memory resources, consider allocating different connection pools for rulesets based on their invocation and update frequency.
 - You can configure separate connection pools for the following types of rulesets:

- Frequently invoked or updated rulesets. Set a smaller timeout value for this pool, so that older ruleset versions are flushed out sooner from the cache, thus releasing memory resources.
- Rarely invoked or updated rulesets. Set a larger timeout value for this pool to avoid long parsing times that would be incurred when the ruleset is invoked after it is flushed out of the cache.

For each pool, configure the XU timeout to be greater than the maximum expected delay between two execution requests of a ruleset. Doing so reduces the number of times a ruleset gets flushed out because it was not invoked or updated.

Increasing the number of JCA connection pools

This approach helps to ensure that large rulesets are kept in memory even if they are not used infrequently, and that they do not take too many resources during parsing when they are invoked.



Improving performance: Java 2 Security

- Java 2 Security
 - Enabling Java 2 Security decreases performance
 - The application server secure mode can integrate Java 2 Security, but it is not mandatory
 - If Java 2 Security is enabled but the application is not prepared for it, the application is likely to cause Java 2 Security access control exceptions at run time and behave incorrectly
 - For best performance, if possible, turn off Java 2 Security, especially on Java EE

© Copyright IBM Corporation 2015

Figure 8-20. Improving performance: Java 2 Security

WB393 / ZB3931.0

Notes:



Tuning performance: Decision traces

- For best ruleset execution performance and memory usage, turn off the traces
- Execution mode with tracing produces these side effects:
 - Memory usage increases as the execution trace is generated, the list of rules and tasks is cached, and the rule events are generated
 - When a rule or task executes with sequential or Fastpath mode and the ruleset property `rulesetSEQUENTIALTRACE.enabled` is set to `true`, the result is code that produces events and stores mapping information in the ruleset
 - The execution response takes longer when the execution trace is integrated
 - The size of the execution trace depends on the ruleset characteristics, such as number of rules and tasks
- Use filters to reduce the size of the trace and the time to generate it
 - Use the execution trace and Decision Warehouse only when it is mandatory
 - Filtering minimizes the performance costs that are associated with execution tracing
 - The `toString` method can affect the performance if you call an execution trace with the trace filter `setInfoBoundObjectByRule` set to `true`

© Copyright IBM Corporation 2015

Figure 8-21. Tuning performance: Decision traces

WB393 / ZB3931.0

Notes:



Tuning performance: Long ruleset execution times

Symptoms:

- Long response time to first ruleset request
- Rule execution slows down intermittently
- Server takes a long time to start

Actions:

- Configure asynchronous ruleset parsing in the `ra.xml` file
 - Set `asynchronousRulesetProperty` to `true`
- Increase the XU connection pool size and increase the number of JCA connection pools
- Make sure that the JVM debug mode is disabled

© Copyright IBM Corporation 2015

Figure 8-22. Tuning performance: Long ruleset execution times

WB393 / ZB3931.0

Notes:

You can reduce ruleset execution times by changing the topology and optimizing decision warehouse and the Rule Execution Server database.

Suggestions:

- Separate test and production environments
- Cluster Rule Execution Server
 - Choose the right deployment architecture and topology
 - Manage production and test environments separately
 - Configure RES for high availability by deploying a cluster
- Configure XU
 - Adjust the size of the pool of JCA connections
 - Configure the execution unit to monitor ruleset usage data by setting the execution unit property `ruleset UsageMonitorEnabled=true` in the execution unit deployment descriptor (`ra.xml`)

- Adjust the timing of the `ruleset.cache.maintenance.period` property
- Reduce trace in logs
- Optimize decision warehouse
 - Write asynchronously to Decision Warehouse
 - Use separate Decision Warehouse instances for the production and test environments
- Optimize Rule Execution Server database

For more information about reducing rule execution time, see “Improve performance for IBM Operational Decision Manager, Part 1”:

http://www.ibm.com/developerworks/bpm/bpmjournal/1503_siddiqui1/1503_siddiqui1.html

Tuning performance: High memory use

Symptoms:

- High memory usage
- You see OutOfMemoryErrors

Actions:

- Make sure that you are using the right topology for your requirements
- Adjust XU connection pools
 - Configure pool size to balance memory resource limits with ruleset execution performance; reduce pool size if necessary
 - Reduce XU timeout value for frequently invoked or updated rulesets and increase XU timeout value for rarely invoked or updated rulesets
- Disable monitoring of ruleset usage data
 - Set the XU property `rulesetUsageMonitorEnabled` to `false`
 - Note: This action disables monitoring of XU memory usage and disables the effects of the value of the `ruleset.maxIdleTime` ruleset property

© Copyright IBM Corporation 2015

Figure 8-23. Tuning performance: High memory use

WB393 / ZB3931.0

Notes:

To diagnose high memory usage, gather information from:

- Heap dumps
 - Ruleset caching information (execution unit dump) from the Rule Execution Server console

You can analyze the heap consumption to pinpoint the source of the problem, which might be during ruleset parsing or ruleset monitoring. To retrieve information about execution unit memory usage, set the `rulesetUsageMonitorEnabled` execution unit (XU) property to true.

You can reduce memory consumption during ruleset execution by changing the ruleset design, server configuration, and application design. Suggestions include:

- Parse one ruleset at a time
- Load ruleset before use
- Share ruleset among engines
- Reduced ruleset parameter size
- Reduce active ruleset versions

- Choose right topology
- Decrease XU connection pool size
- Use multiple XU connection pools
- Reduce compiled archives in cache
- Configure JVM memory arguments

Tuning performance: High processor use

Symptoms:

- Thread dumps might show many threads waiting for connections
 - High processor use might result from a high number of concurrent requests

Actions:

- Gather and analyze diagnostic information, including `trace.log`
 - Set trace level in application server console: `*=info:com.ibm.rules.*=all`
 - Review thread dumps for more information
- Increase XU connection pool size
- Make sure that the workload is evenly spread between nodes to account for the number of processors
- Consider configuring Rule Execution Server for high availability and scalability by deploying to a cluster
- Use application-server-specific settings to constrain the number of current threads

© Copyright IBM Corporation 2015

Figure 8-24. Tuning performance: High processor use

WB393 / ZB3931.0

Notes:

Gathering diagnostic data

When you review the diagnostic data:

- Look for how many concurrent clients execute simultaneous requests, and how often.
- Look for how many different rulesets are being executed.

Constraining number of threads

For example, if Rule Execution Server is deployed on WebSphere Application Server, look at the following settings.

- Reduce the **Web Container thread pool maximum size** to constrain the number of threads that can actively execute incoming HTTP requests.
- Use the `com.ibm.websphere.ejbcontainer.poolSize` setting to constrain the size of the pool for the specified bean type, which can be stateless, message-driven, or entity beans.
- Set the **Maximum concurrent MDB invocations per endpoint** to the maximum number of concurrent threads that pick up messages. This setting specifies the maximum number of endpoints to which messages are delivered concurrently.



Tuning performance: Hosted transparent decision services

Symptoms:

- Slow response times for web requests
- Might see `SocketTimeOutErrors`

Actions:

- Gather diagnostic information from Rule Execution Server and logs
 - **Rule Execution Server console > Explorer > Service information:** Statistics on the ruleset that was executed through the transparent decision service
 - Logs: `res-decisionservice.log` and web server logging
- Use application-server-specific settings to constrain the number of current threads
- Check application server settings for web container tracing, and remove or reduce tracing if necessary

© Copyright IBM Corporation 2015

Figure 8-25. Tuning performance: Hosted transparent decision services

WB393 / ZB3931.0

Notes:

Gathering diagnostic data

To view the `res-decisionservice.log`, add the following property to the `log4j.lcf` file in **jrules-res-htds.WAS8.ear/jrules-res-htds.war/WEB-INF/classes:**

```
log4j.logger.ilog.rules.res.decisionservice.web.IlrWebServiceServlet=DEBUG,  
DS_LOG_FILE
```

The information that you gather and analyze from Rule Execution Server console and the logs can help determine whether the delayed response time results from long ruleset execution times or other factors.

When you review the diagnostic data:

- Look for how many concurrent clients target HTDS at the same time, and how often.
- Look for how many different rulesets are being executed.
- Look for whether web requests are intermittently slow.

Constraining number of threads

For example, if Rule Execution Server is deployed on WebSphere Application Server, look at the following settings.

- Reduce the **Web Container thread pool maximum size** to constrain the number of threads that can actively execute incoming HTTP requests.

Web container tracing application server settings

For example, if you use WebSphere Application Server, go to **Logging and tracing > <servername> > Diagnostic trace service > Change log detail levels**. If you see the following settings, either remove tracing or reduce the tracing level:

- com.ibm.ws.webcontainer.*=all
- com.ibm.ws.webcontainer.srt=all.



Tuning performance: Planning for growth

- Performance tuning on initial rulesets becomes subpar as rule numbers increase
- For example:
 - Maximum heap might need to be higher
 - Ruleset parsing time might increase
- Try to project the number of rules you expect to reach over time and plan for sufficient capacity
- Plan to reevaluate capacity as rule numbers grow

© Copyright IBM Corporation 2015

Figure 8-26. Tuning performance: Planning for growth

WB393 / ZB3931.0

Notes:

One risk to performance tuning can occur when rule authors are solely in charge of rule maintenance with little involvement from IT. Over time, the number of rules in a ruleset grows. Performance tuning on initial rulesets might be insufficient as the number of rules in the rulesets grows.

To accommodate growth, try to determine expected rule numbers and plan sufficient capacity. You should also reevaluate capacity as the rules increase.

8.4. Troubleshooting Rule Execution Server

Troubleshooting Rule Execution Server



© Copyright IBM Corporation 2015

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

10.1

Figure 8-27. Troubleshooting Rule Execution Server

WB393 / ZB3931.0

Notes:



Troubleshooting the Rule Execution Server installation

- When issues occur during rule execution, check the last lines in the logs for failure
 - <InstallDir>/WAS/profiles/<profile_name>/logs/systemout.log
- In WebSphere Application Server, check the data source definition:
 - JNDI name should not change
 - Make sure that you leave the default JNDI value for your Rule Execution Server installation
 - Make sure that the selected data source supports XA features

© Copyright IBM Corporation 2015

Figure 8-28. Troubleshooting the Rule Execution Server installation

WB393 / ZB3931.0

Notes:



Troubleshooting Rule Execution Server performance

- If a ruleset takes a long time to run:
 - Rule out other factors that might affect performance, such as database lookups and network latency
 - Make sure that you are not running profiling and debugging tools in a production environment; restrict use of profiling and debugging tools to test environments
- Various paths are available for you in case of poor performance:
 - Activate the Execution Unit (XU) log to study the execution trace
 - Configure the XU memory profiler
 - Explore database driver issues
 - Allocate more memory to applications
- Activating the XU log:
 - After setting the trace level, you execute the ruleset to generate traces
 - You can review the traces in the `trace.log` file for your application server

© Copyright IBM Corporation 2015

Figure 8-29. Troubleshooting Rule Execution Server performance

WB393 / ZB3931.0

Notes:

For more information about troubleshooting, see:

- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.troubleshooting/topics/con_support_troubleshoot.html
- http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.distrib.troubleshooting/topics/con_support_troubleshoot_res_was.html



Retrieving the XU dump

- To retrieve the XU dump:
 - In Rule Execution Server console, change the logging level to **Debug**
 - Execute a ruleset to generate traces
- The XU dump for your server can be viewed directly in a browser or downloaded as a compressed file that is called `infos.zip`
 - When you extract `infos.zip`, you get a file that is named `infos.xud`
 - The `infos.xud` file is the XU dump in XML format and it can be opened with a text editor

© Copyright IBM Corporation 2015

Figure 8-30. Retrieving the XU dump

WB393 / ZB3931.0

Notes:



Rule Execution Server database driver issues

Symptom

- You see a JDBC not bound error message when an error occurs during the creation of the data source

Resolution

- Refer to the traces to locate the original cause
- Probable causes:
 - A directory does not exist or cannot be read or written
 - A schema or table is missing
 - Privileges to access the database resource are missing

© Copyright IBM Corporation 2015

Figure 8-31. Rule Execution Server database driver issues

WB393 / ZB3931.0

Notes:



Providing more memory for applications

Symptom

- Core dumps or Java out-of-memory errors

Resolution

- Make more memory available by increasing the WebSphere Application Server memory settings through the WebSphere Application Server console
- In WebSphere Application Server console, open **Servers > Server Types > WebSphere application servers**
 - In the “Application servers” pane, click the name of your server
 - Under **Server Infrastructure**, expand **Java and Process Management** and click **Process definition**
 - Under **Additional Properties**, click **Java Virtual Machine**
 - Set the **Initial heap size** and **Maximum heap size** fields to the appropriate values. For example, you can enter the following values:
 - For a 32-bit configuration: 512 and 1024 MB
 - For a 64-bit configuration: 1024 and 4096 MB
 - Apply and save changes directly to the master configuration

© Copyright IBM Corporation 2015

Figure 8-32. Providing more memory for applications

WB393 / ZB3931.0

Notes:

If you experience core dumps or Java out-of-memory errors, you can make more memory available to WebSphere Application Server by increasing the maximum heap size for the JVM.



Unit summary

Having completed this unit, you should be able to:

- Monitor ruleset execution through the Rule Execution Server console
- Manage logging properties for the Execution Unit (XU)
- Describe how to improve Rule Execution Server performance and troubleshoot issues

© Copyright IBM Corporation 2015

Figure 8-33. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** When using WebSphere Application Server to change the trace level for the Execution Unit log, you modify the deployment descriptor (`ra.xml`).
2. **True or False:** To add or modify ruleset properties, you must work in either Rule Designer or Decision Center and then redeploy the ruleset.
3. **True or False:** You can test ruleset execution in Rule Execution Server console on rulesets that are associated with a managed XOM.

© Copyright IBM Corporation 2015

Figure 8-34. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. **False.** *The ra.xml file contains some trace level properties. However, just modifying the ra.xml file does not change the trace level. You must redeploy the XU RAR file for your changes to take effect. On WebSphere Application Server, you can modify the trace level through the console.*
2. **False.** *You can use the Rule Execution Server console to add or modify ruleset properties on a ruleset that was already deployed.*
3. **True.**

© Copyright IBM Corporation 2015

Figure 8-35. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 10



Monitoring rule execution and performance

© Copyright IBM Corporation 2015

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

10.1

Figure 8-36. Exercise 10

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Manage logging properties for the Execution Unit
- Monitor ruleset execution performance
- Troubleshoot configuration issues for Rule Execution Server
- Improve Rule Execution Server performance

© Copyright IBM Corporation 2015

Figure 8-37. Exercise objectives

WB393 / ZB3931.0

Notes:



Performance tuning overview

- Performance tuning involves developers and administrators
- Suggestions to improve performance
 - Use the engine execution mode (algorithm) that is best suited to your application
 - Choose the topology that is adapted to your application type
 - Use the decision engine (instead of the classic rule engine) to increase the scalability
 - Put performance tests in place to verify the evolution of the application

© Copyright IBM Corporation 2015

Figure 8-38. Performance tuning overview

WB393 / ZB3931.0

Notes:

Unit 9. Managing baselines and multiple releases

What this unit is about

This unit teaches you how to use the decision management features that are available in Decision Center.

What you should be able to do

After completing this unit, you should be able to:

- Manage the decision artifact lifecycle in Decision Center
- Work with version control and history
- Work with baselines in Decision Center
- Manage multiple releases in Decision Center
- Describe the decision governance framework for decision services

How you will check your progress

- Checkpoint
- Exercise

Unit objectives

After completing this unit, you should be able to:

- Manage the decision artifact lifecycle in Decision Center
- Work with version control and history
- Work with baselines in Decision Center
- Manage multiple releases in Decision Center
- Describe the decision governance framework for decision services

© Copyright IBM Corporation 2015

Figure 9-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Rule management features
- Smart folders
- Version control and history
- Baselines
- Multiple release management
- Managing decision services in the Enterprise console

© Copyright IBM Corporation 2015

Figure 9-2. Topics

WB393 / ZB3931.0

Notes:

Recall: ODM administrator

- Responsibilities:
 - Deploying and configuring the server and database for Decision Center and Rule Execution Server
 - Managing user access to Decision Center and Rule Execution Server
 - Configuring trace data sources for testing purposes
 - Deploying applications
 - Redeploying rulesets and event assets as changes are made
 - Generating detailed execution reports
 - Tracking and monitoring rule execution
 - Restoring a particular application state
- Tools: Servers for Decision Center or runtime environments



Administrator

© Copyright IBM Corporation 2015

Figure 9-3. Recall: ODM administrator

WB393 / ZB3931.0

Notes:

Restoring an application state involves understanding version management, baselines, and multiple release management. This unit describes these features as they are used in Decision Center.

9.1. Rule management features

Rule management features



© Copyright IBM Corporation 2015

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

10.1

Figure 9-4. Rule management features

WB393 / ZB3931.0

Notes:



Rule management features

- Management features include:
 - Smart folders
 - Version control
 - Locking
 - Project-wide rule checks and reports
 - Consulting baselines
- Features for a configuration manager or administrator
 - Baseline management
 - Multiple release management
 - RuleApp management and deployment
 - Permissions and other security features
- Many tasks that use the **Project** and **Configure** tabs are restricted to technical users, including administrators and developers

© Copyright IBM Corporation 2015

Figure 9-5. Rule management features

WB393 / ZB3931.0

Notes:

9.2. Smart folders

Smart folders



© Copyright IBM Corporation 2015

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

10.1

Figure 9-6. Smart folders

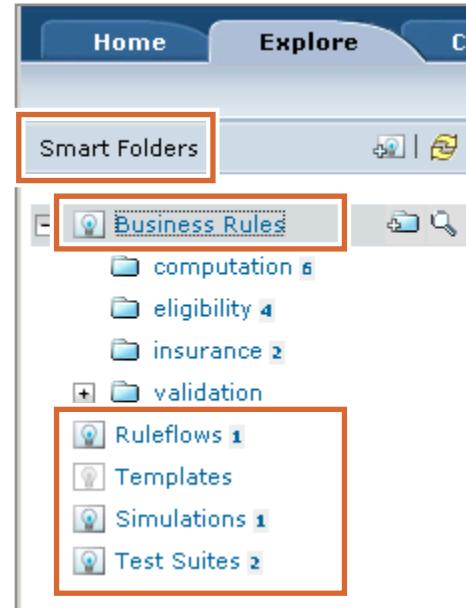
WB393 / ZB3931.0

Notes:



Smart folders for business rule projects

- When business users sign in, they see that the project is organized in *smart folders*
- Default smart folder for a business rule project is **Business Rules**
 - Provides a view of all the rules and rule groups that are contained in the project
- Projects also contain these default smart folders:
 - Ruleflows**
 - Templates**
 - Simulations**
 - Test Suites**
- Smart folders can be customized



© Copyright IBM Corporation 2015

Figure 9-7. Smart folders for business rule projects

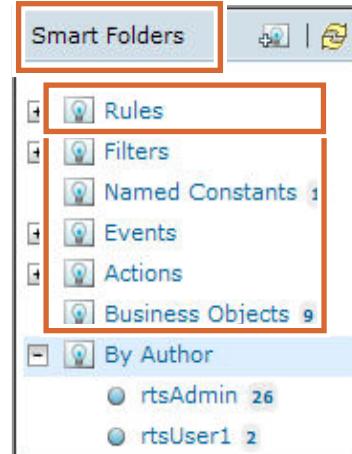
WB393 / ZB3931.0

Notes:

WebSphere Education 

Smart folders for event projects

- The default smart folder for an event rule project is the **Rules** smart folder
 - Lists all the rules and rule groups that are contained in the project
- Projects also contain these default smart folders:
 - Filters**
 - Named Constants**
 - Events**
 - Actions**
 - Business Objects**
- Add new smart folders or customize existing ones



© Copyright IBM Corporation 2015

Figure 9-8. Smart folders for event projects

WB393 / ZB3931.0

Notes:

9.3. Version control and history

Version control and history



© Copyright IBM Corporation 2015

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

10.1

Figure 9-9. Version control and history

WB393 / ZB3931.0

Notes:



Version control in Decision Center

- Versions of specific rules and other project elements are automatically created with each modification
- Baselines create a version for an entire project
 - Keeps a snapshot of a rule project at a specific time
- Baselines and versions allow “rollback” to a previous version

© Copyright IBM Corporation 2015

Figure 9-10. Version control in Decision Center

WB393 / ZB3931.0

Notes:

Decision Center creates archived versions of elements that are contained in your rule projects each time they are modified.

You can use baselines to create a version for the entire project, and you can use versions to roll back when required.

WebSphere Education

Accessing versions in the History menu

- Decision Center tracks every change to each project element
- Every time that you click **Finish** in the editor, a “point” version is created
- You access these versions through the **History** menu

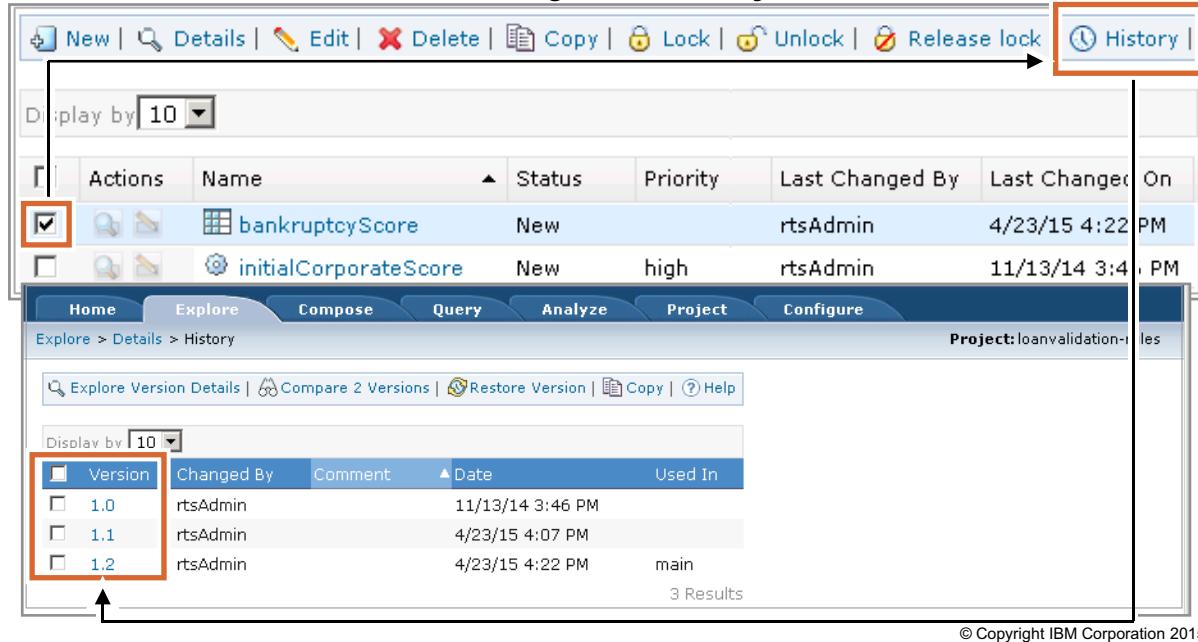


Figure 9-11. Accessing versions in the History menu

WB393 / ZB3931.0

Notes:

Every time that you change an element in the project, a version for that element is created. From the **History** menu, you can consult the history of changes made.



History of changes

- See modifications that were made over time
- Compare two versions
- Restore a former version
 - Creates another version, which is based on the former version
- Past versions cannot be modified
 - You can change only the current version

The screenshot shows a software interface with a top navigation bar containing tabs: Home, Explore, Compose, Query, Analyze, Project, and Configure. The 'Explore' tab is selected. Below the navigation bar, the path 'Explore > Details > History' is displayed. A search bar contains the text 'Explore Version Details | Compare 2 Versions'. To the right of the search bar is a button labeled 'Restore Version' with a small circular icon, which is highlighted with a red box. Below these elements is a table titled 'Display by 10'. The table has columns: Version, Changed By, Comment, Date, and Used In. It lists five versions: 1.0, 1.1, 1.2, 1.3, and 1.4. Version 1.3 is selected, indicated by a blue highlight. The table also includes a checkbox column and a dropdown menu for sorting.

Version	Changed By	Comment	Date	Used In
1.0	rtsAdmin		11/13/14 3:46 PM	
1.1	rtsAdmin		4/23/15 4:07 PM	
1.2	rtsAdmin		4/23/15 4:22 PM	
<input checked="" type="checkbox"/> 1.3	rtsAdmin		4/23/15 4:49 PM	
1.4	rtsAdmin		4/23/15 4:49 PM	main

© Copyright IBM Corporation 2015

Figure 9-12. History of changes

WB393 / ZB3931.0

Notes:

You can use version numbers to identify changes that are made over time and compare versions. You can also restore a past version, which creates a version that is based on the previous version. With Decision Center, you can consult past versions, but you cannot modify them. You can edit only current versions.

9.4. Baselines

Baselines



© Copyright IBM Corporation 2015

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

10.1

Figure 9-13. Baselines

WB393 / ZB3931.0

Notes:

Project baselines

- Baselines capture the state of a project at a specific moment in time
 - Encompass each element in the project and its version
- Baselines can include project dependencies:
 - References to another project
 - References to a baseline of the referred project
- Baselines cannot be edited, only viewed
- The administrator can “unfreeze” a baseline if it is necessary to patch it
 - Version of elements can be changed
 - Elements can be removed or added
- Projects can contain many baselines
 - Regular points for rollback
 - Example: Monthly baselines during major development
 - Useful as part of rule governance processes

© Copyright IBM Corporation 2015

Figure 9-14. Project baselines

WB393 / ZB3931.0

Notes:

With baselines, you can capture the state of a project at a specific moment in time. Creating the baseline is like freezing each element of the project with its version, along with its dependencies and references to other projects.

You can restore a previous baseline so that it becomes the current state of the project. When a baseline is restored, Decision Center automatically creates a version of all the project elements that are found in that baseline.

When you consult a baseline, you cannot edit the elements.

If it is necessary to modify a baseline by adding or removing an element, or changing an element to a different version, the administrator can unfreeze the baseline. But generally you want your baselines to be a stable picture of the project so they can be used for governance purposes.

The configuration manager or administrator creates baselines, so you should coordinate with them when you are ready for a new baseline, such as when your rules are ready for deployment.

Deployment baselines

- You can choose to create a deployment baseline during a RuleApp deployment
 - Use one baseline per ruleset in the RuleApp
 - Developers can automate the creation of baselines when projects are deployed
 - Useful for auditability
- Deployment baselines automatically include all dependent projects
 - Can span projects, taking rules and other artifacts from multiple projects
- Deployment baselines can be redeployed

© Copyright IBM Corporation 2015

Figure 9-15. Deployment baselines

WB393 / ZB3931.0

Notes:

A project can contain many baselines. For example, you can create monthly baselines of a rule project that you can use to consult the state of the project for any specific month.

For auditability, it is good practice to create deployment baselines when deploying a new ruleset.

Baselines versus deployment baselines

- Baselines
 - Created manually at any time by a user
 - Can be restored
 - Can include parent projects if user chooses
- Deployment baselines
 - Cannot be created manually: You create deployment baselines during a RuleApp deployment, by using one baseline per ruleset in the RuleApp
 - Cannot be restored
 - Can be redeployed
 - Automatically includes all dependent projects
- Baselines and deployment baselines
 - Can be consulted as “read only” with no freeze or unfreeze functions
 - Can be cloned to a branch

© Copyright IBM Corporation 2015

Figure 9-16. Baselines versus deployment baselines

WB393 / ZB3931.0

Notes:



Restoring baselines

- Restoring a baseline means that the content of the baseline overwrites the content of the current project state
- You can use this feature to work on or edit previous states of the project:
 1. Create a development baseline that captures the current project state
 2. Restore your development baseline and do the updates, unit tests, and deployments
 3. When you are finished, create a baseline that captures the updated project state
 4. Finally, restore the baseline that is associated with the current project state, created in the first step
- Use this approach for significant bug fixes
 - No unfreezing is involved
 - Uses the current state of the project, which means all the rules are available to facilitate the update process

© Copyright IBM Corporation 2015

Figure 9-17. Restoring baselines

WB393 / ZB3931.0

Notes:

You can restore a baseline so that it becomes the current state of the project or branch.

When you restore a baseline, project elements that were created *after* the baseline was created are overwritten. Before you restore a baseline, you might want to create a temporary baseline, which can help keep a trace of any new project elements that were not captured in the initial baseline.

9.5. Multiple release management

Multiple release management



© Copyright IBM Corporation 2015

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

10.1

Figure 9-18. Multiple release management

WB393 / ZB3931.0

Notes:



Branches

- Branches are integral to working with the Decision Center
 - Before starting any work, you select the target rule project, plus the branch that you want to use in that rule project

Welcome to the Decision Center Home Page

Project in use:	<input type="button" value="loanvalidation-rules"/>
Branch in use:	<input type="button" value="main"/>
Current action:	<input type="button" value="Work upon branch"/>

- By default, a rule project exists with a single branch, the “main” branch

© Copyright IBM Corporation 2015

Figure 9-19. Branches

WB393 / ZB3931.0

Notes:

By using branches, you can manage and update multiple releases in parallel.

Initially, a project contains only a main branch in which you work. When you create a subbranch, it contains an exact replica of every project element that is contained in the parent branch. You can then work on the subbranch without affecting the content of its parent, and eventually merge different branches together.



Managing multiple versions of a rule project

- Changes to a deployed rule application do not interfere with work on an upcoming release

A screenshot of the IBM Decision Center Home Page. The top navigation bar includes "Home", "Explore", "Compose", "Query", and "Analyze". The "Project" dropdown is set to "CustomerAcquisitionSituations". A dropdown menu for "Branch in use" is open, showing "Release 2.1" which is highlighted. Below this, a tree view shows branches: "main", "main", "Release 1.0", "Release 2.0", and "Release 2.1", with "Release 2.1" also being highlighted. A help link "Help is available online" is visible at the bottom left of the dropdown menu.

© Copyright IBM Corporation 2015

Figure 9-20. Managing multiple versions of a rule project

WB393 / ZB3931.0

Notes:

You can implement changes in distinct versions and control how to merge them across different releases. For example, you can deploy a summer pricing rule project while working on the fall or winter one.



Copying rules between branches

Copy item(s)

Select the folder to which you want to copy the selected item
/Pricing/Coverage Pricing/Surc:

OK Cancel

The dialog displays a hierarchical file structure for copying items. The root folder is 'AutoInsuranceQuotingBOM'. Below it are several subfolders: 'bomdomainpopulate', 'custombrl-rules', 'loanvalidation-rules', 'miniloan-rules', 'patternmatching', and 'Pricing'. The 'Pricing' folder is expanded, showing its contents: 'main', 'Pricing', 'queries', 'templates', '2010 release', '2011 release', '2011 Q2 release', '2011 Q4 release', and '2012 release'. Further sub-folders like 'Pricing', 'queries', and 'templates' are visible under the '2011 release' and '2012 release' branches. At the bottom of the list are two more items: 'squery-loanvalidation-rules' and 'valueeditor-rules'.

- You can copy rules from one branch to another

© Copyright IBM Corporation 2015

Figure 9-21. Copying rules between branches

WB393 / ZB3931.0

Notes:



Multiple release management visual support

- You manage branches from the **Project** tab
- Detailed side-by-side comparison highlights differences between releases
- Multi-directional merge operation allows you to decide how to merge changes from one release to another

The screenshot shows the IBM Decision Center interface. The top navigation bar includes Home, Explore, Compose, Query, Analyze, Project, and a dropdown menu. The Project tab is selected, and the sub-project 'CustomerAcquisition' is visible. Below the navigation, a 'Project' section contains a 'Manage Project' heading. Underneath are several options: 'Manage Subbranches and Baselines' (with a sub-note about creating, deleting, and renaming), 'Merge Branches' (which is highlighted with a red box), 'Edit Project Dependencies' (with a note about viewing and editing dependencies), and 'Edit Project Options' (with a note about viewing and editing project options). At the bottom right of the screenshot, there is a copyright notice: '© Copyright IBM Corporation 2015'.

Figure 9-22. Multiple release management visual support

WB393 / ZB3931.0

Notes:

After a release is completed, you can merge specific branches after doing a side-by-side comparison.



Merging branches

- When you choose the two branches to merge, Decision Center opens a table that lists all their differences and proposes possible actions
- You decide what to do with each project element that is different between the two branches
 - Replace the version in the target branch
 - Update the source branch with the version found in the target branch
 - Take no action at all

Differences to Merge

Specify the action to perform for each project element

Name	Folder	main	training	Action
minimum income	eligibility	Unmodified	Modified	Replace in 'main' <input checked="" type="radio"/> Replace in 'main' <input type="radio"/> No action to perform <input type="radio"/> Replace in 'main'

© Copyright IBM Corporation 2015

Figure 9-23. Merging branches

WB393 / ZB3931.0

Notes:

During the merge, the side-by-side comparison helps you to understand rule by rule differences from one release to another.

For each rule, you can control the direction of the merge operation. At the end of the operation, a detailed synchronization status is provided to ensure that the merge was correctly completed and that the two releases are correctly synchronized.

To help Decision Center propose the correct action to take, you can specify your preferred direction for merging from the following options:

Bidirectionally

This option is the default. The proposed action is based on the assumption that you want to reflect changes that are made in either branch as follows:

- If a new rule exists in one branch, you want to add it to the other branch.
- If a rule is deleted in one branch, you want to delete it from the other branch.
- If a rule is modified in one branch but not in the other, you want to update the unmodified one.

- If a rule is modified in both branches, no action is proposed, and you must specify what action to take.

Only to target branch

When you want to push all changes you made in the working branch to the target branch, but not keep the changes made in the target branch.

Only to source branch

When you want to accept all changes from the target branch, but not keep the changes in your working branch.

Using Diff and Merge

- When you cannot resolve differences merely by copying from one branch to another, use **Diff and Merge**
 - Available from the **Merge Branches** page as an icon beside differences that are listed in the **Differences to Merge** section

Differences to Merge

Specify the action to perform for each project element

Name	Folder	training	main
minimum income	eligibility	Modified	Unmodified
repayment and score	eligibility	Modified	Unmodified

Filter:

► Replace in 'main'

- Detailed view of differences for fine-grained difference management

Changes	Content
Defined	<pre>if the yearly repayment of 'the loan' is more than the yearly income of 'the borrower' * 0.42 " add "Too big Debt-To-Income ratio" to the messages of 'the loan'; reject 'the loan';</pre>
	<pre>main (Unmodified) if the yearly repayment of 'the loan' is more than the yearly income of 'the borrower' * 0.3 " add "Too big Debt-To-Income ratio" to the messages of 'the loan'; reject 'the loan';</pre>
Status	New

© Copyright IBM Corporation 2015

Figure 9-24. Using Diff and Merge

WB393 / ZB3931.0

Notes:

With the Diff and Merge function, you can choose to retain a mixture of contents or properties from each branch. A detailed view of the differences helps you decide how to resolve the differences.

Branches and RuleApp deployment

- Each ruleset is extracted from a particular branch or baseline
- You must create one RuleApp or ruleset for each branch that is deployed

Name*	PricingRules
Display Name	
Description	
Project*	Pricing
Version*	2011 Q4 release ▾
Extractor	<input type="checkbox"/> main <input type="checkbox"/> 2010 release <input checked="" type="checkbox"/> 2010 May baseline <input checked="" type="checkbox"/> 2010_09_15 deployment <input type="checkbox"/> 2011 release <input type="checkbox"/> 2011 Q2 release <input type="checkbox"/> 2011 Q4 release
Major	
Minor	
Enabled	
Debug	
Ruleset Prop	

Available RuleApps								
New Details Edit Delete Deploy Redeploy Refresh Help								
Display by 10 ▾								
#	Name	Display Name	Major	Minor	Created On	Created By	Last Changed On	Last Changed By
<input type="checkbox"/>	PricingApp_2011_Q4		1	0	10/25/11 1:34 PM	rtsAdmin	10/25/11 1:34 PM	rtsAdmin
<input checked="" type="checkbox"/>	PricingApp_2012_Q1		1	0	10/25/11 6:14 PM	rtsAdmin	10/25/11 6:14 PM	rtsAdmin

2 Res

© Copyright IBM Corporation 2015

Figure 9-25. Branches and RuleApp deployment

WB393 / ZB3931.0

Notes:



Branches and permissions

- Security is available at the branch level

Branch Security

[Save](#) | [Cancel](#) | [Help](#)

You are currently editing the security settings for project: Pricing, branch: 2011 Q4 release

Do not enforce security for this branch
 Enforce and configure security for this branch
 Inherit security configuration from parent branch: security is not enforced in parent branch

Select the groups that can access the branch:

Available groups:	Selected groups:
Eligibility Validator rtsConfigManage	rtsUser

- Permissions can be defined on branches and baselines in the same way as for other artifacts
 - Can be inherited from parent branch
 - Set according to group

Permission	Value	Type	Property
Create	Yes	Baseline or Branch	-
Apply		Cancel	

© Copyright IBM Corporation 2015

Figure 9-26. Branches and permissions

WB393 / ZB3931.0

Notes:

You can define branch security options in the Decision Center Enterprise console **Configure** tab by going to **Security > Edit Branch Security**.

You can edit permissions in the Decision Center Enterprise console **Configure** tab by going to **Security > Edit Permissions**. Select the group that you want to define permissions for, and then click **New** to define a new set of permissions or **Edit** to define an existing set of permissions. Select the **Permission** (Create, Update, Delete), the **Value** (Yes or No,) and make sure that **Baseline or Branch** is selected in the **Type** menu.

9.6. Managing decision services in the Enterprise console

Managing decision services in the Enterprise console



© Copyright IBM Corporation 2015

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

10.1

Figure 9-27. Managing decision services in the Enterprise console

WB393 / ZB3931.0

Notes:



Decision services and the decision governance framework

- The decision governance framework is a built-in, prescriptive way for business users to manage and govern rules through decision services
 - Business users do most of the work within the governance framework in the Decision Center Business console
- Decision service:
 - Contains one or more rule projects
 - Holds a top-level project and all of its dependent projects
- Governance framework is based on:
 - The states of decision service releases and activities
 - User roles and permissions

© Copyright IBM Corporation 2015

Figure 9-28. Decision services and the decision governance framework

WB393 / ZB3931.0

Notes:

The Decision Center Business console is not covered in this course.



Decision service releases, activities, and snapshots

- Releases
 - Capture and trace changes to rule projects during a specified time
 - Business goals drive changes to rule projects
 - Releases have beginning and end dates
 - To be deployed, a release must be completed and approved
- Activities
 - Change activities: Create, delete, or modify rules
 - Validation activities: Manage test plans for validating changes to rules, and run tests and simulations
 - Activities must be completed and approved before a release can be approved
- Snapshots
 - Capture the state of a release at a specific point in time
 - Can use snapshots to restore release to a different state

© Copyright IBM Corporation 2015

Figure 9-29. Decision service releases, activities, and snapshots

WB393 / ZB3931.0

Notes:

Releases

A release tracks and manages the changes that a group of participants makes and the validation of these changes.

Because rules change over time, Decision Center uses releases to capture and trace all changes that relate to a purpose and period in time.

The purpose is a set of business-driven goals, and the time is a beginning date and an end date. You cannot edit a release directly. Rather, you create and work in change activities of the release, as many as needed to accomplish the objectives.

Each release stems from an existing release. Work occurs on the release while it is open, and ends when you complete, approve, and deploy the release.

Change activities

In the governance framework, change activities manage the work of participants that collaborate on a common goal, in the larger context of a release.

In a change activity, a group of participants creates, modifies, or deletes rules, and they also get these changes approved.

Each change activity is a branch that stems from the release, and contains the version of the rules that are found in the release at the moment the change activity is created. When a change activity is complete, Decision Center merges it back into the release. You can create many change activities for a release.

Validation activities

Validation activities are created and managed in the Business console, and are used to track and manage a test plan for the release and the results.

When all validation activities are completed, the release can be approved and completed, at which point deployment can occur.

Snapshots

Snapshots capture the state of a release or activity at a specific moment in time.

You can create snapshots of releases or activities, look at existing ones for consultation, and compare different snapshots. You can also restore a snapshot or release if you have the appropriate permissions. Each release or activity can contain many snapshots. When you consult a snapshot, you cannot edit any of the project elements; you must return to the current state to do so. Also, Decision Center automatically creates snapshots in some situations.



Managing decision service changes in the Enterprise console

- Decision service changes can be viewed and managed in the Enterprise console
 - Similar to regular rule projects
 - Most work on decision services is done in the Business console
- Project tab
 - View and manage decision service releases as a subbranch of the main decision service branch
 - View and manage decision service change and validation activities as subbranches of the release
 - View and manage release decision service snapshots as baselines
- Administrators can also create and rename release branches in the Enterprise console for business users

© Copyright IBM Corporation 2015

Figure 9-30. Managing decision service changes in the Enterprise console

WB393 / ZB3931.0

Notes:

Decision service releases, activities, and snapshots can be viewed in the Enterprise console as branches, subbranches, and baselines. You can also create and rename release branches in the Enterprise console.

Managing and merging branches

In the Business console, releases and activities are implemented as branches, and baselines are visible as snapshots. Some of these branches and baselines can be renamed or deleted only in the Enterprise console.

Setting permissions

If decision services have project security that is enabled, you must give the appropriate permissions on release and change activity branches. The usual create, update, and delete permissions can be set to include releases and validation activities and their properties.

Unit summary

Having completed this unit, you should be able to:

- Manage the decision artifact lifecycle in Decision Center
- Work with version control and history
- Work with baselines in Decision Center
- Manage multiple releases in Decision Center
- Describe the decision governance framework for decision services

© Copyright IBM Corporation 2015

Figure 9-31. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** With baselines, you can capture the state of a project at a particular moment in time and restore a project to a particular state.

2. **True or False:** Multiple release management tools make it possible to work on separate versions of the same project simultaneously.

© Copyright IBM Corporation 2015

Figure 9-32. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

1.

2.



Checkpoint answers

1. True.

2. True.

© Copyright IBM Corporation 2015

Figure 9-33. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 11



Managing baselines and multiple releases

© Copyright IBM Corporation 2015

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

10.1

Figure 9-34. Exercise 11

WB393 / ZB3931.0

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Work with versions and history of rule artifacts
- Create baselines in the Decision Center Enterprise console, and use the appropriate baseline to restore previous versions of rule artifacts
- Create project branches and merge branches by using Diff and Merge tools

© Copyright IBM Corporation 2015

Figure 9-35. Exercise objectives

WB393 / ZB3931.0

Notes:

Unit 10. Introducing Decision Server Insights for administrators

What this unit is about

This unit describes the Decision Server Insights architecture, programming model, and installation.

What you should be able to do

After completing this unit, you should be able to:

- Describe Decision Server Insights architecture
- Describe Decision Server Insights installation

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Describe Decision Server Insights architecture
- Describe Decision Server Insights installation

© Copyright IBM Corporation 2015

Figure 10-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Programming model
- Decision Server Insights architecture
- Installing Decision Server Insights

© Copyright IBM Corporation 2015

Figure 10-2. Topics

WB393 / ZB3931.0

Notes:

10.1.Programming model

Programming model



© Copyright IBM Corporation 2015

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

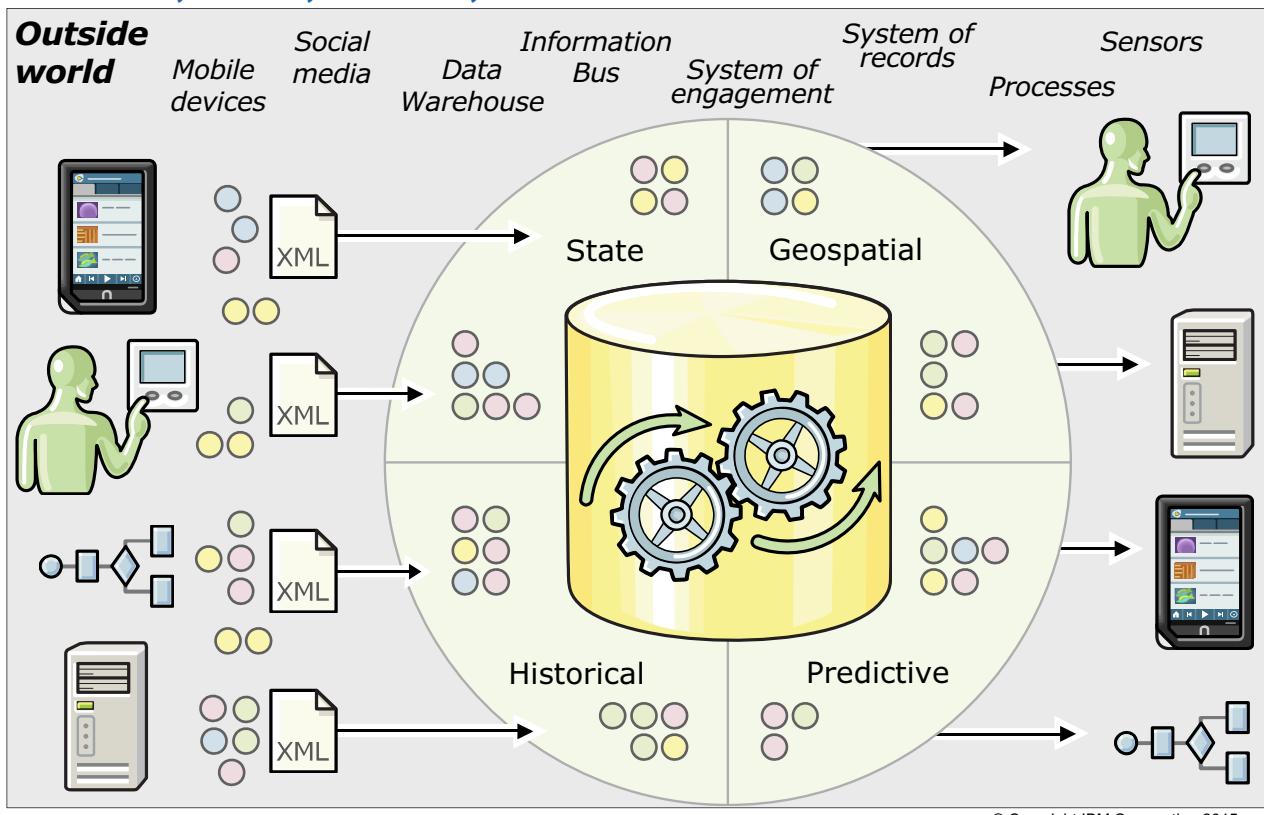
10.1

Figure 10-3. Programming model

WB393 / ZB3931.0

Notes:

Sense, Build, Decide, Act



© Copyright IBM Corporation 2015

Figure 10-4. Sense, Build, Decide, Act

WB393 / ZB3931.0

Notes:

To **sense** what is happening, you must be able to recognize and capture events from the outside world. Decision Server Insights listen to a wide range of event providers, such as social media, physical sensors, mobile devices, and other channels or systems as depicted here.

Events are received as XML messages by Decision Server Insights.

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

Next, **building** the context involves putting data and events into context to make sense of the data and evaluate correlations. The context is built from basic state information, geospatial information such as locations, predictive information by using predictive models, and from historical information that you recorded.

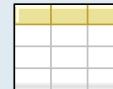
Based on the context, analytics, and by using time and spatial modeling, time-based and location-based reasoning, Decision Server Insights detects and **decides** how to respond to the most intricate patterns and trends.

To **act**, Decision Server Insights emits a response event. That event is published as an XML message on the outbound queue or posted to a URL for some external system to pick up. Actions

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

Core building blocks

Entity



Some business-relevant thing or piece of information

- In Decision Server Insights, entities have an identifier and are composed of a set of attributes and relationships

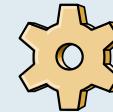
Event



Specific action or measurement that occurred at a specific time and place

- Decision Server Insights encodes events as objects that have a time stamp

Agent



Business logic that is applied to an incoming event to detect situations

- Rule agent
- Java agent
- Predictive scoring agent

© Copyright IBM Corporation 2015

Figure 10-5. Core building blocks

WB393 / ZB3931.0

Notes:

There are three core building blocks in Decision Server Insights: events, entities, and agents.

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

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

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

The third building block is the agent. There are three types of agents:

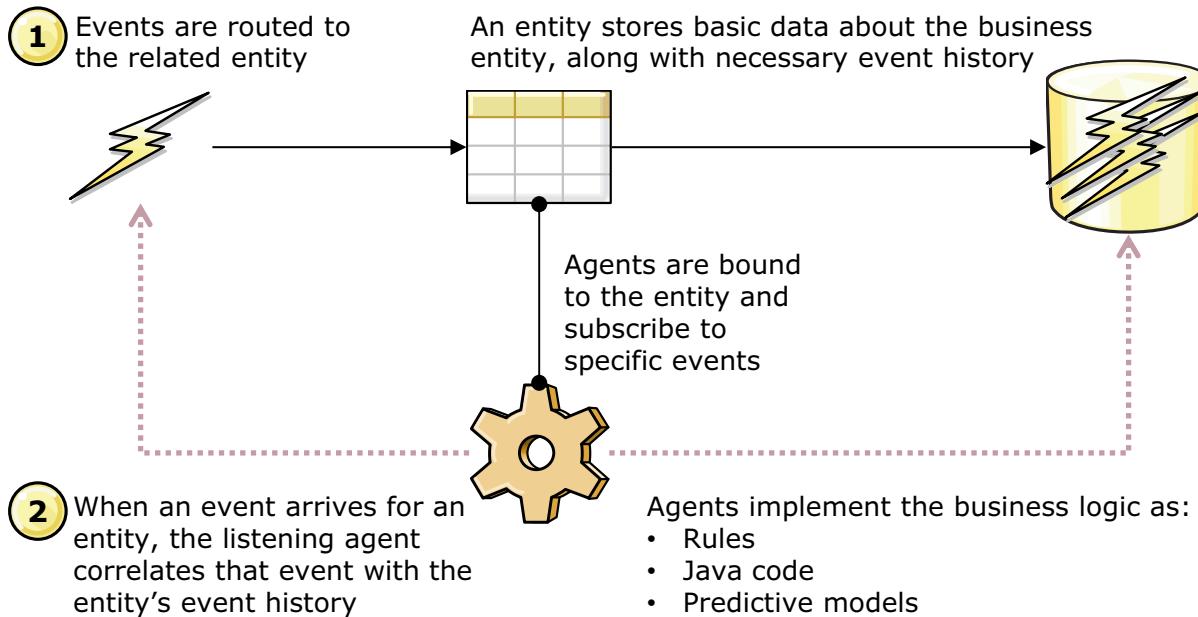
- Rule agents
- Java agents
- Predictive agents

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

Java agents are written in Java code.

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

Programming model



© Copyright IBM Corporation 2015

Figure 10-6. Programming model

WB393 / ZB3931.0

Notes:

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

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

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

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

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

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

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

Aggregates in the programming model

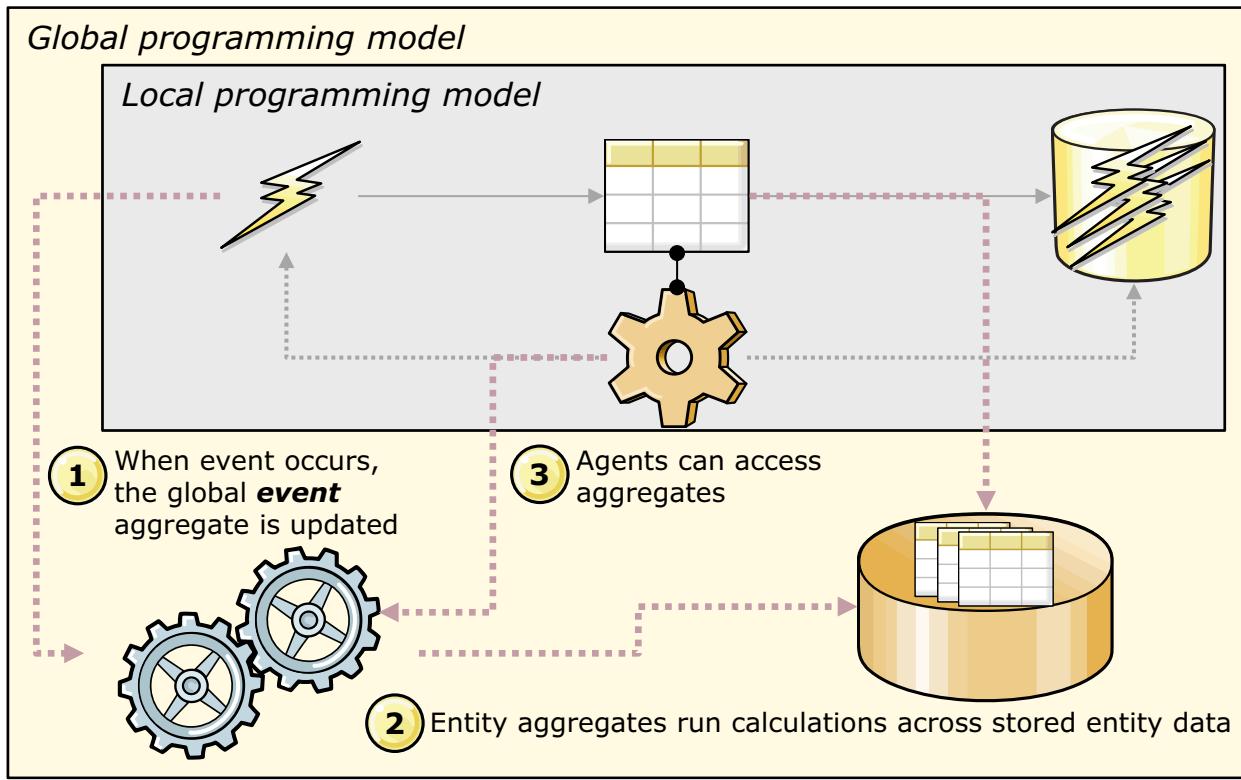


Figure 10-7. Aggregates in the programming model

WB393 / ZB3931.0

Notes:

A solution that uses global aggregates can require large amounts of memory and computer processing capacity. To determine your infrastructure requirements, you must consider many variables, including the extent of event processing that your solution uses and the frequency and complexity of your aggregations. You must also consider the interaction between events and global aggregates, and the size and configuration of your cluster and Insight Server.

10.2. 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.

10.1

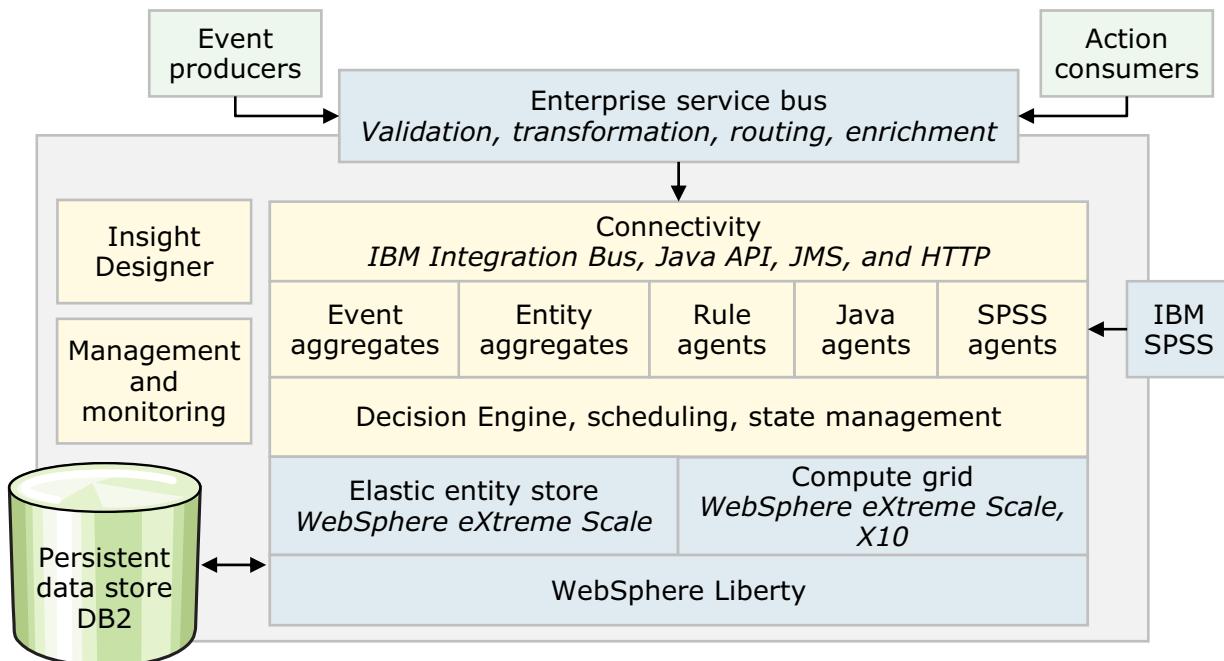
Figure 10-8. Decision Server Insights architecture

WB393 / ZB3931.0

Notes:

Decision Server Insights high-level architecture

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



© Copyright IBM Corporation 2015

Figure 10-9. Decision Server Insights high-level architecture

WB393 / ZB3931.0

Notes:

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

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

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

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

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

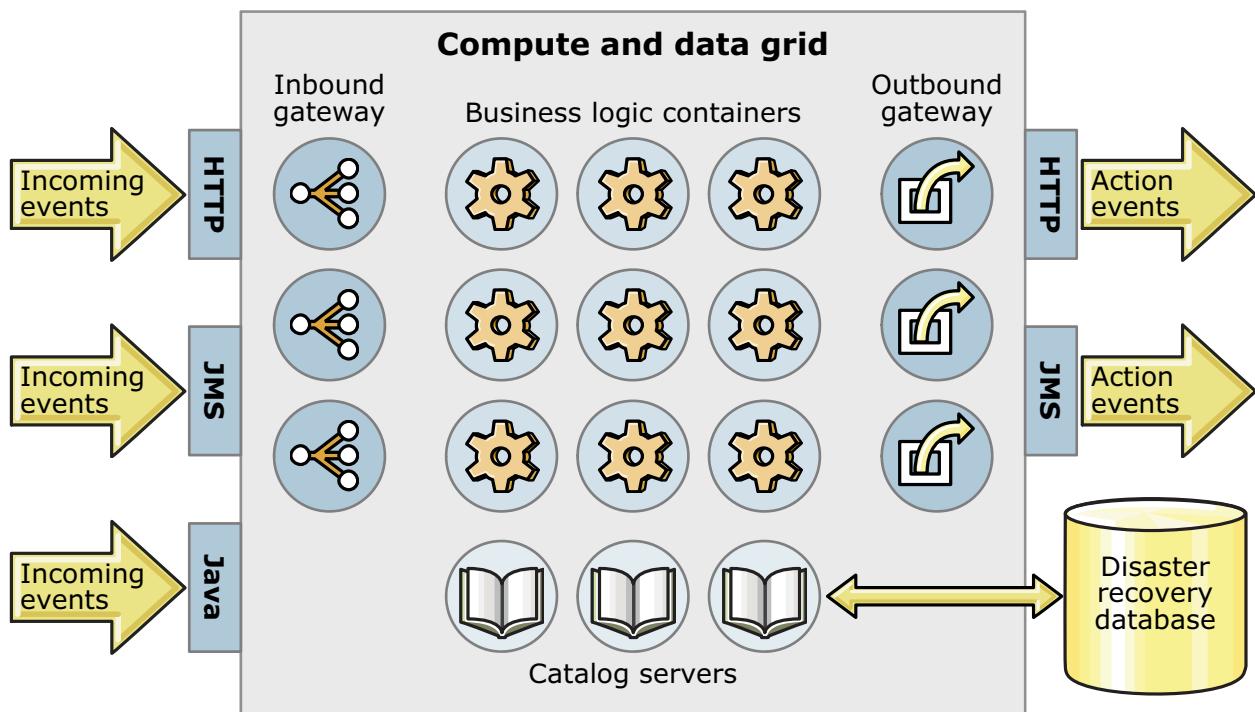
And you do that in the Eclipse-based Insight Designer. Insight Designer helps you easily model the entities and events, implement the business logic agents, and build the aggregates, including easy integration with SPSS scoring models.

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

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

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

Insight Server



© Copyright IBM Corporation 2015

Figure 10-10. Insight Server

WB393 / ZB3931.0

Notes:

The business logic containers are the servers where you store your events and run the agents. The logic includes the components of the programming model (entities, agents, and aggregates), which are deployed as part of a *solution*.

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

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

10.3.Installing Decision Server Insights

Installing 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.

10.1

Figure 10-11. Installing Decision Server Insights

WB393 / ZB3931.0

Notes:



Installing with Installation Manager

- Add Decision Server Insights as a repository

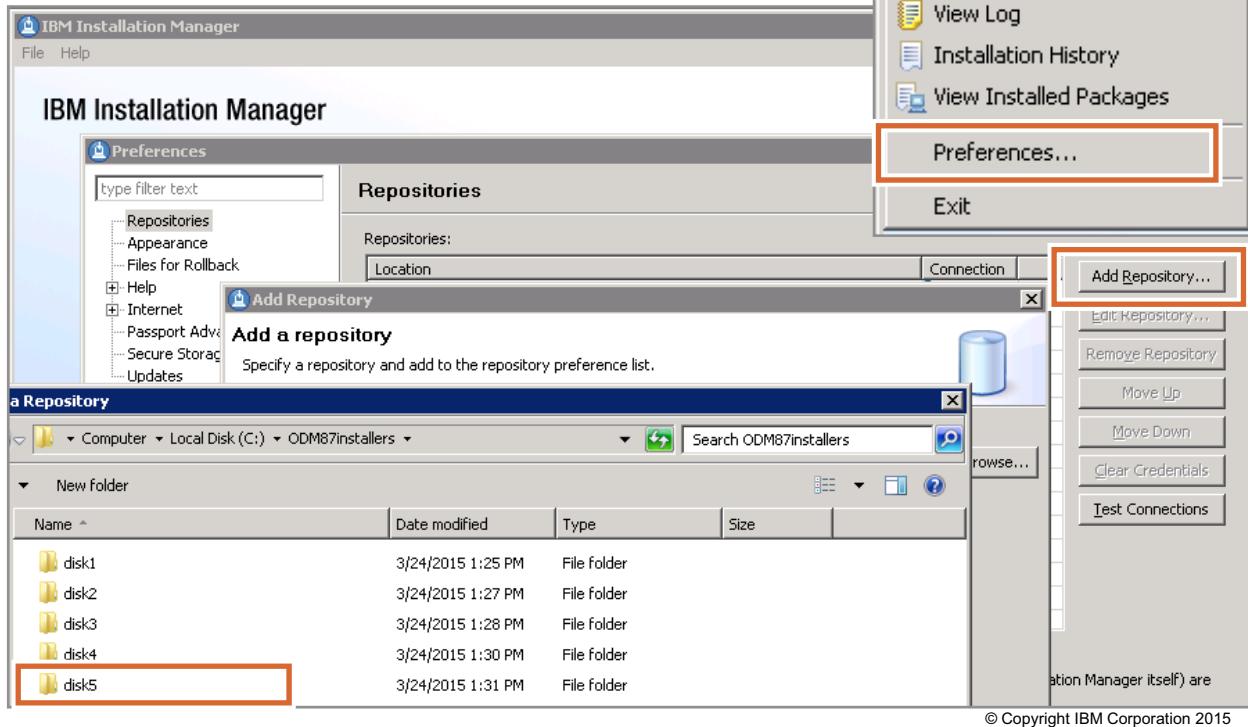


Figure 10-12. Installing with Installation Manager

WB393 / ZB3931.0

Notes:

Decision Server Insights is not installed during installation of Operational Decision Manager Advanced. You must install it separately and in a separate package group.

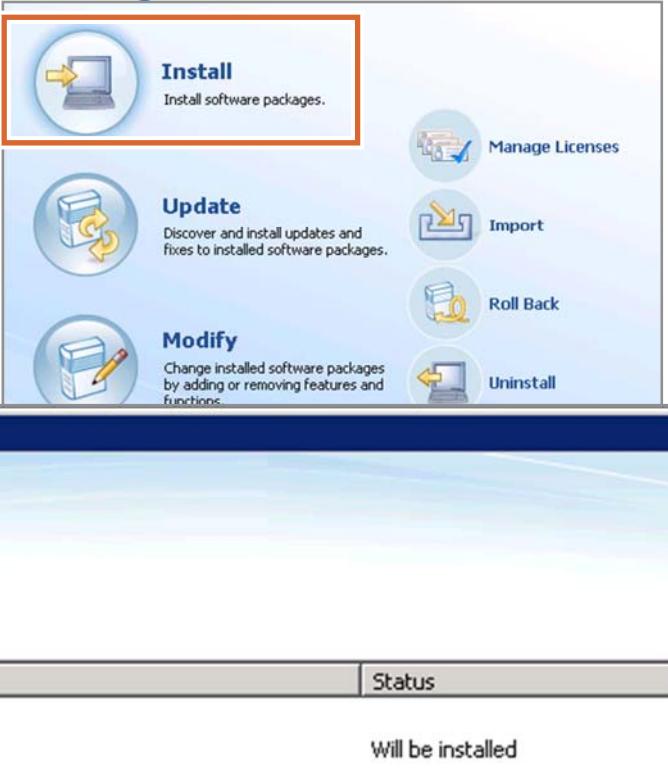
Before you can install Decision Server Insights, you must add the path to the installation files as a repository in Installation Manager.

 WebSphere Education



Installing with Installation Manager console mode

- Click **Install**
- Select Decision Server Insights Version 8.7.1



Installation Packages		Status
<input checked="" type="checkbox"/>  Decision Server Insights		
<input checked="" type="checkbox"/>  Version 8.7.1.0		Will be installed

© Copyright IBM Corporation 2015

Figure 10-13. Installing with Installation Manager console mode

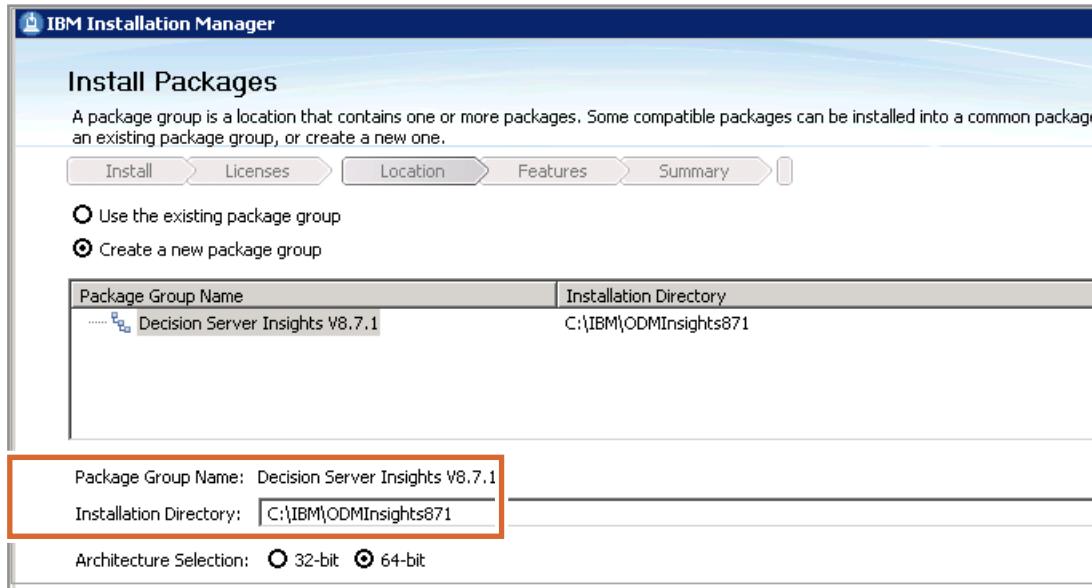
WB393 / ZB3931.0

Notes:



Installing: Defining the installation path

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



© Copyright IBM Corporation 2015

Figure 10-14. Installing: Defining the installation path

WB393 / ZB3931.0

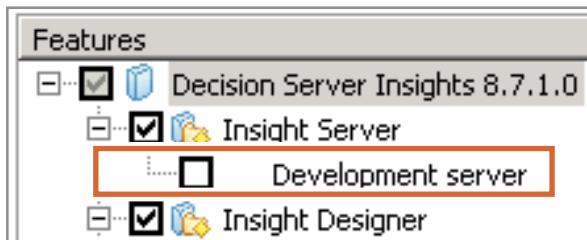
Notes:

 WebSphere Education 

Installing: Choosing the product features

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

- Installing for production
 - Clear all the selected features
 - Select **Insight Server** only
 - Make sure that **Development server** is not selected



© Copyright IBM Corporation 2015

Figure 10-15. Installing: Choosing the product features

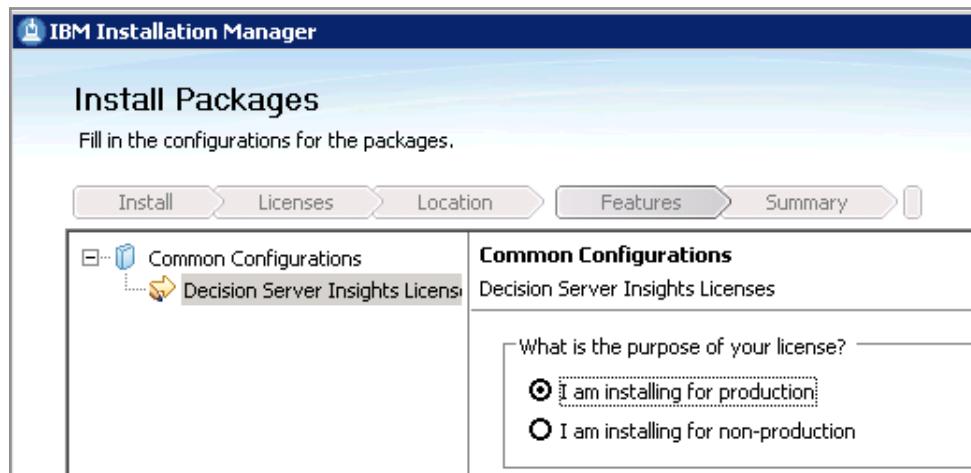
WB393 / ZB3931.0

Notes:



Installing: Choosing the installation type

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



© Copyright IBM Corporation 2015

Figure 10-16. Installing: Choosing the installation type

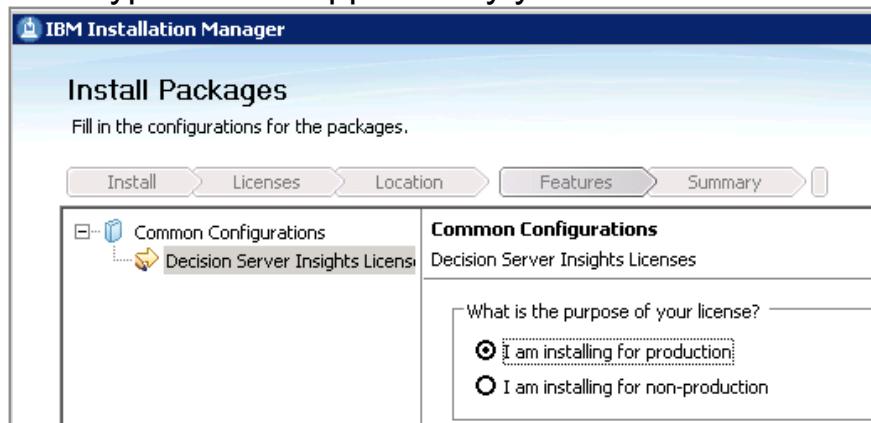
WB393 / ZB3931.0

Notes:

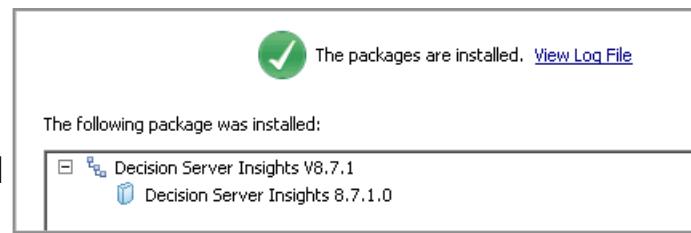
 WebSphere Education 

Installing: Choosing the installation type

- Choose the installation type that is supported by your license



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



© Copyright IBM Corporation 2015

Figure 10-17. Installing: Choosing the installation type

WB393 / ZB3931.0

Notes:



Installing Decision Server Insights on multiple servers

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

Parameter	Description
<code>!CIS_REPOSITORY!</code>	Location of Decision Server Insights installation files
<code>!CIS_PROFILE_ID!</code>	A unique profile name
<code>!CIS_HOME!</code>	Installation path for Decision Server Insights
<code>!CIS_PRODUCTION!</code>	true for production false for non-production
<code>!CIS_FEATURES!</code>	Comma-separated list of features to be installed

© Copyright IBM Corporation 2015

Figure 10-18. Installing Decision Server Insights on multiple servers

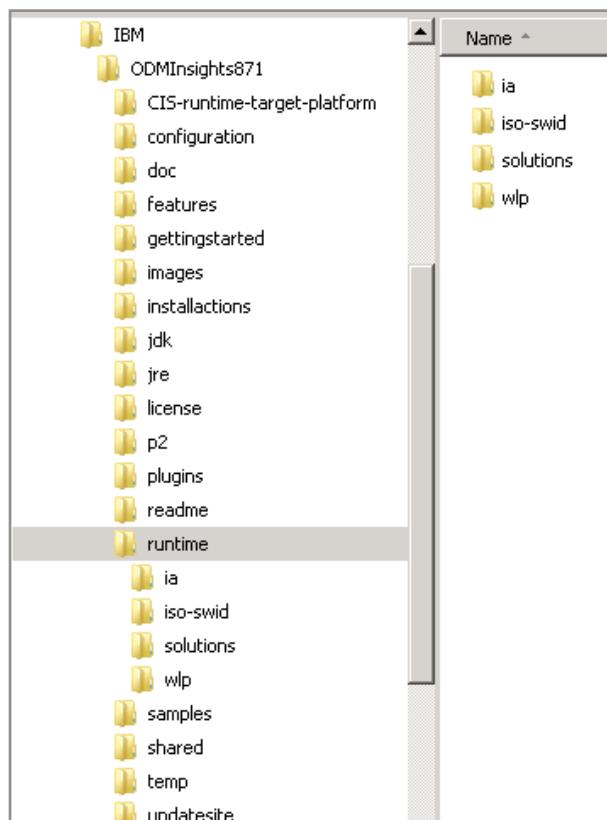
WB393 / ZB3931.0

Notes:



Verifying your installation

- Development environment
 - Full Decision Server Insights installation
 - Includes Insight Designer
- Production environment
 - Install only Insight Server on multiple hosts
- Solutions are deployed to:
 - runtime/solutions/lib
 - runtime/solutions/lib/features
- WebSphere Liberty Profile servers are created in:
 - runtime/wlp/usr/servers



© Copyright IBM Corporation 2015

Figure 10-19. Verifying your installation

WB393 / ZB3931.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Describe Decision Server Insights architecture
- Describe Decision Server Insights installation

© Copyright IBM Corporation 2015

Figure 10-20. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

- 1. True or False:** Decision Server Insights architecture uses WebSphere eXtreme Scale and WebSphere Liberty profile.
- 2. True or False:** A Decision Server Insights solution involves receiving inbound messages as XML files that are processed by the Insights runtime.
- 3. True or False:** Decision Server Insights includes Insight Designer, Insight Server, and Decision Center.

© Copyright IBM Corporation 2015

Figure 10-21. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. **True.**
2. **True.**
3. **False.** *Decision Server Insights does not interact with Decision Center.*

© Copyright IBM Corporation 2015

Figure 10-22. Checkpoint answers

WB393 / ZB3931.0

Notes:



Exercise 12



Installing 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.

10.1

Figure 10-23. Exercise 12

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Prepare the Decision Server Insights silent installation template
- Install Insight Server on multiple hosts

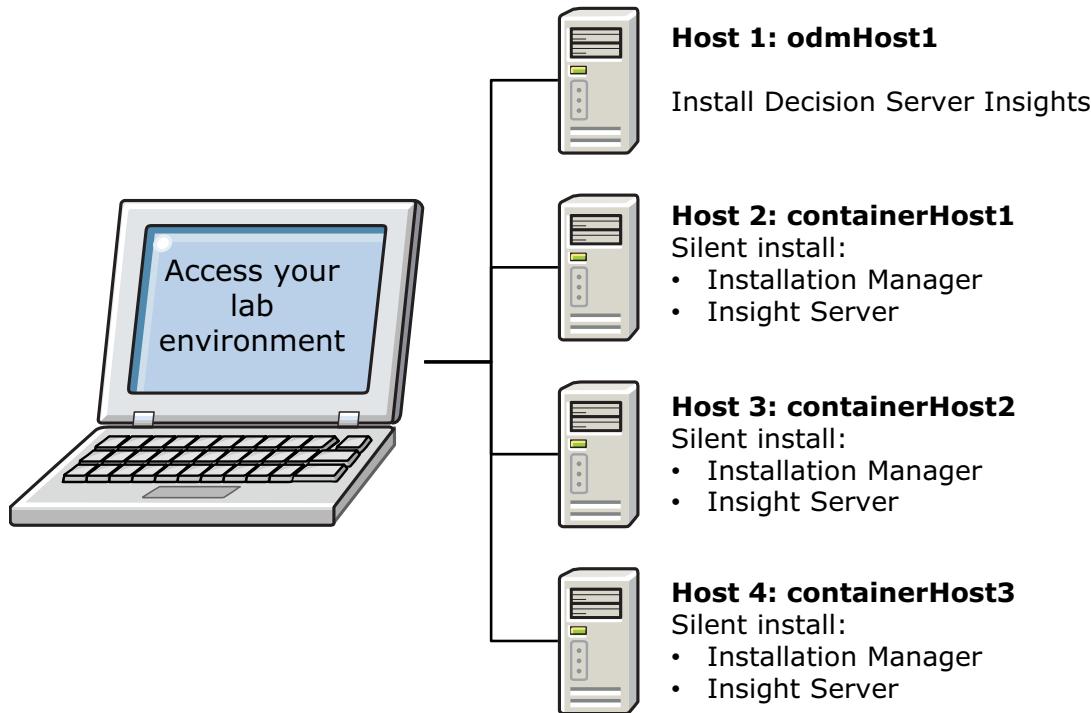
© Copyright IBM Corporation 2015

Figure 10-24. Exercise objectives

WB393 / ZB3931.0

Notes:

Exercise environment (1 of 3)



© Copyright IBM Corporation 2015

Figure 10-25. Exercise environment (1 of 3)

WB393 / ZB3931.0

Notes:

During the lab, you install Decision Server Insights on each of your hosts.



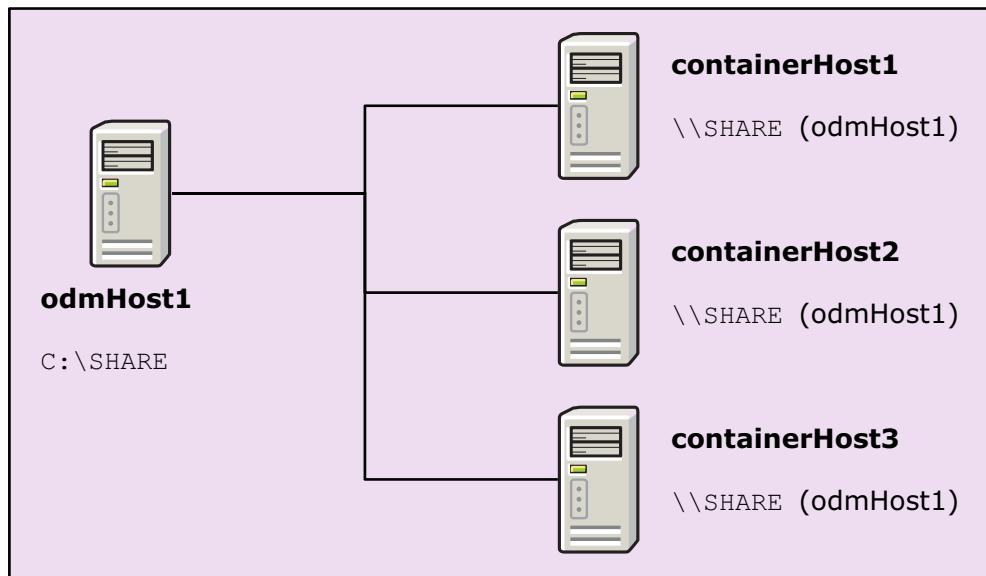
The default host names are: odmHost1, containerHost1, containerHost2, and containerHost3.

Your hosts might be assigned different unique host names. Make sure that you know and use the actual host names for your environment during the exercises.

In the first part of this exercise, you install the complete Decision Server Insights package on your main host (odmHost1).

In the second part of the exercise, you perform a silent installation of Installation Manager and Decision Server Insights (Insight Server only) on Host 2, Host 3, and Host 4, which are your container hosts.

Exercise environment (2 of 3)



© Copyright IBM Corporation 2015

Figure 10-26. Exercise environment (2 of 3)

WB393 / ZB3931.0

Notes:

To transfer files from one virtual machine to another, you use a shared directory.

On your main host (odmHost1), the **C :\ SHARE** folder is used to move files from the main host to the remote host. The other machines have a drive that is mapped to the **SHARE** folder.

Exercise environment (3 of 3)

Main host	Dual core 16 GB RAM	Container 1 host	Single core 8 GB RAM
<i>Default host name:</i> <i>odmHost1</i>		<i>Default host name:</i> <i>containerHost1</i>	
Assigned host name: _____		Assigned host name: _____	
IP: _____		IP: _____	
Container 2 host	Single core 8 GB RAM	Container 3 host	Single core 8 GB RAM
<i>Default host name:</i> <i>containerHost2</i>		<i>Default host name:</i> <i>containerHost3</i>	
Assigned host name: _____		Assigned host name: _____	
IP: _____		IP: _____	

© Copyright IBM Corporation 2015

Figure 10-27. Exercise environment (3 of 3)

WB393 / ZB3931.0

Notes:

This graphic is in the final appendix of your exercise guide. Use that appendix as a reference to make a note of the IP addresses and host names that are assigned to your virtual machines.

Unit 11. Configuring Insight Server

What this unit is about

This unit explains how to configure Decision Server Insights.

What you should be able to do

After completing this unit, you should be able to:

- Describe WebSphere eXtreme Scale basics
- Describe the Decision Server Insights reference topology
- Design and configure a production topology

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Describe WebSphere eXtreme Scale basics
- Describe the Decision Server Insights reference topology
- Design and configure a production topology

© Copyright IBM Corporation 2015

Figure 11-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- eXtreme Scale basics
- Topologies
- Sizing
- Configuring production topology

© Copyright IBM Corporation 2015

Figure 11-2. Topics

WB393 / ZB3931.0

Notes:

11.1. WebSphere eXtreme Scale basics

WebSphere eXtreme Scale basics



© Copyright IBM Corporation 2015

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

10.1

Figure 11-3. WebSphere eXtreme Scale basics

WB393 / ZB3931.0

Notes:



What is IBM WebSphere eXtreme Scale

- eXtreme Scale is an elastic, scalable, in-memory data grid
 - Dynamically processes, partitions, replicates, and manages application data across hundreds of servers
 - Provides transactional integrity and transparent failover
- Principles of extreme scalability
 - Put data in memory
 - Partition the data to enable linear horizontal scale-out
 - Caching

© Copyright IBM Corporation 2015

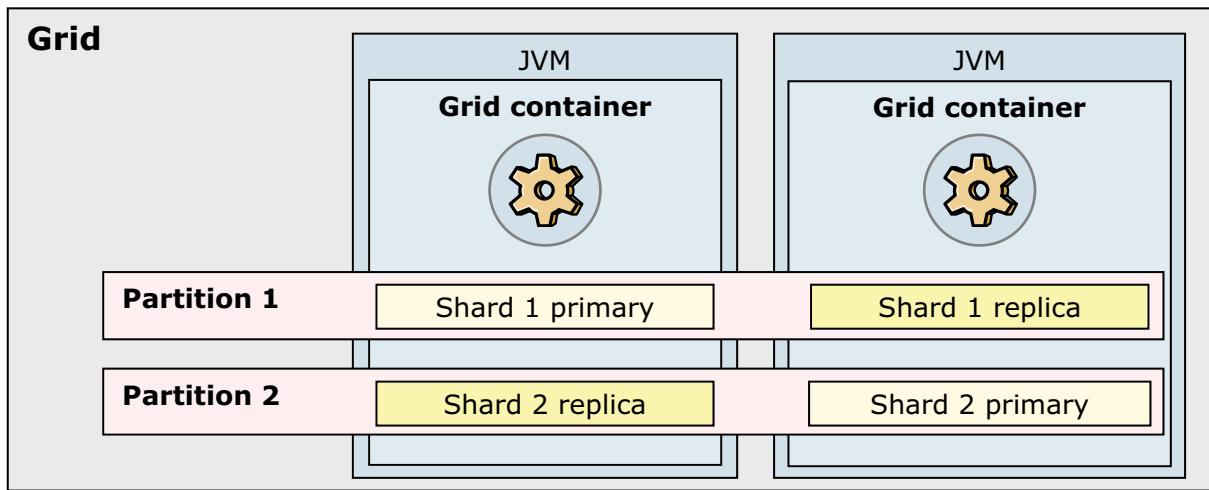
Figure 11-4. What is IBM WebSphere eXtreme Scale

WB393 / ZB3931.0

Notes:

Understanding a grid

- *Grids* divide the data into partitions
- Each *partition* holds an exclusive subset of the data
- Within the partition, the data is stored in *shards*
 - Primary shard contains the primary copy of the subset of data
 - Replica shards contain copies of the primary shard



© Copyright IBM Corporation 2015

Figure 11-5. Understanding a grid

WB393 / ZB3931.0

Notes:

Grids, partitions, and shards are the major building blocks when you do capacity planning for WebSphere eXtreme Scale. Maps are important as a way for you to determine the size of the data that you store in the grid. You need to calculate the data that is stored in each map in your grid to arrive at a total size of data that you want to store.

- A *grid* divides the data set into partitions
- Each partition holds an exclusive subset of the data. The data can be partitioned based on the key and the data for a partition is stored at run time in a set of shards.
- A shard represents a partition that is placed on a *container*. Each partition has an instance that is a *primary shard* for the primary copy of the data. You can also configure a number of *replica shards*. The replica shards are either synchronous or asynchronous.

The relationship between the grid, partitions, and shards is illustrated on the slide.

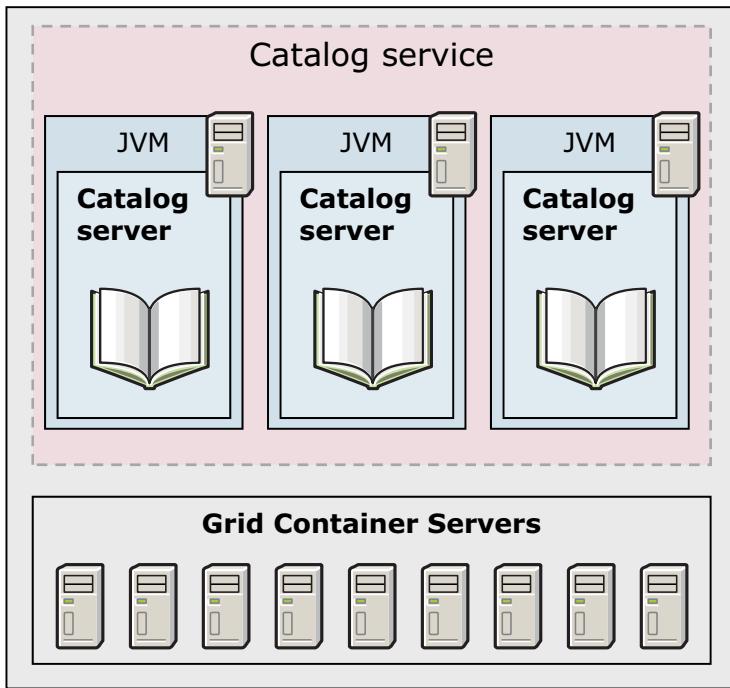
A *container server* physically holds the application data for the grid.

A grid has one or more partitions. Each partition is represented by a primary shard that is hosted on a container, and optionally one or more replica shards that are hosted on other containers.

Any number of containers can be run on a host (vertical scaling) and any number of hosts can run extra containers (horizontal scaling). A key feature of an eXtreme Scale grid is that more containers can be added dynamically on any server in the network.

Catalog service

- Catalogs become central nervous system of the grid



Catalog service:

- Keeps track of partitions and shards
- Redistributes shards when a container joins or leaves the grid
- Monitors grid health
- Provides data location services to grid clients
- Enforces policies and rules
- Ensures high availability and group service

© Copyright IBM Corporation 2015

Figure 11-6. Catalog service

WB393 / ZB3931.0

Notes:

The *catalog service* parses the deployment policy and grid configuration files. It uses their definitions to control placement of shards across the available container servers in the grid. It also discovers and monitors the health of the containers, automatically balancing shard placement as necessary when a new container is added to the configuration, or if a container is stopped.

One or several catalog servers can run in your grid environment. However, only one is chosen automatically as the *master catalog*. The catalog service is designed to service hundreds or thousands of container servers.

WebSphere eXtreme Scale uses logical placement rules (implemented by the catalog server) to ensure that replica shards are held in containers that are running on different host machines than the primary shard. WebSphere eXtreme Scale determines the types and placement of replica shards by using a deployment policy, which specifies the minimum and maximum number of synchronous and asynchronous replicas.

WebSphere eXtreme Scale terminology (1 of 3)

Term	Definition
Map	A cache that stores Java objects based on key-value pairs.
Mapset	A collection of logically related maps that can be partitioned and replicated over a number of servers.
Grid	A collection of mapsets that might span multiple Java virtual machines and that you can connect to and access data.
Partition	Logical representation of a subset of the data in the mapset, plus any replicas that each subset might have. <ul style="list-style-type: none"> • The number of partitions (n) is a configurable attribute of a mapset • Partition numbering starts at 0 • The mapset data is distributed across the n partitions
Replica	A copy of the primary data that is stored remotely about the primary and other replicas. <ul style="list-style-type: none"> • <i>Synchronous replica</i>: Updated transactionally when the primary is updated to ensure no data loss when the primary data is lost. • <i>Asynchronous replica</i>: Updated after the transaction is complete for faster transaction performance, but with increased risk of data loss in the face of failures.

© Copyright IBM Corporation 2015

Figure 11-7. WebSphere eXtreme Scale terminology (1 of 3)

WB393 / ZB3931.0

Notes:

WebSphere eXtreme Scale terminology (2 of 3)

Term	Definition
Shard	Provides the physical memory storage for the contents of a partition. <ul style="list-style-type: none"> • <i>Primary shard</i>: Contains the primary partition. • <i>Replica shards</i>: Backup of all the data in the primary shard.
Grid container	A container for the shards (all the cached data).
Container server	A Java virtual machine that runs WebSphere eXtreme Scale and hosts one or more grid containers. Decision Server Insights uses a WebSphere Application Server Liberty profile as its grid container servers.
Catalog server	Provides management of the entire grid. When there is more than one catalog server, one of them is the <i>master</i> or <i>primary</i> catalog server, and coordinates work among the servers to provide the catalog services.
Catalog service	<ul style="list-style-type: none"> • Keeps track of partitions and shards. • Redistributions shards when a container joins or leaves the grid. • Monitors the health of the grid. • Provides data location services to grid clients.

© Copyright IBM Corporation 2015

Figure 11-8. WebSphere eXtreme Scale terminology (2 of 3)

WB393 / ZB3931.0

Notes:

For Decision Server Insights, the preferred grid topology has only one grid container per grid container server. In addition, only one grid container server runs on each host machine. For this reason, this tutorial uses the term *container* to refer to both the grid container server and grid container, except in cases where appropriate differentiation is required.

WebSphere eXtreme Scale terminology (3 of 3)

Term	Definition
Catalog service domain	The group of catalog servers, together with the group of container servers that they oversee.
Quorum	<p>An agreement between members of the catalog server group on what needs to be done for grid lifecycle operations.</p> <ul style="list-style-type: none"> • For example, when a network brownout occurs, communication between catalog servers might be lost, and more than one catalog server becomes the primary server; known as <i>split-brain syndrome</i>. • If a split-brain syndrome occurs when quorum is enabled, grid work is suspended. • Recovery from the loss of quorum typically requires manual intervention.
Majority quorum	Ensures that quorum is achieved and grid work can be performed while more than half of the catalog service members are active and aware of each other.
Catalog server cluster endpoints	Configured for container servers to establish a communications link with the catalog servers. These endpoints become part of the catalog service domain (which means they are part of the grid).

© Copyright IBM Corporation 2015

Figure 11-9. WebSphere eXtreme Scale terminology (3 of 3)

WB393 / ZB3931.0

Notes:

11.2. Topologies

Topologies



© Copyright IBM Corporation 2015

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

10.1

Figure 11-10. Topologies

WB393 / ZB3931.0

Notes:

Designing a topology

- Consider:
 - How many servers of each type you require, and allocate unique names to them
 - Whether to configure persistence for your grid data
- Inbound and outbound connectivity servers
 - Use one of each?
 - HTTP or JMS?
- Catalog service
 - High availability requires 3 or more catalog servers
 - Each catalog server should be on a separate machine
- Container servers
 - High availability requires a minimum of 3 servers
 - How many solutions must run in the grid?
 - What is the volume of events that must be processed?
 - How many partitions must be allotted for the primary data and replicas?

© Copyright IBM Corporation 2015

Figure 11-11. Designing a topology

WB393 / ZB3931.0

Notes:

Reference topology goals

- The grid must be highly and continuously available in a normal operation mode
 - Must withstand the loss of one container server without any loss of data and without loss of access to data
 - Must tolerate the controlled shutdown of one container at a time for the purpose of applying "rolling updates" or for any other maintenance activity
 - Should accept and use new containers (within a limit that is determined by the configured number of partitions and replicas)
 - Should tolerate the simultaneous loss of 2 containers with the accepted risk of some data loss
- Catalog service should be highly available and rely on *majority quorum*
- Grid data must be recoverable in the case of disaster or in the case of a controlled shutdown of the grid
- The system should tolerate the failure of at least one inbound and one outbound connectivity server
- Event throughput should not suffer significantly as a result of the previous requirements

© Copyright IBM Corporation 2015

Figure 11-12. Reference topology goals

WB393 / ZB3931.0

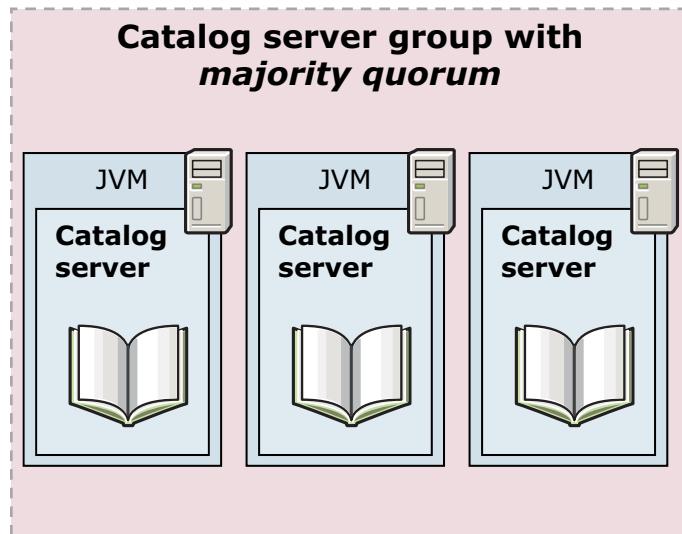
Notes:

To meet the goals of high and continuous availability, the grid must be able to support loss of a container without losing data or access to data.

Grid data must be recoverable in the case of a disaster where more than two containers are lost, such as during a power outage. If there is a need for controlled shutdown of the grid, such as when increasing the number of partitions, the data must be recoverable.

Catalog service

- For high availability: 3 catalog servers on 3 separate hosts



© Copyright IBM Corporation 2015

Figure 11-13. Catalog service

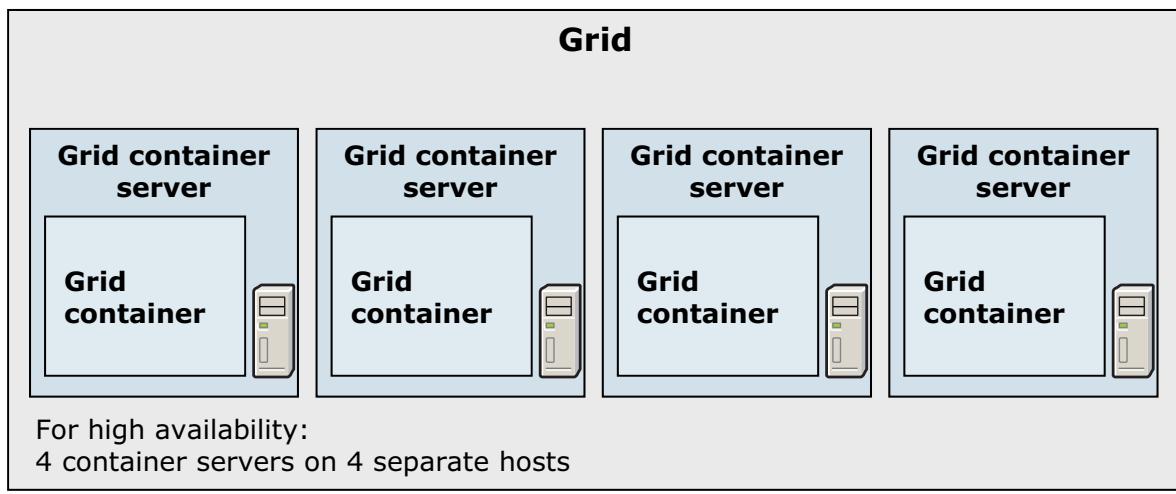
WB393 / ZB3931.0

Notes:

In a normally functioning grid, there is little work for a catalog server to do, and so it has little influence on scalability.

Containers and partitions (1 of 2)

- For high availability, use a minimum of 3 container servers
 - Use 4 container servers to allow for one server to be shut down for upgrade or maintenance
- Each container should run on a single host
- Each host should have a minimum of two cores



© Copyright IBM Corporation 2015

Figure 11-14. Containers and partitions (1 of 2)

WB393 / ZB3931.0

Notes:

The number of required container servers depends on the number and type of solutions that are running on the grid and the volume of events that they process. This number can grow or shrink.

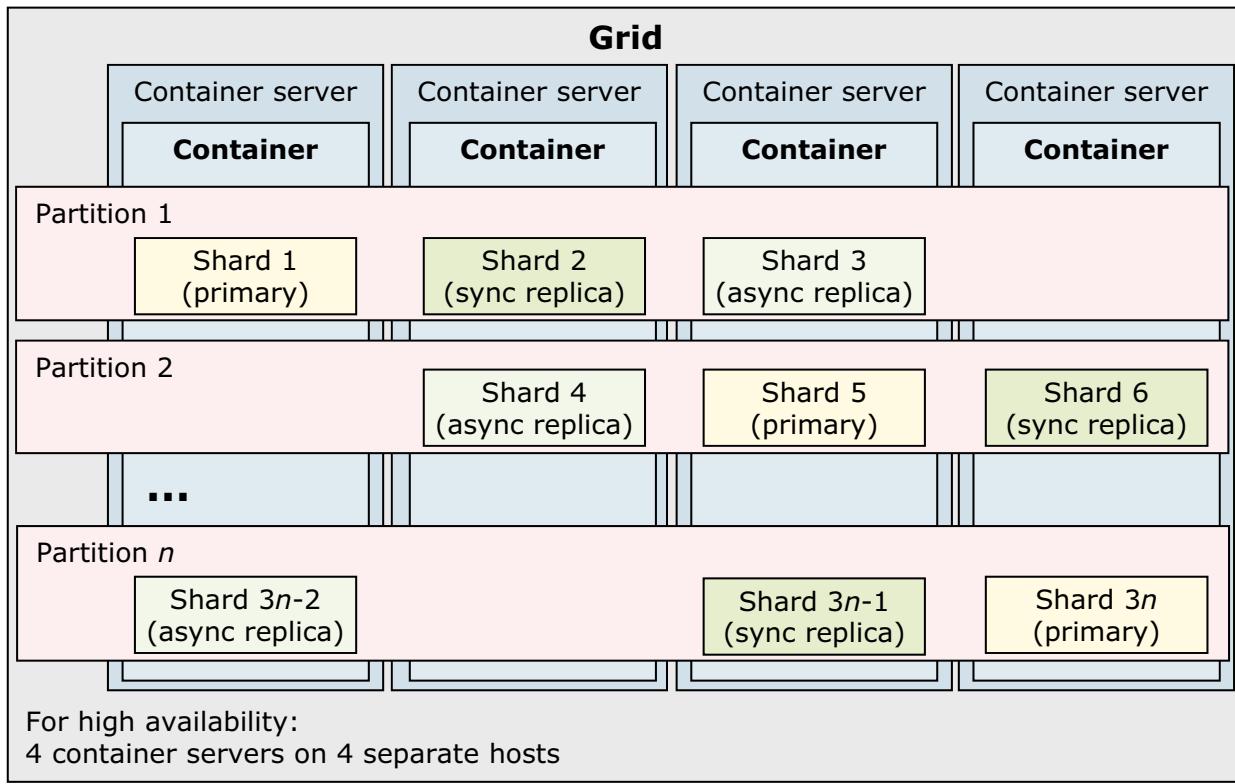
From the topology goals, you can infer that, in addition to the primary data, you should have a synchronous replica and an asynchronous replica. If the primary data is lost, the synchronous replica immediately becomes the primary, and the asynchronous replica becomes synchronous. This configuration meets high availability goals and ensures that maintenance can be performed on one container at a time. A suggested practice is to use four containers that run on four separate hosts, regardless of throughput.

The maximum number of containers that you can dynamically add to the grid is constrained by the number of partitions and the number of replicas that you configure for each partition. The number of partitions is configured when you set up the grid and can be changed only when the grid is down. You do not need more container servers than the maximum number that can be used to allocate partitions.

Having many partitions can facilitate growth and better balancing of memory resources, but also means more threads, increased use of grid communication resources, and greater level of

balancing work. When you have many partitions but few containers, adding or removing containers involves a significant amount of grid reconfiguration work.

Containers and partitions (2 of 2)



© Copyright IBM Corporation 2015

Figure 11-15. Containers and partitions (2 of 2)

WB393 / ZB3931.0

Notes:

The minimum suggested number of cores is two per server to enable faster start, stop, and management of external operating system (OS) requests without interrupting the workload.

Because the suggested practice is to use one container server per machine, you can determine the number of partitions to use with this equation: $C \times M \times 2$

Where:

- C is the maximum number of containers that you expect to have
- M is the number of cores of the container host

This equation assumes that you have an equal number of cores in each container host machine.

Certain eXtreme Scale operations are more efficient if the number of partitions is a prime number, so a good practice is to round the number of partitions up to the next prime number.

Example:

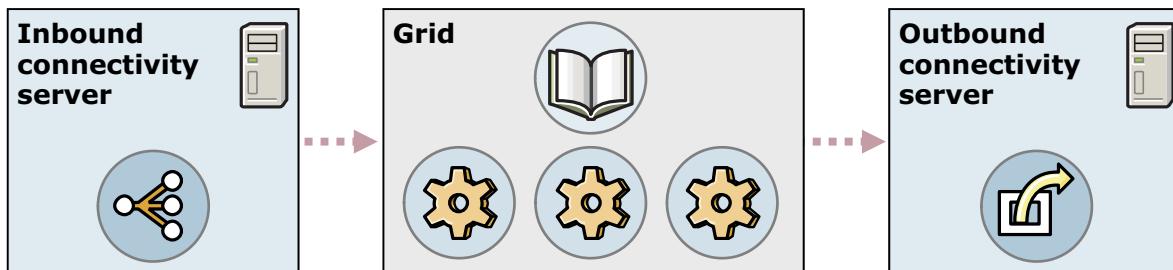
- Assume that you have a maximum of 6 container server hosts (C = 6), each with 4 cores (M = 4).

- Number of partitions = C x M x 2 = 6 x 4 x 2 = 48
- Round the result up to the next prime number: 53

The default product configuration for the number of partitions is 127.

Inbound and outbound connectivity

- Minimum configuration:
 - One inbound server on one physical host
 - One outbound server on a separate physical host



- For high availability, use two nodes each for inbound and outbound connectivity
 - You can reduce to two nodes, each with an inbound and outbound connectivity server
 - To scale vertically, add more servers to the same host
 - To scale horizontally, add more hosts

© Copyright IBM Corporation 2015

Figure 11-16. Inbound and outbound connectivity

WB393 / ZB3931.0

Notes:

Connectivity servers manage event flow to and from the system. Redundancy is required to avoid a single point of failure.

Inbound and outbound connectivity servers are responsible for the event flow in and out of the system, so there should be some redundancy for these servers. A good starting point is to use one of each on two different machines, which is the minimum requirement to avoid a single point of failure.

There is no rule for determining the actual number of inbound or outbound servers that are required for a particular event throughput. The number of inbound and outbound servers varies according to several parameters, including:

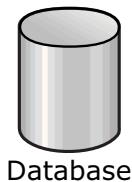
- Size of the event data
- Type of event transformations that might be performed
- Whether event persistence is used for Messaging Service (JMS)
- Type of protocol used (for example, HTTP or JMS)

For best results, start with the two servers and monitor resource usage (for example, CPU and memory).

For high-event rates, you can add inbound and outbound connectivity servers to the same machine until the machine's ideal resource consumption threshold is reached. After the threshold is reached, you can add new machines if required.

Data persistence

- In a single site topology, persist grid data to a database
 - Enable recovery from a disaster
 - Enable any rare maintenance activities that require the grid to be shut down
- Synchronous mode (write-through)
 - High performance penalty
 - Ensures no data loss
- Asynchronous mode (write-behind)
 - Performs database updates in batches
 - Risk of some data loss
- Reference topology uses asynchronous persistence



Database

© Copyright IBM Corporation 2015

Figure 11-17. Data persistence

WB393 / ZB3931.0

Notes:

Decision Server Insights operates on an in-memory data grid. However, persistence of grid data is required to enable disaster recovery or maintenance shutdown of the grid.

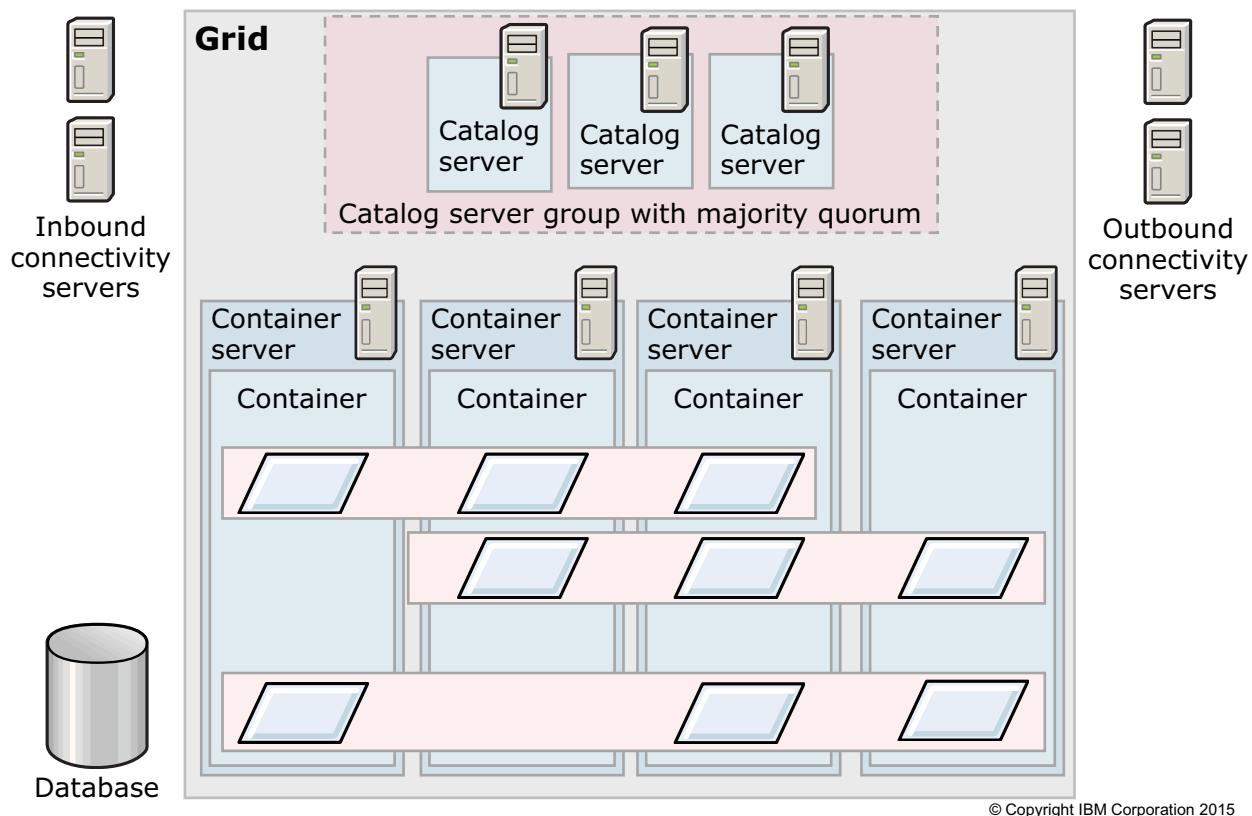
You can choose between synchronous (write-through) or asynchronous (write-behind) database persistence modes. Synchronous persistence typically implies a high performance penalty.

Asynchronous persistence performs the database updates in batches and carries a small performance penalty. The frequency of database updates might vary depending on the level of accepted risk and it is set in terms of a time interval and the number of batched updates that use a parameter that is called `writeBehind`. For example, if you set the `writeBehind` value to `T20;C200` then a write to the database happens every 20 seconds or every time the number of pending updates reaches 200, whichever condition happens first.

The reference topology uses write-behind persistence mode to avoid the performance cost and throughput penalty of the synchronous persistence option.

WebSphere eXtreme Scale can queue the database updates, allowing for some database downtime, but ideally the database should be highly available (for example, by using DB2 high availability disaster recovery). For high event loads, consider using a highly scalable database to ensure that the database does not become a bottleneck.

Decision Server Insights reference topology



© Copyright IBM Corporation 2015

Figure 11-18. Decision Server Insights reference topology

WB393 / ZB3931.0

Notes:

The reference topology uses a different physical server for each Decision Server Insights server in the topology. This configuration provides a high degree of isolation between components.

The high availability and recoverability objectives can be met with a minimum of three physical servers, not counting the database. However, a topology with four servers would be more appropriate as a minimal high availability topology. The reference topology uses four container servers, one on each host machine. Three machines can have a catalog server. Additionally, you should have inbound and outbound servers on at least 2 of the 4 physical servers.

11.3. Sizing

Sizing



© Copyright IBM Corporation 2015

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

10.1

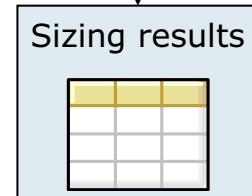
Figure 11-19. Sizing

WB393 / ZB3931.0

Notes:

Capacity planning for containers in the grid

- Steps to estimate the hardware, JVM, and grid configuration
 1. Collect the information about the data (Java objects) to be stored
 2. Determine the memory requirements and partition count
 3. Determine the CPU size and server count
- Sizing results include:
 - Number of servers (physical machines)
 - Number of CPUs
 - Total physical memory
 - Number of containers
 - Maximum heap size per JVM (Xmx)
 - Number of partitions
- JVM recommendation for each container: 60% heap utilization



© Copyright IBM Corporation 2015

Figure 11-20. Capacity planning for containers in the grid

WB393 / ZB3931.0

Notes:

To compute the container memory requirements, catalog servers should not share the host with containers.

Calculating the required memory for the data in the grid

- To determine the maximum size of the data to be stored in the grid
 - Estimate the maximum number of objects that are needed at usage peaks (*numberOfObjects*)
 - Estimate the average size of each object (*averageObjectSize*)
- Use values for total object data to be stored, event data, and number and type of replicas to determine physical requirements

Sizing variable	Calculation
Total object data to be stored	$\text{averageObjectSize} * \text{numberOfObjects} * (1 + \text{number of replicas})$
Maximum memory per container	$\text{maxHeapSizePerJVM} * \text{maxHeapUsage} - \text{ExtremeScale footprint}$
Minimum number of containers	$\text{totalObjectMemory} * \text{containerObjectMemory}$
Total physical memory for grid	$\text{physicalMemoryPerJVM} * \text{numberOfContainers}$
Container object memory	Total object memory / Maximum number of containers

© Copyright IBM Corporation 2015

Figure 11-21. Calculating the required memory for the data in the grid

WB393 / ZB3931.0

Notes:

To compute container object memory, you must know how much memory the object data (primary and replica) uses in each container JVM.

Make sure that your estimate is based on real data.

Determining number of partitions

- Set the number of partitions (P) to a prime number
 - Maximum number of containers (C)
 - Number of cores per container host (M)

Number of partitions = $(C * M * 2)$ rounded up to next prime number

- You must limit the number of containers that are running to match the number of partitions
 - If number of partitions is P, you cannot use more than P containers to host data
 - Additional containers start but do not receive any partition to host
- One container per host of physical server
- Take into account the number of cores on the host machine
 - With more cores, increase the number of partitions to increase the work load on each machine

© Copyright IBM Corporation 2015

Figure 11-22. Determining number of partitions

WB393 / ZB3931.0

Notes:

If feasible, use one container per host of physical server so that failure of a machine does not bring down more than one container. However, it is common to have multiple containers per server in cases where there are at least 3 or more servers.

Example: Calculating memory storage

- Assumptions for the solution:
 - Number of primary objects to be stored: 18
 - Average size of primary objects: ~10K per object
 - Total size of PrimaryObjectMemory: 18 GB
 - Number of required replicas: 1
 - Physical host machines: 4
 - Footprint for WebSphere eXtreme Scale (WXS) + WebSphere Application Server: 100 MB
 - Maximum heap size per JVM =

$$(\text{Container object memory} + \text{WebSphere eXtreme Scale / WebSphere Application Server footprint}) * 1.7 \text{ (60\% utilization)}$$

Step	Calculations	Data (GB)
1	Total object memory = primary object memory * 2 (primary data + 1 replica)	36
2	Container object memory: total object memory / number of containers (4 hosts)	9
3	Maximum heap size per JVM = (9 GB + 0.1 GB) * 1.7	15.47

© Copyright IBM Corporation 2015

Figure 11-23. Example: Calculating memory storage

WB393 / ZB3931.0

Notes:

For this example, the solution must store object data for 18 types of primary objects. This data includes event data that the customer wants to keep a history of. Other event data is deleted. The average size per object is 10 KB.

1. To calculate the total amount of memory that must be stored across all containers, you calculate the amount of memory that is required for the objects that are processed by your grid. You then multiply that value by the number of replicas that you require.
2. The amount of memory that you need to store in each container is calculated by dividing the total object memory by the number of containers that you can use. In this example, the organization allotted a maximum of 4 physical hosts, and the plan is to use one container per host.
3. The maximum memory that can be used per container is limited by the JVM heap size. The container memory should not exceed 60% usage of the heap. To ensure that you remain under 60%, you can multiply the total content memory value by 1.7.

Based on the calculations for this example, by using 4 containers, one per physical host, the maximum heap size is almost 16 GB, which is too large for a 64-bit operating system.

To accommodate this memory requirement, more JVMs are needed to reduce the load per JVM.

As a general guideline, keep the maximum heap size for each JVM to under 5-6 GB.

Example: Determining heap size

- Goal: Maximum heap size should be 5-6 GB per JVM recommendation
 - Increase the number of containers from 4 to 11
 - All containers to run on the 4 physical hosts

Step	Calculations	Data (GB)
4	Maximum heap size per JVM <ul style="list-style-type: none"> containerObjectMemory = 36 GB / number of containers (11) = 3.2727 GB maxHeapSizePerJVM = (3.2727 + WebSphere eXtreme Scale / WebSphere Application Server footprint (.1 GB)) * 1.7 = 5.7336 GB Result = Next highest multiple of 256 MB 	6.144
5	Operating system (OS) tax <ul style="list-style-type: none"> For a 64-bit Linux platform: 0.5 GB Other native methods or tools running on the OS: ~0.5 	1
6	Physical Memory per JVM: OS tax + maximum heap size = 7.144 ; round up to 7.5	7.5
7	How many JVM containers on each server? <ul style="list-style-type: none"> maxNumberJVMsPerServer = 32 GB available memory per server / 7.5 GB physicalMemoryPerJVM = 4 JVMs per server numberOfServers = 12 JVMs / 4 JVMs per server 	4

© Copyright IBM Corporation 2015

Figure 11-24. Example: Determining heap size

WB393 / ZB3931.0

Notes:

MaxHeapUsage is 60% heap utilization. To represent the 60% heap utilization factor, this calculation uses a 1.7 multiplier.

- To keep the heap size within the suggested range of 5-6 GB, the topology would require 11 containers as a minimum.

The number of physical hosts is still 4, but multiple containers can run on a single host.

You divide the total object memory by 11 containers. The result is the amount of memory to be stored on each container (containerObjectMemory), which is then added to the footprint and multiplied by 1.7 to determine maximum heap size.

The final result can then be rounded up to the next highest multiple of 256 MB. In this example, the next highest multiple is 6.144 GB (which is 256 * 24).

- In addition to the calculation for the grid object memory, you must also account for memory usage by the operating system (OS) and other native methods or tools. Use 1 GB as the OS tax value.
- The physical memory per JVM is calculated by adding the OS tax to the maximum heap size.

7. To determine how many containers to run per server, you divide the available memory per server by the physical JVM memory. The result shows that you can host 4 container JVMs on a single host.

To allow for an even distribution of 4 containers per host across the physical hosts, the number of containers is rounded up to 12. So only 3 physical hosts are required to store the total object memory.

Example: Planning for failover

- Consider failure and maintenance shutdown
 - Physical memory per JVM requires 4 JVM per server
 - Total containers: 11, rounded up to 12 for even distribution of the 4 JVMs
 - Physical hosts required: 3
- To plan for failure or maintenance, add one host that also runs four containers

Step	Calculations	Data (GB)
8	Failure scenario or maintenance shutdown: Loss of 1 JVM <ul style="list-style-type: none"> • $12 - 1 = 11$ • $11 * \text{containerObjectMemory} (3.2727 \text{ GB})$ 	35.9997
9	Recommendation to ensure performance: Add 1 host to run 4 additional containers <ul style="list-style-type: none"> • Total physical hosts: 4 • Total containers: 16 • Use previously calculated maxHeapSizePerJVM and physicalMemoryPerJVM values, so no change in physical memory per server 	

© Copyright IBM Corporation 2015

Figure 11-25. Example: Planning for failover

WB393 / ZB3931.0

Notes:

8. The failure and maintenance scenarios account for the failure of a single JVM. In this case, because there are 12 JVMs, losing one JVM means that the 11 remaining containers must be able to hold the object memory. Each container is expected to hold 3.2727 GB. The maintenance scenario accounts for shutdown of one server at a time, if required, without affecting the grid.
9. To ensure grid performance, the recommendation is to increase the number of physical hosts from 3 to 4. Each host can run 4 container servers. The grid would now have 16 container JVMs, which continue to use the previously calculated heap size and physical memory values. The amount of physical memory per server is unchanged.

Example: Number of partitions

- Number of partitions = $C * M * 2$
 - C: Maximum number of containers
 - M: Number of cores on the container host
 - To account for growth: Multiply result by 2
 - Final result is rounded up to next prime number

Calculations	Data (GB)
C	16
M	2
$(C * M * 2) * 2$	64
Number of partitions (result rounded to next prime number)	67

© Copyright IBM Corporation 2015

Figure 11-26. Example: Number of partitions

WB393 / ZB3931.0

Notes:



Example: Sizing results

Results	Extended growth: 7+ years
Number of servers (physical machines)	4
Number of CPUs	$4 * 2 = 8$ cores
Total physical memory	32 GB x 4 = 128 GB
Number of WebSphere eXtreme Scale container JVMs	16
MaxHeapSize per JVM, also known as Xmx	6144 MB or 6.144 GB
numberOfPartitions	67

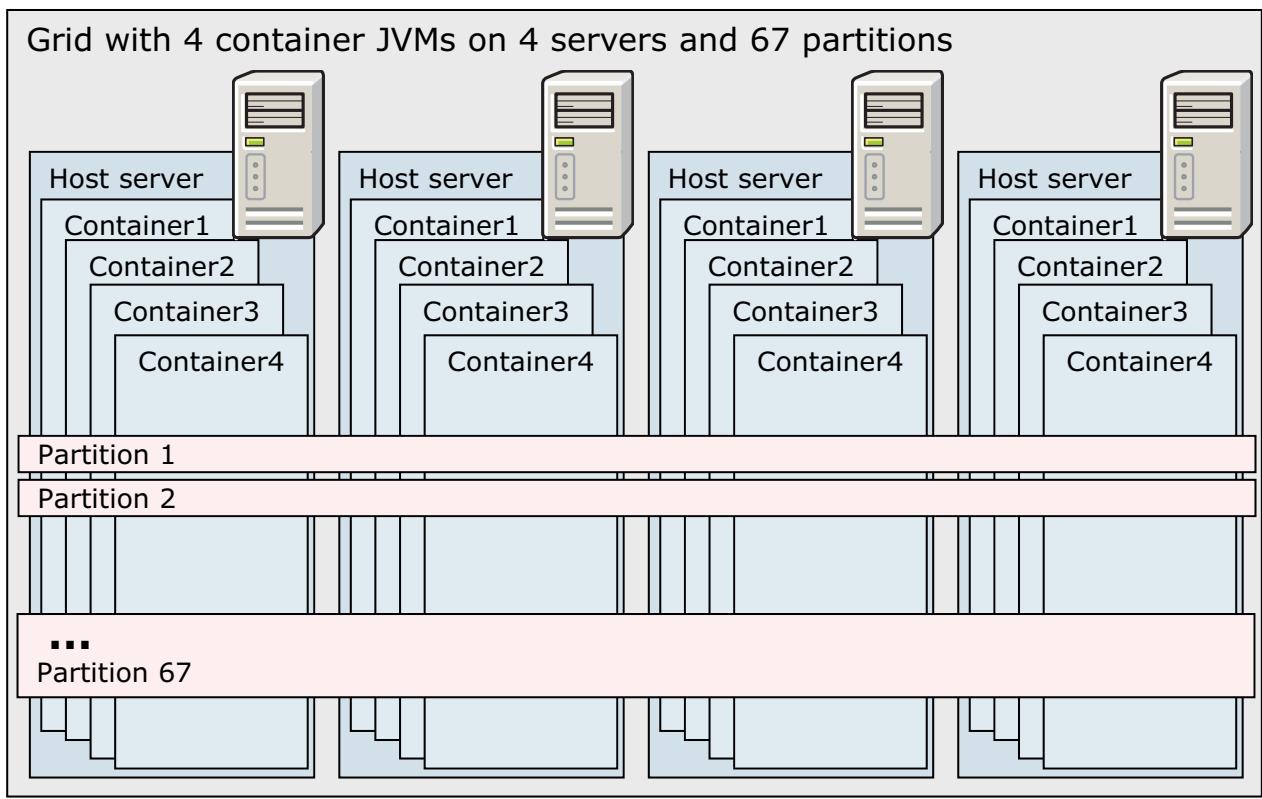
© Copyright IBM Corporation 2015

Figure 11-27. Example: Sizing results

WB393 / ZB3931.0

Notes:

Example: Resulting grid



© Copyright IBM Corporation 2015

Figure 11-28. Example: Resulting grid

WB393 / ZB3931.0

Notes:

11.4. Configuring a production topology

Configuring a production topology



© Copyright IBM Corporation 2015

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

10.1

Figure 11-29. Configuring a production topology

WB393 / ZB3931.0

Notes:

Steps for creating and configuring a production server topology

1. Install IBM Installation Manager on each machine where you want to run a server



2. Install Decision Server Insights on each machine where you want to run a server



3. Create servers of each type from the template

- cisCatalog
- cisContainer
- cisInbound
- cisOutbound



4. Customize the servers

- Configure the bootstrap properties
- Configure the Secure Sockets Layer (SSL)
- Configure the user registry and roles

For container servers:

- Configure the grid
- Configure the database persistence (for container servers)



5. Start the production servers

© Copyright IBM Corporation 2015

Figure 11-30. Steps for creating and configuring a production server topology

WB393 / ZB3931.0

Notes:

Server templates

- Create servers with Decision Server Insights server templates

Server templates	Description
Catalog cisCatalog	<ul style="list-style-type: none"> • Required for high availability • A production topology contains a minimum of three catalog servers • Catalog servers and container servers cannot coexist on the same host
Container cisContainer	<ul style="list-style-type: none"> • Stores entities and system information, and runs agents and analytics jobs • A production topology contains a minimum of three container servers • Container servers and catalog servers cannot coexist on the same host
Inbound cisInbound	<ul style="list-style-type: none"> • Submits inbound events to the grid either through the gateway API, through HTTP, or through JMS • A production topology contains at least one inbound server
Outbound cisOutbound	<ul style="list-style-type: none"> • Emits outbound events through HTTP or through JMS • A production topology contains at least one outbound server

© Copyright IBM Corporation 2015

Figure 11-31. Server templates

WB393 / ZB3931.0

Notes:

Decision Server Insights provides templates for you to generate the various server types that you use to set up your topology.

After you create a server from the template, you customize it by configuring the bootstrap properties, security, and JVM options.



Customizing servers

- For each server that you create, you edit these files:
 - bootstrap.properties
 - server.xml
- For catalog servers, edit bootstrap.properties to configure ia.catalogClusterEndpoints
 - Declares the host and server names, and peer connection ports of all catalog servers in the topology
- For container servers, inbound servers, and outbound servers, edit bootstrap.properties to configure ia.bootstrapEndpoints
 - Declares the host names and client listener ports of the catalog servers in your topology

© Copyright IBM Corporation 2015

Figure 11-32. Customizing servers

WB393 / ZB3931.0

Notes:

For catalog servers, edit bootstrap.properties to configure ia.catalogClusterEndpoints. This property declares the host and server names, and peer connection ports of all catalog servers in the topology.

Set this property by using the following format:

```
computer_name-server_name:computer_name:port_number:port_number,  
computer_name-server_name:computer_name:port_number:port_number
```

For container servers, inbound servers, and outbound servers, edit bootstrap.properties to configure ia.bootstrapEndpoints. This property declares the host names and client listener ports of the catalog servers in your topology.

Set this property by using the following format:

```
computer_name:port_number,computer_name:port_number
```



Configuring security (1 of 2)

- To configure SSL, security and roles, you edit the `server.xml` file for each of the servers
- Generate a key store (`key.jks`) by using the `securityUtility`
 - It is not necessary for all of the servers to use the same key store
 - The key stores contain the certificates that are required for establishing trust
 - Example:
`securityUtility createSSLCertificate --server=cisCatalog1 --password=inslights`
- You must also provide a user registry configuration that defines which users are authorized to access the server
 - The user registry can also authorize the administrator to use the JMX and REST APIs, if necessary

© Copyright IBM Corporation 2015

Figure 11-33. Configuring security (1 of 2)

WB393 / ZB3931.0

Notes:

Configuring security (2 of 2)

- Provide a basic or LDAP user registry configuration

- Defines which users are authorized to access the server
 - Example:

```
<basicRegistry id="basic" realm="SimpleRealm">
    <user name="SimpleAdmin" password="abcdefg"/>
    <group name="SimpleAdministratorsGroup">
        <member name="SimpleAdmin"/>
    </group>
</basicRegistry>
```

- Configure authorization roles for server administration

- Example:

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

© Copyright IBM Corporation 2015

Figure 11-34. Configuring security (2 of 2)

WB393 / ZB3931.0

Notes:



Configuring heap size for container servers

- Customize the heap size for container servers in the `jvm.options` file by setting the `-Xms` property
 - Default: `-Xms28g`
- Configure the grid in the `objectGridDeployment.xml` file, including these properties:
 - `numberOfPartitions`
 - `numInitialContainers`
 - `maxSyncReplicas`
 - `maxAsyncReplicas`

© Copyright IBM Corporation 2015

Figure 11-35. Configuring heap size for container servers

WB393 / ZB3931.0

Notes:

Starting the servers

- Start all servers from the `runtime/wlp/bin/server` directory
 - `server start <serverName>`
- Startup sequence:
 - Catalogs
 - Containers
 - Outbound connectivity
 - Inbound connectivity
- When using multiple catalog servers, at least two must be started concurrently

© Copyright IBM Corporation 2015

Figure 11-36. Starting the servers

WB393 / ZB3931.0

Notes:

To manage the servers, you use the `server` and the `serverManager` utilities.

- Use `server` to start your three catalog servers; all catalogs should be started concurrently
- Use `serverManager` to suspend balancing
- SSH into your four containers and start them by using the `server` command.
- Use `serverManager` to resume balancing
- Use `server` to start your outbound server
- Use `server` to start your inbound server



Unit summary

Having completed this unit, you should be able to:

- Describe WebSphere eXtreme Scale basics
- Describe the Decision Server Insights reference topology
- Design and configure a production topology

© Copyright IBM Corporation 2015

Figure 11-37. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. **True or False:** A Decision Server Insights reference topology requires a minimum of 9 servers.
2. **True or False:** To reduce hardware requirements, you can run inbound and outbound servers on the same physical machine.
3. **True or False:** To compute the number of partitions for containers, you should consider the name of container hosts * the number of cores per host.

© Copyright IBM Corporation 2015

Figure 11-38. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. True
2. True
3. **False.** *The number of partitions is calculated by the name of container hosts * the number of cores per host * 2. The result should be rounded up to the next prime number.*

© Copyright IBM Corporation 2015

Figure 11-39. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 13



Configuring 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.

10.1

Figure 11-40. Exercise 13

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

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

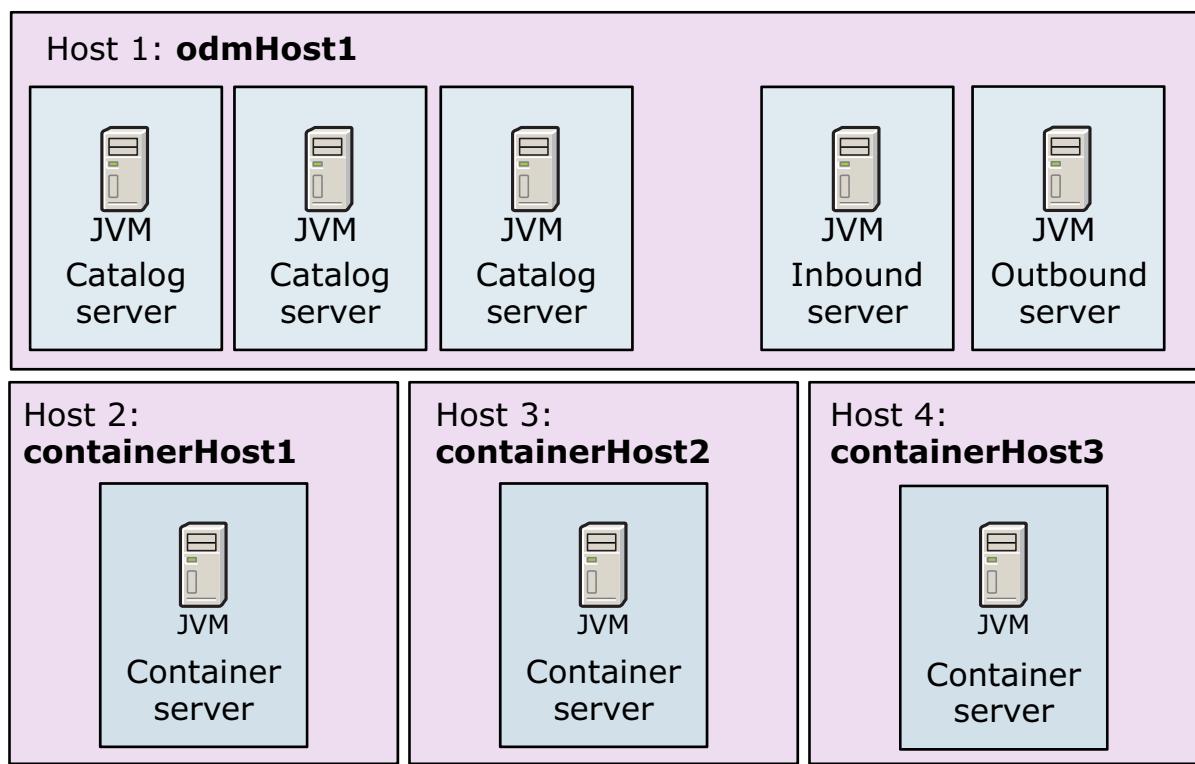
© Copyright IBM Corporation 2015

Figure 11-41. Exercise objectives

WB393 / ZB3931.0

Notes:

Course topology: 4 VMware images



© Copyright IBM Corporation 2015

Figure 11-42. Course topology: 4 VMware images

WB393 / ZB3931.0

Notes:

For this lab, you use four VMware images.

- **odmHost1:** Hosts the complete ODM Advanced installation, including Decision Server Insights. On this host, you create the catalog, inbound, and outbound servers. Each catalog server is configured to be aware of the other catalogs and the other server types in the grid.

For this course, because these servers are all on the same host, you must modify the ports so the catalogs, inbound, and outbound servers are not all listening to the same ports.

- **containerHost1, containerHost2, and containerHost3:** Each of these hosts is used as a container server.



Questions

Configuring the grid: How many partitions would this topology use, with each container running on a single core machine?

$C * M * 2 = <\text{next prime number}>$

Answer: 3 containers * 3 cores * 2 = 18, which must be rounded up to the next prime number: 19.



Stop

The default host names are: odmHost1, containerHost1, containerHost2, and containerHost3.

Your hosts might be assigned different unique host names. Make sure that you know and use the actual host names for your environment during the exercises.

Unit 12. Managing deployment for Decision Server Insights

What this unit is about

This unit explains how to deploy to Insight Server.

What you should be able to do

After completing this unit, you should be able to:

- Manage solution deployment to multiple hosts
- Manage connectivity configuration and deployment

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Manage solution deployment to multiple hosts
- Manage connectivity configuration and deployment

© Copyright IBM Corporation 2015

Figure 12-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Managing deployment
- Managing connectivity

© Copyright IBM Corporation 2015

Figure 12-2. Topics

WB393 / ZB3931.0

Notes:

12.1.Managing deployment

Managing deployment



© Copyright IBM Corporation 2015

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

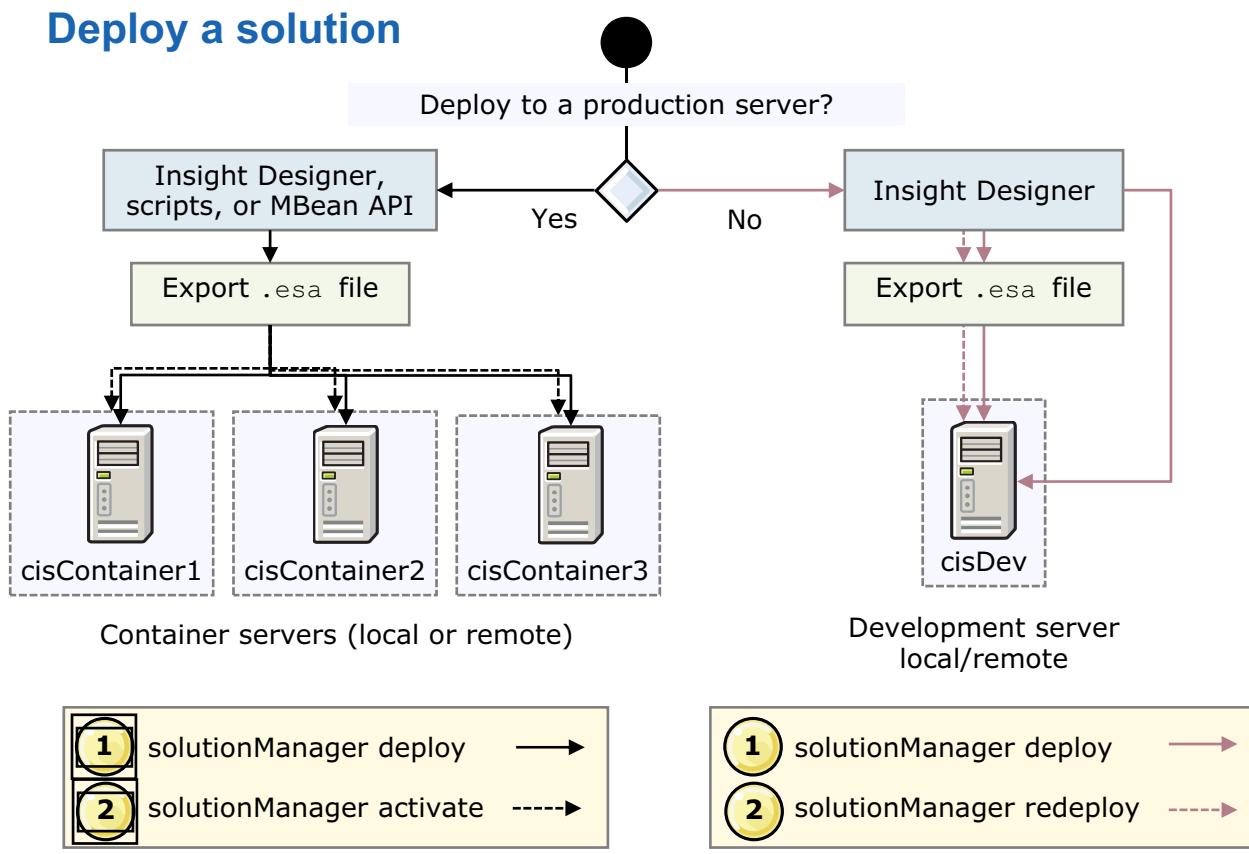
10.1

Figure 12-3. Managing deployment

WB393 / ZB3931.0

Notes:

Deploy a solution



© Copyright IBM Corporation 2015

Figure 12-4. Deploy a solution

WB393 / ZB3931.0

Notes:

Deployment of a solution involves first, exporting the solution to an archive (.esa), for example: mySolution-1.0.esa

Next, you deploy the archive to the server.

Decision Server Insights provides several ways to deploy solutions to both local and remote servers.

In the development environment, you can use Insight Designer to deploy to the development server, cisDev, either locally or remotely. If you have a test or development environment that is set up on multiple hosts to match your production environment, you can test both local and remote deployment.

To deploy to multiple container servers on remote hosts, you can use scripts to automate deployment. The deployment scripts can automate the export of the solution to an archive, deployment of the archive to the servers, and activation of the solution on the servers.



Local deployment

- Using the local parameter
 - Run the script with the Liberty server started or stopped
 - If the server is stopped when you deploy an archive, the solution or agent is accounted for when the server starts
- Example

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

© Copyright IBM Corporation 2015

Figure 12-5. Local deployment

WB393 / ZB3931.0

Notes:

Remote deployment

- Using the `remote` parameter
 - The script establishes an HTTP connection to a running Liberty server on the remote host
 - You must provide administrator authentication credentials to run the script

- Example

```
solutionManager deploy remote C:\solution.esa --  
host=cisContainer1 --port=9080 --username=user1  
--password=insights  
--trustStoreLocation=C:\IBM\ODMInsights871\runtime\wlp\usr\  
servers\cisCatalog1\resources\security\key.jks  
--trustStorePassword=insights
```

© Copyright IBM Corporation 2015

Figure 12-6. Remote deployment

WB393 / ZB3931.0

Notes:



Using connection properties files

- Create a unique `connection.properties` file for each remote server

```
solutionManager deploy remote C:\mySolution-0.0.esa  
--propertiesFile=../etc/connectionC01.properties
```

- Example connection properties for a container

- `server=cisContainer1`
- `host=containerHost1`
- `port=9443`
- `username=admin`
- `password=insights`
- `trustStoreLocation=${wlp.user.dir}/servers/cisCatalog1/
resources/security/key.jks`
- `trustStorePassword=insights`
- `sslProtocol=TLS`
- `disableSSLHostnameVerification=true`

© Copyright IBM Corporation 2015

Figure 12-7. Using connection properties files

WB393 / ZB3931.0

Notes:

To simplify passing the remote host parameters to the remote `deploy` command, you can pass a properties file through the `propertiesFile` parameter.

```
--propertiesFile=InstallDir/runtime/ia/etc/myconnection.properties
```

You can define unique `connection.properties` files for each server that you are using.

Deploying to multiple servers

- Multiple servers that are hosted on the same computer
 1. Deploy a solution to one container server
 2. Deploy the same solution to the other container servers by using the `activateOnly=true` parameter with the `solutionManager deploy` command
- Multiple servers on remote hosts
 - Set the `solutionAutoStart` property in the `server.xml` files to false
 - The auto-start property places a new version of a deployed solution on hold so that you can deploy the new solution version on all the container servers in the topology before you make it active

© Copyright IBM Corporation 2015

Figure 12-8. Deploying to multiple servers

WB393 / ZB3931.0

Notes:

In a topology where multiple servers are hosted on the same computer, you deploy a solution to one container server. You then deploy the same solution to the other container servers by using the `activateOnly=true` parameter with the `solutionManager deploy` command. This parameter ensures that the `server.xml` file is updated, without attempting to redeploy the solution archives. Another use of the `activateOnly` parameter is when you undeploy a solution version and you then want to deploy it again. The `undeploy` command does not delete the archives from the host.

In a production topology with multiple servers, the `solutionAutoStart` property in the `server.xml` files is set to false. This auto-start property places a new version of a deployed solution on hold. You can then deploy the new solution version on all the container servers in the topology before you make it active by running the `activate` command.



Deploying a solution across a grid

1. For all catalogs, containers, and inbound/outbound servers:
 - a. Copy `key.jks` file to all servers
 - b. Make sure the `bootstrap.properties` file for all servers use the same keystore encrypted password
2. On the main host, create a `connection.properties` file for each local and remote server in the grid
 - The connection files are stored in the `ia\etc` folder
3. Deploy the `.esa` file to containers only
4. Deploy connectivity configuration XML files to the inbound and outbound servers

© Copyright IBM Corporation 2015

Figure 12-9. Deploying a solution across a grid

WB393 / ZB3931.0

Notes:

Security must be the same for all servers in the grid.



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

© Copyright IBM Corporation 2015

Figure 12-10. Deployed solutions files

WB393 / ZB3931.0

Notes:



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 12-11. Undeploying (1 of 2)

WB393 / ZB3931.0

Notes:



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
- Increment the solution version number before redeploying

© Copyright IBM Corporation 2015

Figure 12-12. Undeploying (2 of 2)

WB393 / ZB3931.0

Notes:

After you undeploy a solution, you must increment the solution version number before you redeploy. If you try to deploy the solution again without incrementing the version, the script detects the solution feature in the manifest files and displays an error.



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
- Example: To delete a 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 12-13. Deleting solution files

WB393 / ZB3931.0

Notes:

12.2.Managing connectivity

Managing connectivity



© Copyright IBM Corporation 2015

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

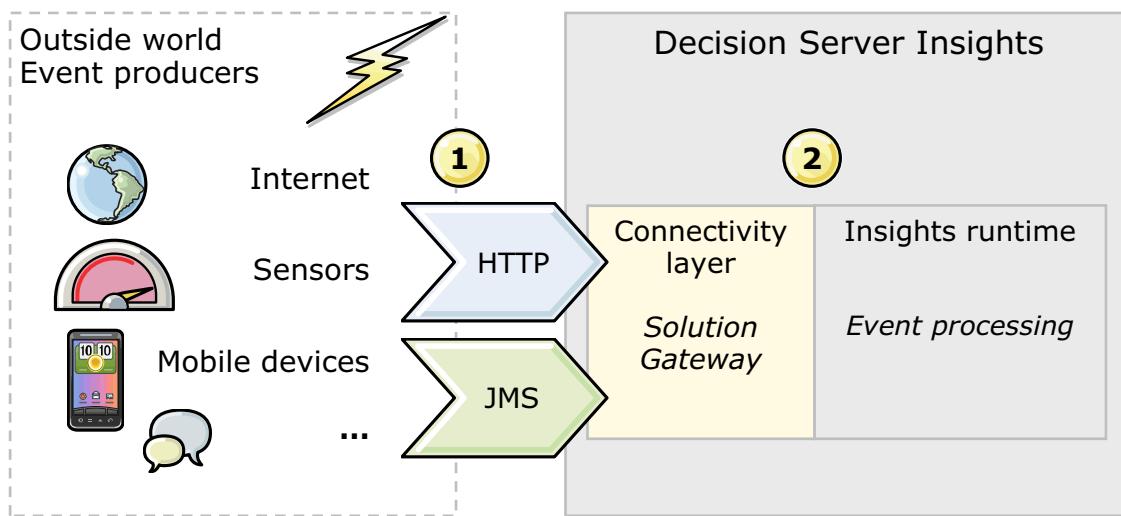
10.1

Figure 12-14. Managing connectivity

WB393 / ZB3931.0

Notes:

Connectivity integration



1. Event producers in the outside world pass HTTP or JMS messages to Decision Server Insights
2. Inbound messages are received by the Solution Gateway and routed to the runtime

© Copyright IBM Corporation 2015

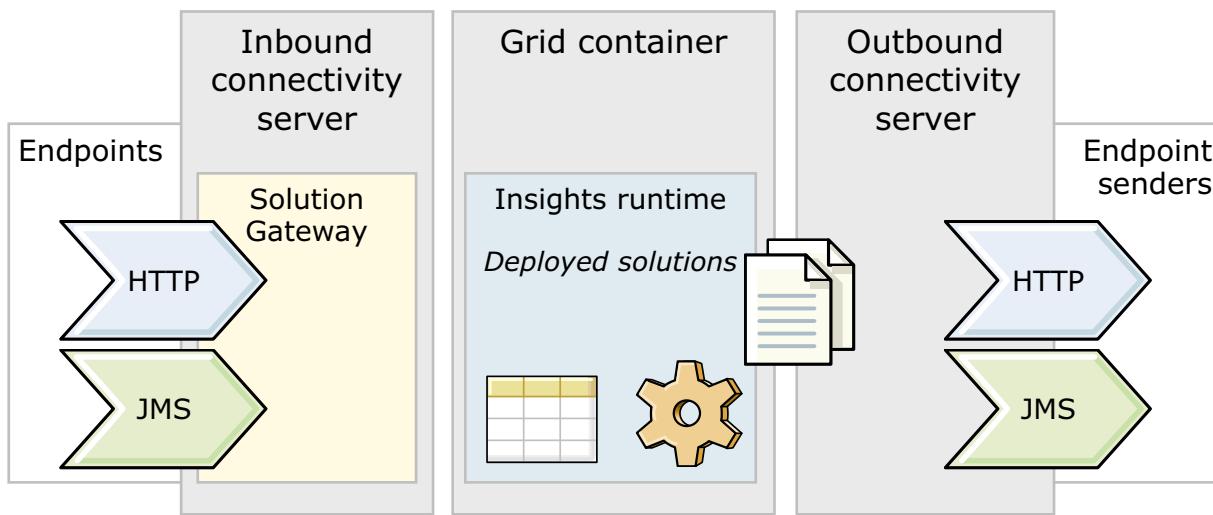
Figure 12-15. Connectivity integration

WB393 / ZB3931.0

Notes:

Decision Server Insights interacts with the outside world through the connectivity layer.

Connectivity architecture



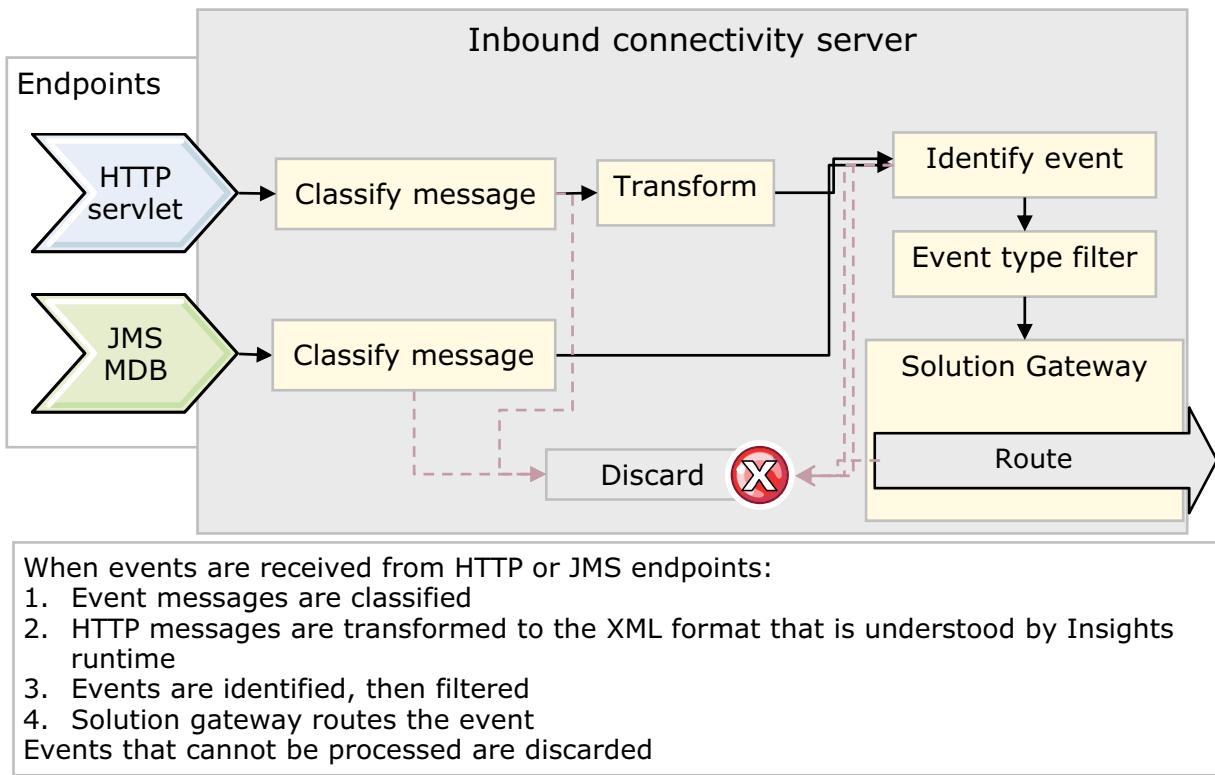
© Copyright IBM Corporation 2015

Figure 12-16. Connectivity architecture

WB393 / ZB3931.0

Notes:

Connectivity architecture: Inbound



© Copyright IBM Corporation 2015

Figure 12-17. Connectivity architecture: Inbound

WB393 / ZB3931.0

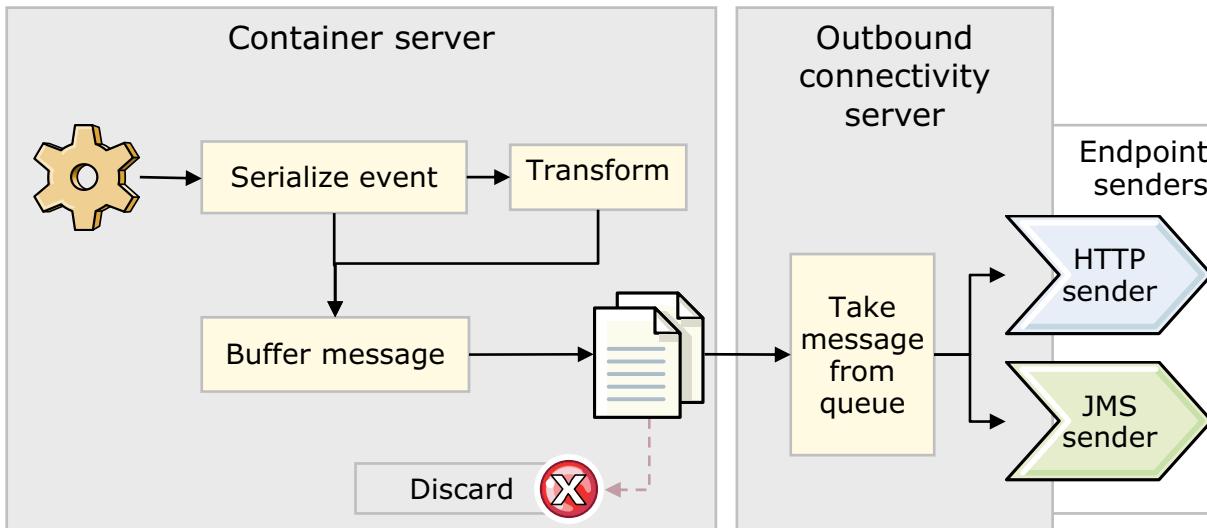
Notes:

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.

Connectivity architecture: Outbound



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

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

© Copyright IBM Corporation 2015

Figure 12-18. Connectivity architecture: Outbound

WB393 / ZB3931.0

Notes:

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



Connectivity definitions

- Connectivity definitions are contained in one or more .cdef files in the solution project
 - Bindings define the events and their representation
 - Endpoints represent a logical JMS queue or topic or HTTP URL

Example

Inbound: "Transaction" events from HTTP client

Outbound:
"Authorization Response" events to a JMS queue

```

inboundConnectivity.cdef [x] outboundConnectivity.cdef
// Connectivity definitions for the solution
define inbound binding 'HTTPTransactionEventBinding'
  with description "HTTP Incoming Transaction",
  using
    message format application/xml ,
    protocol HTTP,
    accepting events :
      - Transaction .

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

inboundConnectivity.cdef [x] outboundConnectivity.cdef [x]
define outbound binding 'JMSAuthorizationResponseOutputBinding'
  with description "JMS Output authorization response",
  using message format application/xml ,
  protocol JMS ,
  delivering events :
    - Authorization Response .

define outbound JMS endpoint 'JMSAuthorizationResponseEndPoint'
  with description "JMS Output authorization response end point",
  using binding 'JMSAuthorizationResponseOutputBinding',
  connection factory "InsightsCF",
  destination "AuthTopic".

```

© Copyright IBM Corporation 2015

Figure 12-19. Connectivity definitions

WB393 / ZB3931.0

Notes:

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

In the connectivity definition file, you define:

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

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

These definitions include the following information:

- Binding name

- Message format, either `application/xml` or `text/xml`
- Protocol, either JMS or HTTP
- Inbound messages to be received over this binding
- Endpoint name
- Endpoint binding name
- URL path (for HTTP only)

For HTTP inbound endpoints, 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`.



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

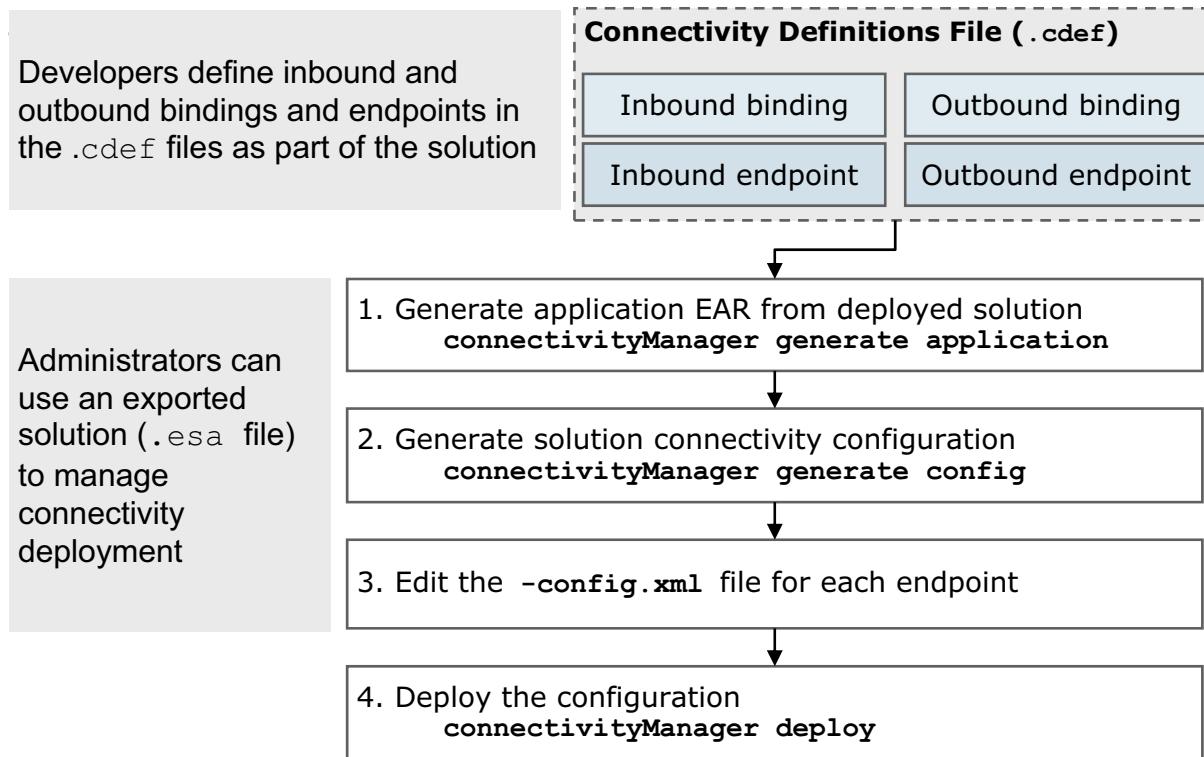
Figure 12-20. Providers for JMS

WB393 / ZB3931.0

Notes:

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

Connectivity deployment steps



© Copyright IBM Corporation 2015

Figure 12-21. Connectivity deployment steps

WB393 / ZB3931.0

Notes:

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

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



Configuration and deployment

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

© Copyright IBM Corporation 2015

Figure 12-22. Configuration and deployment

WB393 / ZB3931.0

Notes:

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

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

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

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



Creating a solution connectivity configuration

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

- Use the `validate` command to validate your endpoint configuration

- Example:
`connectivityManager validate C:\lab\banking_solution-1.1.esa
C:\lab\banking-solution-inbound-config.xml`

© Copyright IBM Corporation 2015

Figure 12-23. Creating a solution connectivity configuration

WB393 / ZB3931.0

Notes:

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

Editing the connectivity configuration

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

```

<?xml version="1.0" encoding="utf-8"?><server>
  <!--Application definition for inbound connectivity application for solution:
  ConnectivitySolution-->
  <application location="ConnectivitySolution-inbound.ear">
    <application-bnd>
      <security-role name="iaEventSubmitter"/>
    </application-bnd>
  </application>
  ...
  <ia_inboundHttpEndpoint endpoint=
    "ConnectivitySolution/HTTPTransactionEventEndPoint" />
</server>

```

© Copyright IBM Corporation 2015

Figure 12-24. Editing the connectivity configuration

WB393 / ZB3931.0

Notes:

The configuration points to the application EAR file in the `<application>` entry, which you uncomment. At the end of the file, the inbound endpoint is defined for an HTTP endpoint.



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 12-25. Deploying a connectivity configuration

WB393 / ZB3931.0

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 files
 - `Server.xml` file, and any connectivity configuration files that `server.xml` includes
 - The `.cdef` files from the solution
 - The solution
- 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 12-26. Troubleshooting

WB393 / ZB3931.0

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 to the Insight runtime on the inbound channel creates events to send through outbound channels
- To check that the server processes events correctly, use a test client
 - Using a test client removes the channels from the path and determines whether events are being handled correctly
 - For information on using a test client, see the product documentation

© Copyright IBM Corporation 2015

Figure 12-27. Common issues and resources

WB393 / ZB3931.0

Notes:



Unit summary

Having completed this unit, you should be able to:

- Manage solution deployment to multiple hosts
- Manage connectivity configuration and deployment

© Copyright IBM Corporation 2015

Figure 12-28. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

- 1. True or False:** You use the `solutionManager deploy` command for local deployment only.

- 2. True or False:** You can use the `connectivityManager` utility to generate and deploy connectivity configuration files for your solution.

© Copyright IBM Corporation 2015

Figure 12-29. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.

- 2.



Checkpoint answers

1. **False.** You use the *solutionManager deploy* command for both local and remote deployment.
2. **True.**

© Copyright IBM Corporation 2015

Figure 12-30. Checkpoint answers

WB393 / ZB3931.0

Notes:

Exercise 14



Managing deployment and connectivity

© Copyright IBM Corporation 2015

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

10.1

Figure 12-31. Exercise 14

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Deploy solutions across the grid
- Configure inbound and outbound endpoints
- Deploy and activate inbound and outbound connectivity configurations

© Copyright IBM Corporation 2015

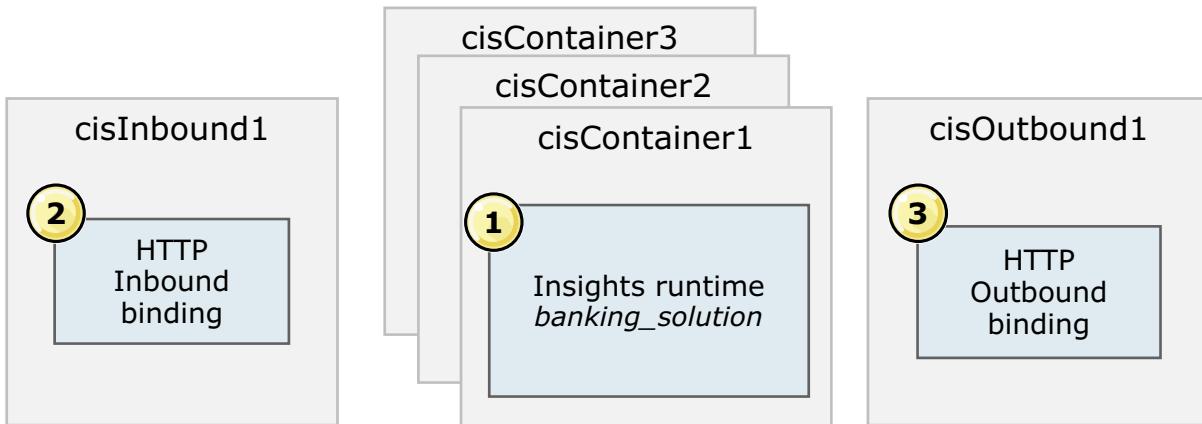
Figure 12-32. Exercise objectives

WB393 / ZB3931.0

Notes:



Exercise overview: Deploying a solution and connectivity



1. Deploy solution to grid containers
2. Generate and deploy inbound configuration to inbound server
3. Generate and deploy outbound configuration to outbound server

© Copyright IBM Corporation 2015

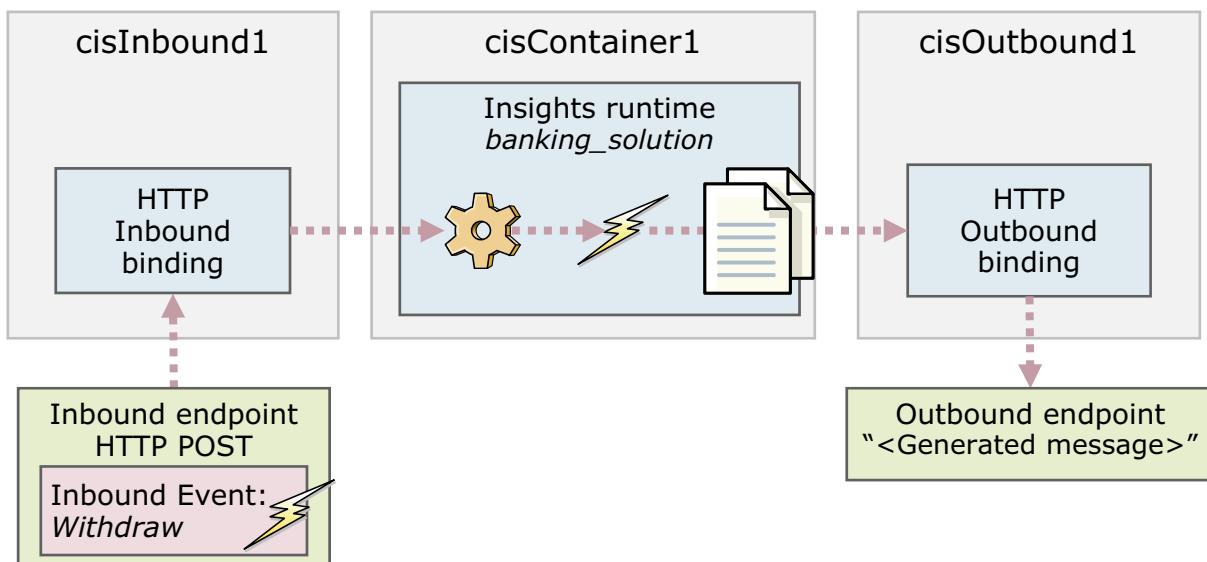
Figure 12-33. Exercise overview: Deploying a solution and connectivity

WB393 / ZB3931.0

Notes:

During the exercise, you deploy a solution to the grid containers. You deploy inbound connectivity to the inbound server, and outbound connectivity to the outbound server.

Exercise overview: Testing connectivity



© Copyright IBM Corporation 2015

Figure 12-34. Exercise overview: Testing connectivity

WB393 / ZB3931.0

Notes:

To test the deployment, you use an HTTP tool to submit an event through the inbound server. The result of successful processing of the event is that an outbound message is generated by the runtime and the outbound connectivity server sends that message to the outbound endpoint.

Unit 13. Administering Decision Server Insights

What this unit is about

This unit explains how to administer Decision Server Insights.

What you should be able to do

After completing this unit, you should be able to:

- Use administration scripts to monitor status and activity of your servers and grid
- Use trace files and logging to monitor Decision Server Insights
- Describe how to monitor WebSphere eXtreme Scale and WebSphere MQ

How you will check your progress

- Checkpoint
- Exercise



Unit objectives

After completing this unit, you should be able to:

- Use administration scripts to monitor status and activity of your servers and grid
- Use trace files and logging to monitor Decision Server Insights
- Describe how to monitor WebSphere eXtreme Scale and WebSphere MQ

© Copyright IBM Corporation 2015

Figure 13-1. Unit objectives

WB393 / ZB3931.0

Notes:



Topics

- Administration tools
- Logging
- Monitoring WebSphere eXtreme Scale
- Monitoring WebSphere MQ

© Copyright IBM Corporation 2015

Figure 13-2. Topics

WB393 / ZB3931.0

Notes:

13.1. Administration tools

Administration tools



© Copyright IBM Corporation 2015

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

10.1

Figure 13-3. Administration tools

WB393 / ZB3931.0

Notes:



Liberty profile monitoring

- JVM monitoring
 - Use the JvmStats MXBean for JVM monitoring in the Liberty profile
- ThreadPool monitoring
 - Use a ThreadPool MXBean
- Multiple components monitoring
 - You can filter the components that you want to monitor by using the monitor-1.0 feature

© Copyright IBM Corporation 2015

Figure 13-4. Liberty profile monitoring

WB393 / ZB3931.0

Notes:

Some of the important monitor components that you want to monitor as part of Decision Server Insights and WebSphere eXtreme Scale are JVM and ThreadPool. ThreadPool is critical when you have a large number of partitions on a server or on a single host server. If you have hundreds of partitions per server, thousands of threads are allocated, which can quickly lead to resource contention.



Server administration scripts

- Decision Server Insights administration scripts to:
 - Deploy and manage solutions
 - Deploy and manage connectivity
 - Manage server properties and activity

Property	Description
propertyManager	Use the propertyManager script to get, set, and manage server properties
connectivityManager	Use the connectivityManager script to generate, deploy, and activate connectivity for your solution
serverManager	Use the serverManager script to pause or shutdown server processes, and check the <code>isonline</code> status for a server
solutionManager	Use solutionManager to deploy and manage solutions across your servers

© Copyright IBM Corporation 2015

Figure 13-5. Server administration scripts

WB393 / ZB3931.0

Notes:

You worked with the `solutionManager` and `connectivityManager` scripts during the deployment exercises.

For more information about using these scripts, see the “Reference” section of the product documentation.



Managing servers

- Use `serverManager isonline` command to determine whether a server is running
- Run from the `InstallDir\runtime\ia\bin` directory

```
serverManager isonline [--propertiesFile=properties_file]
[--username=username] [--password=password]
[--host=hostname] [--port=port]
[--keyStoreLocation=keystore_location]
[--keyStorePassword=keystore_password]
[--sslProtocol=sslProtocol]
[--trustStorePassword=truststore_password]
[--trustStoreLocation=truststore_location]
[--disableSSLHostnameVerification=true|false]
```

- Example

```
serverManager isonline --propertiesFile=../etc/
connectionC1.properties
```

© Copyright IBM Corporation 2015

Figure 13-6. Managing servers

WB393 / ZB3931.0

Notes:

The `serverManager` script includes the `isonline` command, which checks for running and active local or remote servers. Use the command to identify a local or remote server where you can deploy or activate solutions.

You must provide administrator authentication credentials to run the script. Specify the server host name and port to determine the status of a remote server. You can pass these credentials and server details through a unique `connection.properties` file with the `propertiesFile` parameter.

Managing server properties

- Use `propertyManager` to set and retrieve server properties and log settings
- Run from the `InstallDir\runtime\ia\bin` directory

```
propertyManager list [--propertiesFile=properties_file]
[--username=username] [--password=password]
[--host=hostname] [--port=port]
[--keyStoreLocation=keystore_location]
[--keyStorePassword=keystore_password]
[--sslProtocol=sslProtocol]
[--trustStorePassword=truststore_password]
[--trustStoreLocation=truststore_location]
[--disableSSLHostnameVerification=true|false]
```

- Example

```
propertyManager list --propertiesFile=../etc/
connectionC1.properties
```

© Copyright IBM Corporation 2015

Figure 13-7. Managing server properties

WB393 / ZB3931.0

Notes:

You use the `propertyManager` script to manage log settings and server properties. This script includes the `list` command to list which properties can be set on your server.

To view the list of properties that you can set with the `propertyManager` script, run the `list` command.

As with the `serverManager` script, you can pass credentials and server details through a unique `connection.properties` file with the `propertiesFile` parameter.

13.2.Logging

Logging



© Copyright IBM Corporation 2015

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

10.1

Figure 13-8. Logging

WB393 / ZB3931.0

Notes:

Log and trace files

- Use these files to monitor the activity and status of your servers

File	When to use	Supported tasks
<code>console.log</code>	<p>Use to view standard process output and error messages from the runtime environment, for run and debug actions.</p> <p>Messages are redirected to the <code>server_name/logs/console.log</code> file when you start the server.</p>	<p>View a message that confirms that a solution is deployed and ready to use, for example: CWMBD0060I: Solution ConnectivitySolution-0.1 ready</p> <p>View other standard process output and error messages.</p>
<code>messages.log</code>	Use to view operational messages and information that is generated by the system.	<p>View INFO, AUDIT, WARNING, ERROR, and FAILURE messages.</p> <p>View time stamps and the IDs of issuing threads.</p>
<code>trace.log</code>	Use to gather and view debug information.	Gather and view debug information that is obtained by basic, enhanced, or advanced tracing.

© Copyright IBM Corporation 2015

Figure 13-9. Log and trace files

WB393 / ZB3931.0

Notes:

For more information, see "Trace and logging" in the WebSphere Application Server, Liberty core documentation.

Server log settings

Attribute	Description
maxFileSize	If an enforced maximum file size exists, this setting is used to determine how many of each of the log files are kept. This setting also applies to the number of exception logs that summarize exceptions that occurred on any particular day. The default value is 2.
	Note: maxFiles does not apply to the <code>console.log</code> file.
consoleLogLevel	This filter controls the granularity of messages that go to the <code>console.log</code> file. The valid values are INFO, AUDIT, WARNING, ERROR, and OFF. The default level is AUDIT.
traceSpecification	This attribute controls the format of the trace log. The default format for the Liberty profile is ENHANCED. You can also use BASIC and ADVANCED formats as in the full profile.

© Copyright IBM Corporation 2015

Figure 13-10. Server log settings

WB393 / ZB3931.0

Notes:

The logging component can be controlled through the server configuration. The primary location for the logging configuration is in the `server.xml` file. Occasionally, you might need to configure trace to diagnose a problem that occurs before the `server.xml` file is processed. In this case, the equivalent configuration properties can be specified in the `bootstrap.properties` file. If a configuration property is specified in both the `bootstrap.properties` file and the `server.xml` file, the value in `bootstrap.properties` is used until the `server.xml` file is processed. Then, the value in the `server.xml` file is used. Avoid specifying different values for the same configuration property in both the `bootstrap.properties` and the `server.xml` file.

You can set logging properties in the server configuration file by selecting **Logging and Tracing** in the Server Configuration view in the developer tools. You can also add a logging element to the server configuration file as follows:

```
<component> = <level>
```

In this case, `<component>` is the component for which to set a log detail level, and `<level>` is one of the valid log levels (off, fatal, severe, warning, audit, info, config, detail, fine, finer, finest, all). Separate multiple log detail level specifications with colons (:).

For more information, see "Trace and logging" in the WebSphere Application Server, Liberty core documentation.

Server log settings

- Default log levels are set for each server type
- The following `server.xml` elements define the default log levels for each server type

Server type	Log level
cisDev (default development server)	<code><logging traceSpecification="*=info" maxFiles="5" /></code>
cisCatalog	<code><logging traceSpecification="*=info" maxFiles="5" /></code>
cisContainer , cisInbound , cisOutbound	<code><logging traceSpecification="com.ibm.ia.*=info:com.ibm.rules.*=info: *=info" maxFiles="5" /></code>

- You can set the logging level to one of the following values:
`severe, warning, audit, info, fine, finer, finest, off`

© Copyright IBM Corporation 2015

Figure 13-11. Server log settings

WB393 / ZB3931.0

Notes:

You change the level for a log by editing the `server.xml` files, and in some cases the `bootstrap.properties` file.

See

http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.itoa.admin/topics/con_logging_settings.html

Changing log and trace settings

- Edit the trace and logging properties in the `server.xml` file on each of the containers, inbound and outbound servers, and catalogs
 - **Note:** Make a backup of the `server.xml` file before you edit

- Create a trace file for a solution with `messageFileName` parameter to the `<logging>` entry
 - Example:


```
<logging
traceSpecification="com.ibm.rules.generated.dataie.banking_
scenari.* =detail:com.ibm.ia.*=warning:
com.ibm.ia.runtime.SolutionProviderMgr=finest:com.ibm.rules.
*=info:*=warning" maxFiles="10"
messageFileName="bankingSolutionMessages.log" />
```

© Copyright IBM Corporation 2015

Figure 13-12. Changing log and trace settings

WB393 / ZB3931.0

Notes:

To change log settings for your grid, you edit the `server.xml` files for each of the containers, the inbound and outbound servers, and the catalogs. You can make these changes while the servers are running. You do not need to restart the servers because the changes are detected and applied automatically.

To generate more extensive traces of a solution, you can add or modify logging entries in the `server.xml` file. For example, you can create a trace file to capture logs for a particular solution by adding the `messageFileName` parameter. You can also increase the number of files that are generated for each log by setting the `maxFiles` parameter. If you increase this number to 10, for example, you might have 10 message logs, 10 trace logs, and 10 exception summaries in the `ffdc` directory. The trace file is created only if you enable additional trace.

Finding troubleshooting information in log and trace files

- Analyze the `trace.log` and `messages.log` files
- Search for key phrases

Key phrase	Information
<code>received</code>	ID, timestamp, and other attributes of an incoming event
<code>event_name</code>	
<code>begin processing</code>	Agent processing activity
<code>retrieved entity</code>	Class name, ID, and other attributes of a bound entity
<code>rule rule_name</code>	Rule processing and output
<code>execution started</code>	

© Copyright IBM Corporation 2015

Figure 13-13. Finding troubleshooting information in log and trace files

WB393 / ZB3931.0

Notes:

See:

http://www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/com.ibm.odm.admin/topics/con_find_trb_log_files.html

Server administration properties

- Server properties include the following properties

Property	Description	Default
debugPort	Indicates one or more ports that the server monitors for connections from a test driver instance. The value is a single port, or a range or ports, which matches the <code>httpport</code> property as specified in the <code>testdriver.properties</code> file.	none
LogSuppressionThreshold	Defines the number of occurrences of any log message that is allowed in a specified time period. The value of this property determines the number of times a log message is produced before it is suppressed.	10
LogSuppressionThresholdPeriod	Defines the time period during which log messages are counted. A log message rate is calculated from the value of the log threshold and the time period for the threshold. When the message rate is deemed too high, all subsequent messages are suppressed. Messages are produced again only when the rate falls to an acceptable volume.	20000

© Copyright IBM Corporation 2015

Figure 13-14. Server administration properties

WB393 / ZB3931.0

Notes:

The `propertyManager set` command sets properties that modify the behavior and characteristics of the server. The properties apply across to all solutions on the server.

In the development environment, the `<ia runtime>` entry might include the `debugPort` property and `solutionAutoStart="true"`.

```
<ia_runtime debugPort="6543" solutionAutoStart="true" />
```

You can use the `LogSuppressionThreshold` to limit the number of times a log message is produced before it is suppressed. For example, you can reduce the threshold from 10 to 2. This value means that after 2 log messages are generated within the threshold period, subsequent messages are suppressed.

You can also change other values, such as `LogSuppressionThresholdPeriod`, which defines the time period over which log messages are counted. The default is 20000 ms. If you increase this value, the same number of messages are produced, but over a longer period. As a result, you receive fewer messages.

13.3. Monitoring WebSphere eXtreme Scale

Monitoring WebSphere eXtreme Scale



© Copyright IBM Corporation 2015

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

10.1

Figure 13-15. Monitoring WebSphere eXtreme Scale

WB393 / ZB3931.0

Notes:



WebSphere eXtreme Scale: xscmd

- The xscmd utility displays textual information about the grid topology
 - Make sure that catalog servers and container servers are started before using this tool
 - If your catalog servers are set up with majority quorum enabled, at least two catalog servers must be started

© Copyright IBM Corporation 2015

Figure 13-16. WebSphere eXtreme Scale: xscmd

WB393 / ZB3931.0

Notes:

With `xscmd`, you can display textual information about your Decision Server Insights WebSphere eXtreme Scale grid topology.

There are a number of tools and techniques to determine what is happening in an eXtreme Scale environment during run time.

As mentioned earlier, you want to monitor the JVM and ThreadPool, especially when you have a large number of partitions that are on a single host server to avoid resource contention.

Monitoring the status of quorum (1 of 2)

- To verify the status of your catalogs, use the eXtreme Scale xscmd showQuorumStatus command

- Example:

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

- This command returns the status of the catalogs that are associated with the catalog service

```
C:\IBM\ODMInsights87\runtime\wlp\bin>xscmd -c showQuorumStatus -cep localhost:2809
Starting at: 2015-04-30 02:48:58.181
CWXSI0068I: Executing command: showQuorumStatus
Server          Host    Quorum   Quorum Size Active Servers
-----        ----    -----   -----  -----
localhost-cisCatalog1 dsiMain DISABLED -
                               localhost-cisCatalog1,
                               localhost-cisCatalog2,
                               localhost-cisCatalog3
localhost-cisCatalog2 dsiMain DISABLED -
                               localhost-cisCatalog1,
                               localhost-cisCatalog2,
                               localhost-cisCatalog3
localhost-cisCatalog3 dsiMain DISABLED -
                               localhost-cisCatalog1,
                               localhost-cisCatalog2,
                               localhost-cisCatalog3
CWXSI0040I: The showQuorumStatus command completed successfully.
Ending at: 2015-04-30 02:48:59.884
```

© Copyright IBM Corporation 2015

Figure 13-17. Monitoring the status of quorum (1 of 2)

WB393 / ZB3931.0

Notes:



Monitoring the status of quorum (2 of 2)

- Quorum includes these statuses:

Status	Descriptions
TRUE	The server has quorum enabled and the system is working normally. Quorum is met.
FALSE	The server has quorum enabled, but quorum is lost. The catalog servers do not allow changes to the catalog service domain.
UNAVAILABLE	The server cannot be contacted. It is either not running or there is a network problem and the server cannot be reached.
DISABLED	The server does not have quorum enabled.

© Copyright IBM Corporation 2015

Figure 13-18. Monitoring the status of quorum (2 of 2)

WB393 / ZB3931.0

Notes:



Monitoring that all catalogs are online

- To verify whether the catalogs are running, use the `xscmd routetable` command
 - Example:
`xscmd -c routetable -cep localhost:2810`
- This command returns the online status of the catalogs that are associated with the catalog service

© Copyright IBM Corporation 2015

Figure 13-19. Monitoring that all catalogs are online

WB393 / ZB3931.0

Notes:

13.4. Monitoring WebSphere MQ

Monitoring WebSphere MQ



© Copyright IBM Corporation 2015

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

10.1

Figure 13-20. Monitoring WebSphere MQ

WB393 / ZB3931.0

Notes:

Tracing the WebSphere MQ resource adapter (1 of 2)

Property	Description	Default
logWriterEnabled	A flag to enable or disable the sending of a diagnostic trace to a LogWriter object provided by the application server. If the value is true, the trace is sent to a LogWriter object instead of the location that is specified by the <code>traceDestination</code> property. If the value is <code>false</code> , any LogWriter object that is provided by the application server is not used.	False
traceEnabled	A flag to enable or disable diagnostic tracing. If the value is <code>false</code> , tracing is turned off. If the value is <code>true</code> , a trace is sent to the location specified by the <code>traceDestination</code> property.	False
traceDestination	<p>The location to where a diagnostic trace is sent</p> <ul style="list-style-type: none"> If the value is <code>System.err</code>, the trace is directed to the system error stream instead of a file If the value is <code>System.out</code>, the trace is directed to the system output stream <p>Example: <code>/tmp/wmq_jca.trace</code></p> <p>Note: If <code>logWriterEnabled</code> is set to <code>True</code>, the log redirects to the LogWriter</p>	

© Copyright IBM Corporation 2015

Figure 13-21. Tracing the WebSphere MQ resource adapter (1 of 2)

WB393 / ZB3931.0

Notes:

With the WebSphere MQ resource adapter, you can configure diagnostic trace as a property on the resource adapter. The resource adapter RAR file contains a file that is called `META-INF/ra.xml`, which contains a deployment descriptor for the resource.

To change the behavior of the log and the settings, you must change the `ra.xml` that is included in the `wmq.jmsra.rar` file.

You extract the files to locate the `ra.xml` file in the **META-INF** folder. After you make the change, you replace the `ra.xml` file back in the `wmq.jmsra.rar` file.

Tracing the WebSphere MQ resource adapter (1 of 2)

- Example

```

<config-property>
  <config-property-name>logWriterEnabled</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>false</config-property-value>
</config-property>
<config-property>
  <config-property-name>traceEnabled</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>true</config-property-value>
</config-property>
<config-property>
  <config-property-name>traceLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>6</config-property-value>
</config-property>
<config-property>
  <config-property-name>traceDestination</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>/tmp/wm_jca.trace</config-property-value>
</config-property>

```

© Copyright IBM Corporation 2015

Figure 13-22. Tracing the WebSphere MQ resource adapter (1 of 2)

WB393 / ZB3931.0

Notes:

In the example that is shown here, the `traceDestination` property is added as the location where the log file should be stored. The `traceLevel` property is changed from 3 (error and warning messages) to 6 (error, warning, and information messages).

For more information, see the IBM Knowledge Center about enabling and changing the trace level for JMS logging and about tracing the WebSphere MQ resource adapter.



Unit summary

Having completed this unit, you should be able to:

- Use administration scripts to monitor status and activity of your servers and grid
- Use trace files and logging to monitor Decision Server Insights
- Describe how to monitor WebSphere eXtreme Scale and WebSphere MQ

© Copyright IBM Corporation 2015

Figure 13-23. Unit summary

WB393 / ZB3931.0

Notes:



Checkpoint questions

1. You run the Decision Server Insights server administration scripts from which directory:
 - A. <InstallDir>\runtime\ia\bin
 - B. <InstallDir>\runtime\wlp\bin

2. **True or False:** After modifying logging properties in the `server.xml` file for your containers and catalogs, you must restart the servers.

3. **True or False:** You use `xscmd` to manage the WebSphere eXtreme Scale grid.

4. You run the `xscmd` script from which directory:
 - A. <InstallDir>\runtime\ia\bin
 - B. <InstallDir>\runtime\wlp\bin

© Copyright IBM Corporation 2015

Figure 13-24. Checkpoint questions

WB393 / ZB3931.0

Notes:

Write your answers here:

- 1.

- 2.

- 3.

- 4.



Checkpoint answers

1. **A.** <InstallDir>\runtime\ia\bin
2. **False.** Changes to the logging properties in the server.xml file are automatically applied without restarting the server.
3. **True.**
4. **B.** <InstallDir>\runtime\wlp\bin

© Copyright IBM Corporation 2015

Figure 13-25. Checkpoint answers

WB393 / ZB3931.0

Notes:



Exercise 15



Figure 13-26. Exercise 15

WB393 / ZB3931.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Monitor and manage the hosts in a Decision Server Insights grid

© Copyright IBM Corporation 2015

Figure 13-27. Exercise objectives

WB393 / ZB3931.0

Notes:



General administration utilities

- WebSphere eXtreme Scale monitoring
 - Use `xscmd` utility to view information about the grid
- Liberty profile server management
 - Use `server` and `serverManager` utilities to manage servers within your grid
 - Run these utilities from the `<InstallDir>/runtime/wlp/bin` directory
- Decision Server Insights solutions and connectivity
 - Use `solutionManager` and `connectivityManger`
 - Run these utilities from the `<InstallDir>/runtime/ia/bin` directory

© Copyright IBM Corporation 2015

Figure 13-28. General administration utilities

WB393 / ZB3931.0

Notes:

Unit 14. 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 14-1. Unit objectives

WB393 / ZB3931.0

Notes:

Course learning objectives

After completing this course, you should be able to:

- Describe the IBM Operational Decision Manager Advanced technical architecture
- Explain topology options for a clustered environment
- Complete postinstallation configuration of Decision Server Rules, Decision Center, and Decision Server Insights
- Manage security for ODM Advanced components
- Synchronize projects across business and development environments
- Manage deployment for Decision Server Rules and Decision Server Insights
- Monitor rule execution and troubleshoot performance issues

© Copyright IBM Corporation 2015

Figure 14-2. Course learning objectives

WB393 / ZB3931.0

Notes:



Course learning objectives

After completing this course, you should be able to:

- Manage versions, baselines, and multiple releases in Decision Center
- Install and configure a Decision Server Insights reference topology
- Configure inbound and outbound servers to manage connectivity

© Copyright IBM Corporation 2015

Figure 14-3. Course learning objectives

WB393 / ZB3931.0

Notes:



To learn more on the subject (1 of 2)

- For more information about Operational Decision Manager and Decision Server Insights:
 - DeveloperWorks tutorial: *Improve performance for IBM Operational Decision Manager, Part 1*
http://www.ibm.com/developerworks/bpm/bpmjournal/1503_siddiqui1/1503_siddiqui1.html
 - DeveloperWorks tutorial: *Improve performance for IBM Operational Decision Manager, Part 2*
http://www.ibm.com/developerworks/bpm/library/techarticles/1505_siddiqui2/1505_siddiqui2.html
 - DeveloperWorks tutorial: *Install and configure a Decision Server Insights reference topology*
http://www.ibm.com/developerworks/bpm/bpmjournal/1503_defreitas1/1503_defreitas1.html
 - Kolban's Book on IBM Decision Server Insights
<http://neilkolban.com/ibm/decision-server-insights>

© Copyright IBM Corporation 2015

Figure 14-4. To learn more on the subject (1 of 2)

WB393 / ZB3931.0

Notes:



To learn more on the subject (2 of 2)

- Product documentation
 - www.ibm.com/support/knowledgecenter/SSQP76_8.7.1/welcome/kc_welcome_odmV.html
- IBM Training website:
www.ibm.com/training
- For more information about technical resources for ODM:
developer.ibm.com/odm
- For more information about decision management:
www.ibm.com/software/decision-management

© Copyright IBM Corporation 2015

Figure 14-5. To learn more on the subject (2 of 2)

WB393 / ZB3931.0

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 14-6. Unit summary

WB393 / ZB3931.0

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
BI	business intelligence
BMD	business model definition
BOM	business object model
BPM	business process management
BPMN	Business Process Modeling Notation
BQL	Business Query Language
BRM	business rule management
BRMS	business rule management system
CEP	complex event processing
CICS	Customer Information Control System
CPU	central processing unit
CRM	customer relationship management
CVS	Concurrent Versions System
DAO	Data Access Object
Db	Database
DSI	Decision Server Insights
DVD	digital versatile disc
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

FIPS	Federal Information Processing Standards
GUI	Graphical user interface
HADR	high availability and disaster recovery
HMM	Hidden Markov Models
HTDS	hosted transparent decision service
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure-as-a-Service
IBM	International Business Machines Corporation
IDE	Integrated development environment
IIB	IBM Information Bus
IP	Internet Protocol
IRL	ILOG Rule Language
IT	information technology
J2C	Java EE Connector Architecture
JAR	Java archive
Java EE	Java Platform, Enterprise Edition
Java SE	Java Platform, Standard Edition
JAXB	Java Architecture for XML Binding
JCA	Java EE Connector Architecture
JDBC	Java Database Connectivity
JDK	Java Development Kit
JMS	Java Message Service
JMX	Java Management Extension
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JVM	Java virtual machine
KPI	key performance indicator
LAN	Local area network
LDAP	Lightweight Directory Access Protocol
MAT	Memory Analyzer

MBean	managed bean
MDB	message-driven bean
MQ	message queue
MTDS	monitored transparent decision service
NIST	National Institute of Standards and Technology
NPE	null pointer error
ODM	Operational Decision Manager
OSGi	Open Services Gateway Initiative
PaaS	Platform-as-a-Service
PMI	Performance Monitoring Infrastructure
POJO	plain old Java object
POP	point of presence
POS	point of sale
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
SAML	Security Assertion Markup Language
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)
SP	Special Publications
SPI	service provider interface
SPSS	Statistical Product and Service Solutions
SQL	Structured Query Language
SSL	Secure Sockets Layer

SSN	Social Security Number
SSO	single sign-on
SSP	Scenario Service Provider
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDS	Transparent decision service
UI	user interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
UUID	Universal Unique Identifiers
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
WXS	WebSphere eXtreme Scale
XA	eXtended Architecture
XML	Extensible Markup Language
XOM	execution object model
XSD	XML Schema Definition
XU	Execution Unit
z/OS	Z Series Operating System

Appendix B. Appendix: IBM Operational Decision Manager on Cloud

What this unit is about

This appendix describes Operational Decision Manager on Cloud.

What you should be able to do

After completing this appendix, you should be able to:

- Describe Operational Decision Manager on Cloud.



Introduction to IBM ODM on Cloud

- Enterprise-grade ODM cloud service for development, testing, and production
- Cloud-based, collaborative, and role-based environment
 - Capture, automate, and manage frequently occurring, repeatable rules-based business decisions
- Ready-to-use development, test, and production environments are available
- Monthly subscription plans
- Available exclusively on IBM Cloud infrastructure
- Managed by IBM
- Artifacts that are created with IBM ODM on Cloud are compatible with IBM ODM on-premises product

© Copyright IBM Corporation 2015

Figure B-1. Introduction to IBM ODM on Cloud

WB393 / ZB3931.0

Notes:



IBM ODM on Cloud

- Targets born-on-the-cloud projects and hybrid cloud scenarios
- Born-on-the-cloud project:
 - Development, testing, and production in the cloud
 - Prefer cloud solutions to on-premises solutions
 - Urgency for implementation
- Pilot project
 - Proving business value
 - Try a newer version of IBM Operational Decision Manager
- Development
 - Organization can manage the production solution on-premises, but needs to get started quickly to meet go-live dates

© Copyright IBM Corporation 2015

Figure B-2. IBM ODM on Cloud

WB393 / ZB3931.0

Notes:

IBM on Cloud: SoftLayer

- IBM ODM on Cloud can be provisioned in any SoftLayer data center

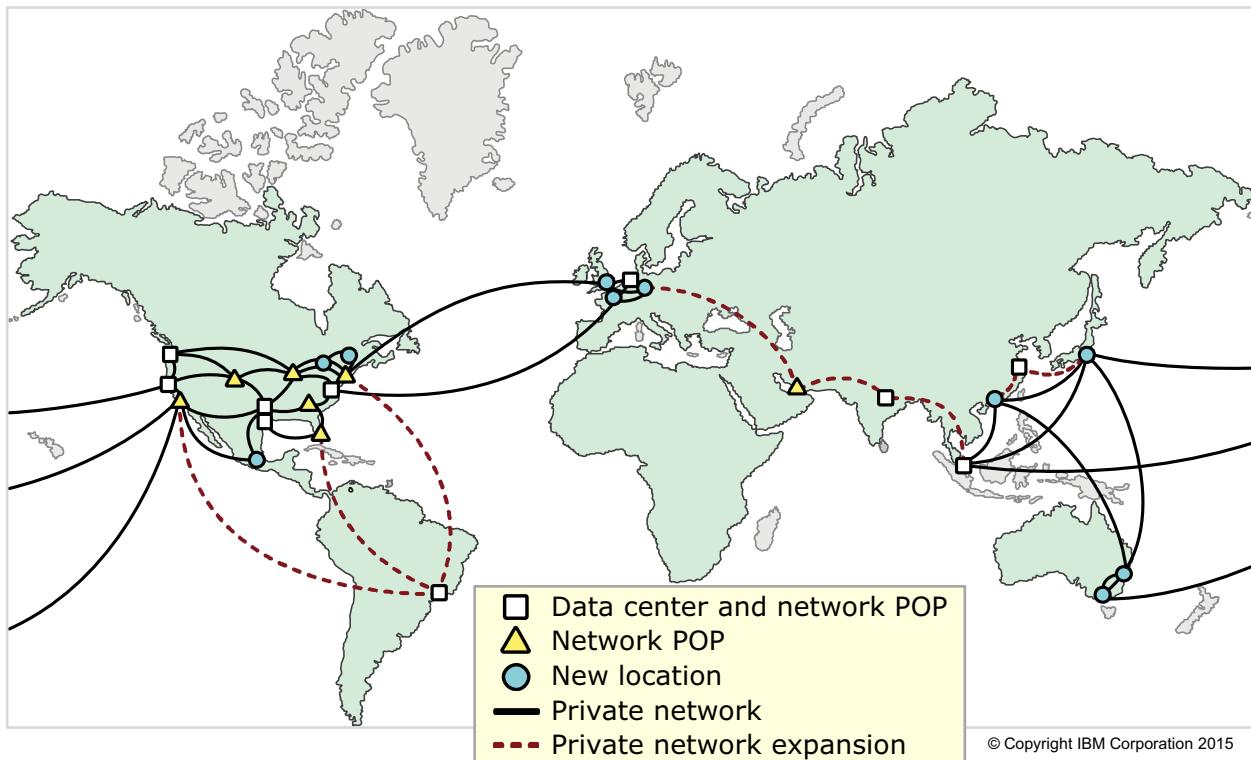


Figure B-3. IBM on Cloud: SoftLayer

WB393 / ZB3931.0

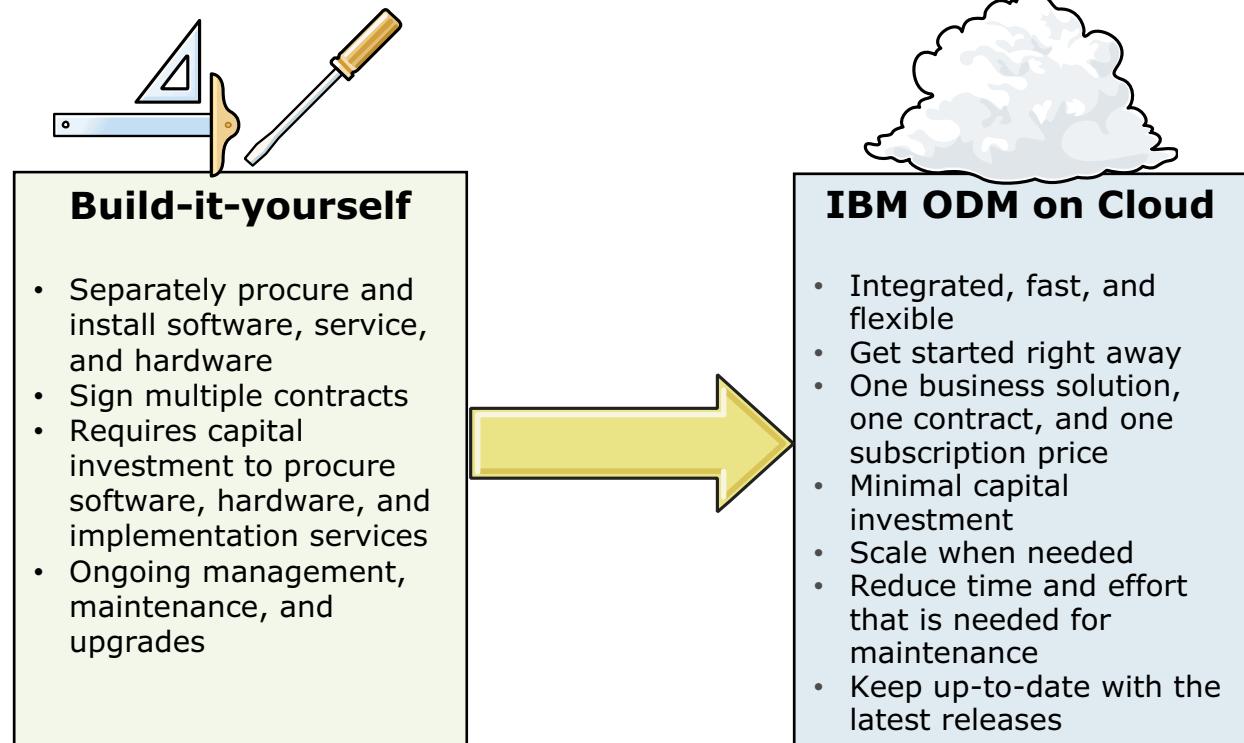
Notes:

IBM ODM on Cloud can be provisioned in any SoftLayer data center. SoftLayer has data centers across North America, Europe, and Asia. Each data center is connected to the SoftLayer global private network.

For more information about SoftLayer data centers, see:

<http://www.softlayer.com/data-centers>

Accelerate decision management solution deployment with IBM ODM on Cloud



© Copyright IBM Corporation 2015

Figure B-4. Accelerate decision management solution deployment with IBM ODM on Cloud

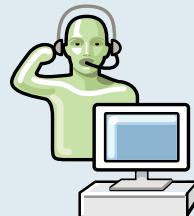
WB393 / ZB3931.0

Notes:

ODM on Cloud customer focus: Manage and automate decisions

IBM manages:

- Uptime
- Monitoring
- Backup
- High availability
- Disaster recovery
- Updates
- Maintenance



Customers manage:

- Application development
- Application integration
- Application support



© Copyright IBM Corporation 2015

Figure B-5. ODM on Cloud customer focus: Manage and automate decisions

WB393 / ZB3931.0

Notes:

IBM ODM on Cloud: Key features

<p>Develop, Test, and Run</p> <ul style="list-style-type: none"> • Each instance comes with three environments 	<p>Security</p> <ul style="list-style-type: none"> • Onboard directory, SAML to integrate with enterprise security, SSL 
<p>Clustering</p> <ul style="list-style-type: none"> • Includes clustering for service resilience and availability 	<p>Capacity and scaling</p> <ul style="list-style-type: none"> • Capacity in any IBM SoftLayer data center; additional capacity available 

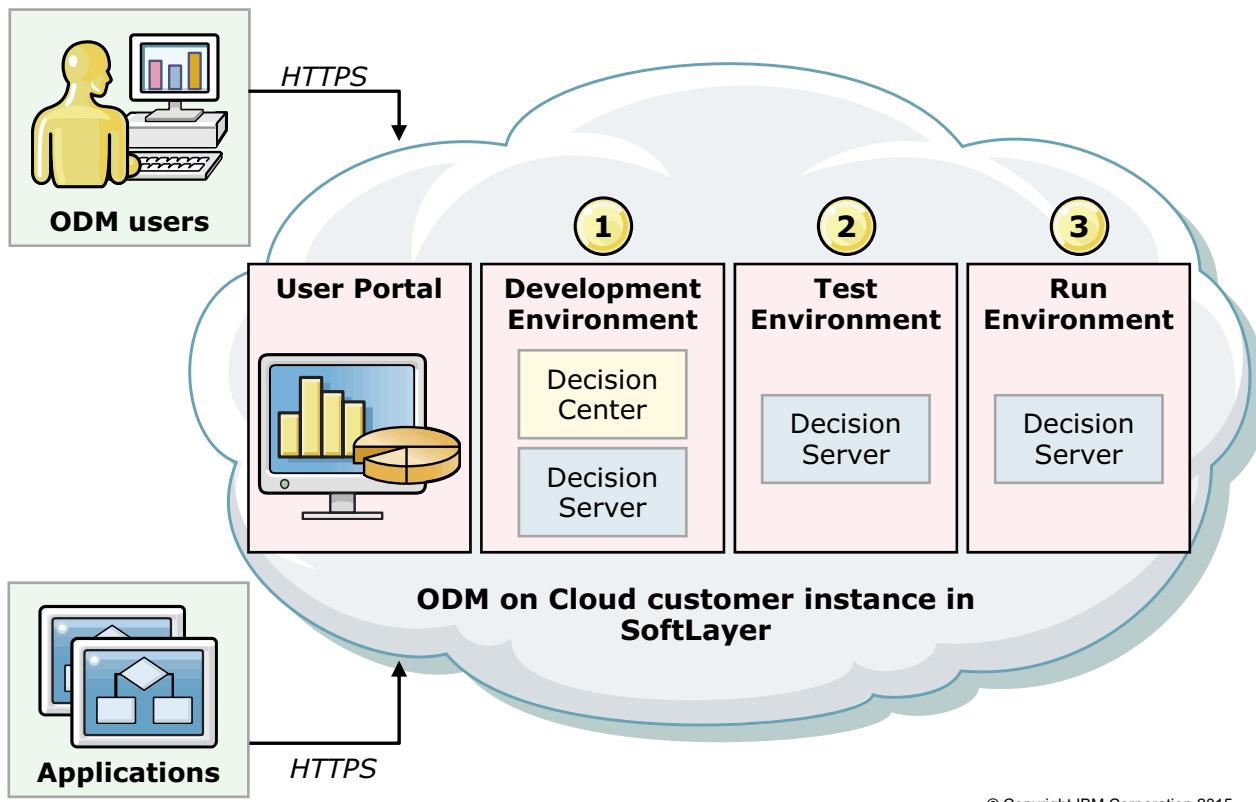
© Copyright IBM Corporation 2015

Figure B-6. IBM ODM on Cloud: Key features

WB393 / ZB3931.0

Notes:

IBM ODM on Cloud: Three runtime environments



© Copyright IBM Corporation 2015

Figure B-7. IBM ODM on Cloud: Three runtime environments

WB393 / ZB3931.0

Notes:

IBM ODM on Cloud provides three runtime environments for decision management:

1. Development
2. Test
3. Run

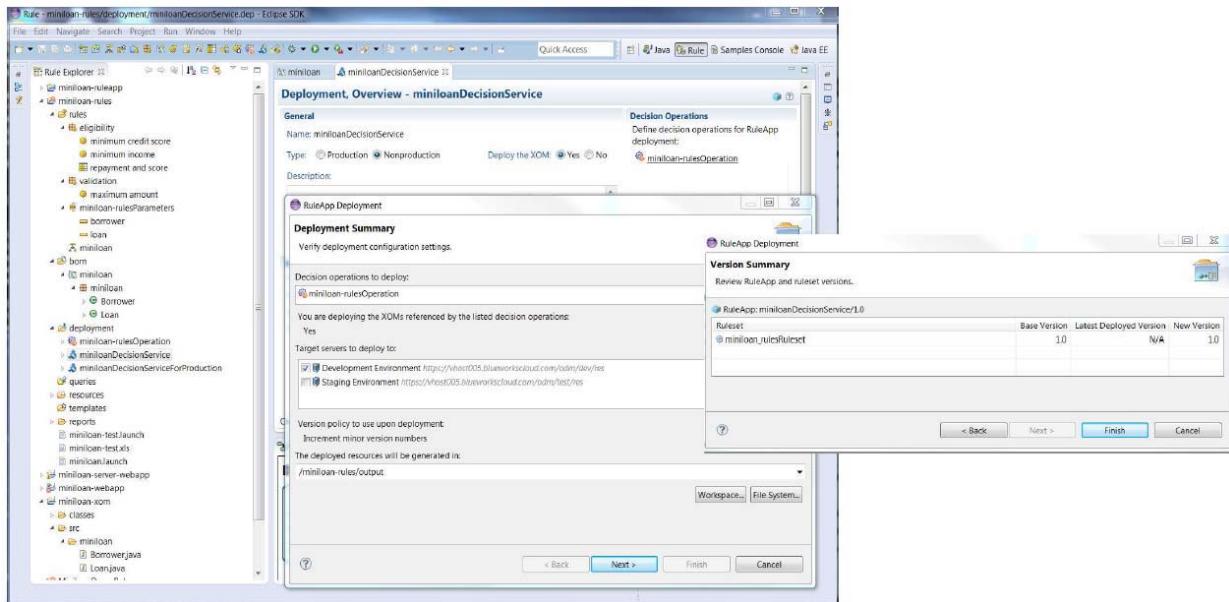
In this diagram:

- **ODM users** include developers, business analysts, business users, and rule authors who access Rule Designer, Decision Center, and the various user consoles.
- **Applications** are applications that call deployed decision services.



Rule Designer and IBM ODM Cloud

- Download Rule Designer from the portal
- Rule Designer is configured to work with the cloud instance



© Copyright IBM Corporation 2015

Figure B-8. Rule Designer and IBM ODM Cloud

WB393 / ZB3931.0

Notes:

Appendix C. 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/IBMTutorial

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/>

IBM
®