Course Guide

Developing Applications in IBM Business Process Manager Advanced V8.5.7 - II

Course code WB861 / ZB861 ERC 1.2

IBM Training

March 2017 edition

Notices

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

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

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

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

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

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

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

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

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

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

Trademarks

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

© Copyright International Business Machines Corporation 2017.
This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Contents

Trademarks . . . . . **xvi**

Agenda . . . . . **xix**

Unit 1. Introduction to IBM Business Process Manager V8.5.7 . . . . . **1-1**
How to check online for course material updates 1-2
Unit objectives 1-3
Topics 1-4
1.1. Introduction to business process management 1-5
Introduction to business process management 1-6
A typical manual system: Loan application 1-7
Challenges with the old system 1-8
Loan application as an automated business process 1-9
Services and service-oriented architecture 1-10
Evolution of enterprise integration toward SOA 1-11
SOA model: Open standards with IBM extensions 1-13
Business processes support business goals 1-14
Business Process Management 1-15
Business process management and IBM BPM solution 1-16
1.2. IBM Business Process Manager. 1-17
IBM Business Process Manager 1-18
IBM Business Process Manager 1-19
Business integration roles in IBM BPM development 1-20
Three editions of IBM Business Process Manager 1-21
IBM Business Process Manager Standard (1 of 2) 1-22
IBM Business Process Manager Standard (2 of 2) 1-23
IBM Business Process Manager Advanced (1 of 2) 1-24
IBM Business Process Manager Advanced (2 of 2) 1-25
Major tools in IBM Business Process Manager Advanced 1-26
IBM Process Designer 1-27
IBM Integration Designer 1-28
IBM Process Center 1-29
Process Portal 1-30
IBM Process Server (1 of 2) 1-32
IBM Process Server (2 of 2) 1-33
IBM Process Server components 1-34
User roles and tools in a typical deployment scenario 1-35
Introduction to IBM BPM on Cloud 1-36
BPM on Cloud customer focus: Manage and automate decisions 1-37
IBM BPM on Cloud: Three runtime environments 1-38
IBM BPM on Cloud user portal 1-39
Unit summary 1-40
Review questions 1-41
Review answers 1-42
References 1-43
Exercise 1 1-44
Exercise objectives 1-45
WB861 lab environment 1-46
Components that are required for Exercise 1 (1 of 2) 1-47
Components that are required for Exercise 1 (2 of 2) 1-48

© Copyright IBM Corp. 2017

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

Account verification process: Overview . . . 1-49
Course business scenario: What you work with in Ex 1 (1 of 2) . . . 1-51
Course business scenario: What you work with in Ex 1 (2 of 2) . . . 1-52

Unit 2. Course business scenario . . . 2-1
Unit objectives . . . 2-2
Some important points . . . 2-3
WB861 lab environment . . . 2-4
Exercise 1 . . . 2-5
What are some key tasks that you did in Exercise 1? . . . 2-6
Components that are required for Exercise 1 (1 of 2) . . . 2-7
Components that are required for Exercise 1 (2 of 2) . . . 2-8
Account verification process: Overview . . . 2-9
Course business scenario: What you work with in Ex 1 (1 of 2) . . . 2-11
Course business scenario: What you work with in Ex 1 (2 of 2) . . . 2-12
Exercise 2 . . . 2-13
What are some key tasks that you do in Exercise 2? . . . 2-14
Components that are required for Exercise 2 . . . 2-15
Run the serviceDeploy command to create a version-aware EAR file in Exercise 2 . . . 2-16
Course business scenario: What you work with in Ex 2 . . . 2-17
Exercise 3 . . . 2-18
What are some key tasks that you do in Exercise 3? . . . 2-19
Components that are required for Exercise 3 . . . 2-21
Enable the event sequencing qualifier in Exercise 3 . . . 2-22
Course business scenario: What you work with in Ex 3 (1 of 2) . . . 2-23
Course business scenario: What you work with in Ex 3 (2 of 2) . . . 2-24
Exercise 4 . . . 2-25
What are some key tasks that you do in Exercise 4? . . . 2-26
Components that are required for Exercise 4 . . . 2-27
Create fault handlers to catch and throw faults in Exercise 4 . . . 2-29
Course business scenario: What you work with in Ex 4 . . . 2-30
Exercise 5 . . . 2-31
What are some key tasks that you do in Exercise 5? . . . 2-32
Components that are required for Exercise 5 . . . 2-33
Building specific blocks of the AccountVerification process in Exercise 5 . . . 2-34
Course business scenario: What you work with in Ex 5 . . . 2-35
Exercise 6 . . . 2-36
What are some key tasks that you do in Exercise 6? . . . 2-37
Components that are required for Exercise 6 . . . 2-38
Explore SodaMachine business state machine in Exercise 6 . . . 2-39
Course business scenario: What you work with in Ex 6 . . . 2-40
Exercise 7 . . . 2-41
What are some key tasks that you do in Exercise 7? . . . 2-42
Components that are required for Exercise 7 . . . 2-43
Explore Create Account Solution diagram in Exercise 7 . . . 2-44
Course business scenario: What you work with in Ex 7 . . . 2-45
Exercise 8 . . . 2-46
What are some key tasks that you do in Exercise 8? . . . 2-47
Components that are required for Exercise 8 . . . 2-48
Create the CreditScoreProviderSelector component in Exercise 8 . . . 2-49
Course business scenario: What you work with in Ex 8 . . . 2-50
Exercise 9 . . . 2-51
What are some key tasks that you do in Exercise 9? . . . 2-52
Components that are required for Exercise 9 . . . 2-53
Create the CountryCode static relationship in Exercise 9 . . . 2-54
Course business scenario: What you work with in Ex 9 (1 of 2) . . . 2-55

© Copyright IBM Corp. 2017

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

Course business scenario: What you work with in Ex 9 (2 of 2) 2-56
Exercise 10 2-57
What are some key tasks that you do in Exercise 10? 2-58
Components that are required for Exercise 10 2-59
Implement the CreditScoreMediation in Exercise 10 2-60
Course business scenario: What you with in Ex 10 2-62
Exercise 11 2-63
What are some key tasks that you do in Exercise 11? 2-64
Components that are required for Exercise 11 2-65
Implement the CreditScoreErrorMediation subflow in Ex 11 2-66
Course business scenario: What you work with in Ex 11 2-67
Exercise 12 2-68
What are some key tasks that you do in Exercise 12? 2-69
Components that are required for Exercise 12 2-70
Create users and groups in administrative console in Exercise 12 2-71
Manage application security by using security qualifier in Ex 12 2-72
Create LTPA encryption by using global security in Exercise 12 (1 of 2) 2-73
Create LTPA encryption by using global security in Exercise 12 (2 of 2) 2-74
Test task security in Business Process Choreographer Explorer in Exercise 12 (1 of 2) 2-75
Test task security in Business Process Choreographer Explorer in Exercise 12 (2 of 2) 2-76
Course business scenario: What you work with in Ex 12 (1 of 2) 2-77
Course business scenario: What you work with in Ex 12 (2 of 2) 2-78
Exercise 13 2-79
What are some key tasks that you do in Exercise 13? 2-80
Components that are required for Exercise 13 2-81
Test governance template in Process Center in Ex 13 2-82
Course business scenario: What you work with in Ex 13 (1 of 2) 2-83
Course business scenario: What you work with in Ex 13 (2 of 2) 2-84
Exercise 14 2-85
What are some key tasks that you do in Exercise 14? 2-86
Components that are required for Exercise 14 2-87
Unit summary 2-88

Unit 3. Version control and testing of SCA applications 3-1
Unit objectives 3-2
Topics 3-3
3.1. Understanding version control in SCA components 3-4
Understanding version control in SCA components 3-5
What is version control? 3-6
What can be versioned in IBM Integration Designer? 3-7
Version control for SCA imports 3-8
Version control for modules and libraries 3-9
Version control of business process templates (1 of 2) 3-11
Version control of business process templates (2 of 2) 3-12
Running instances with the old version 3-13
Process instance migration 3-14
Early binding (static binding) 3-15
Long-running process dynamicity: Late binding (1 of 2) 3-16
Long-running process dynamicity: Late binding (2 of 2) 3-17
3.2. Considerations during version control 3-19
Considerations during version control 3-20
Considerations when using version control 3-21
Considerations for deploying versioned modules 3-22
Late binding consideration 3-23
Intelligent packing decisions 3-24
Using source control management (SCM) 3-25

© Copyright IBM Corp. 2017

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

3.3. Cross-component trace capabilities and settings . . . 3-26
Cross-component trace capabilities and settings . . . 3-27
Cross-component trace capabilities . . . 3-28
Use cross-component trace for problem determination . . . 3-29
Settings in cross-component trace . . . 3-30
Activating cross-component trace in production . . . 3-31
Activating cross-component trace in testing . . . 3-32
Server Logs view . . . 3-33
Trace records in detail . . . 3-35
Troubleshoot SCA call chains by using the Cross-Component Trace view (1 of 2) . . . 3-36
Troubleshoot SCA call chains by using the Cross-Component Trace view (2 of 2) . . . 3-37
Call and log sequence . . . 3-38
3.4. Testing and logging with the Component Test Explorer . . . 3-39
Testing and logging with the Component Test Explorer . . . 3-40
Component test projects (1 of 2) . . . 3-41
Component test projects (2 of 2) . . . 3-43
Working with test cases (1 of 2) . . . 3-44
Working with test cases (2 of 2) . . . 3-45
Using emulation in test cases (1 of 2) . . . 3-46
Using emulation in test cases (2 of 2) . . . 3-47
Building a new emulation rule . . . 3-48
Using the scheduler . . . 3-49
IBM Integration Designer problem determination . . . 3-50
The IBM Support Assistant . . . 3-52
Unit summary . . . 3-53
Review questions . . . 3-54
Review answers . . . 3-55
Exercise 2 . . . 3-56
Exercise objectives . . . 3-57
Components that are required for Exercise 2 . . . 3-58
Run the serviceDeploy command to create a version-aware EAR file in Exercise 2 . . . 3-59
Course business scenario: What you work with in Ex 2 . . . 3-60

Unit 4. SCA bindings, the SCA runtime, and SCA quality of service qualifiers . . . 4-1
Unit objectives . . . 4-2
Topics . . . 4-3
4.1. SCA runtime . . . 4-4
SCA runtime . . . 4-5
SCA concepts: Interactions . . . 4-6
SCA invocation styles . . . 4-7
Reference: SCA synchronous intramodule . . . 4-8
Reference: SCA synchronous intermodule (one server) . . . 4-9
Reference: SCA synchronous intermodule (two servers) . . . 4-10
SCA-to-runtime artifact mapping: Asynchronous . . . 4-11
Asynchronous messaging resources at deployment . . . 4-12
SCA asynchronous intramodule . . . 4-13
SCA asynchronous intermodule (one server) . . . 4-14
4.2. Failed Event Manager . . . 4-15
Failed Event Manager . . . 4-16
Failed Event Manager . . . 4-17
Failed Event Manager event types . . . 4-18
Managing failed events . . . 4-20
Events, notifications, and exceptions (1 of 3) . . . 4-21
Events, notifications, and exceptions (2 of 3) . . . 4-22
Events, notifications, and exceptions (3 of 3) . . . 4-23
SCA asynchronous intermodule failures . . . 4-24

© Copyright IBM Corp. 2017

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

Recovery service runtime (1 of 3) . . . 4-25
Recovery service runtime (2 of 3) . . . 4-26
Recovery service runtime (3 of 3) . . . 4-27
4.3. SCA bindings . . . 4-28
SCA bindings . . . 4-29
Incoming and outgoing interactions with external components . . . 4-30
Export binding types . . . 4-32
Import binding types . . . 4-33
JMS bindings (1 of 2) . . . 4-34
JMS bindings (2 of 2) . . . 4-35
Generic JMS bindings . . . 4-36
Import and export resources . . . 4-37
Binding resource example: JMS binding . . . 4-38
4.4. SCA quality of service qualifiers . . . 4-39
SCA quality of service qualifiers . . . 4-40
Quality of service qualifiers . . . 4-41
Qualifiers that are supported in IBM Integration Designer (1 of 3) . . . 4-43
Qualifiers that are supported in IBM Integration Designer (2 of 3) . . . 4-44
Qualifiers that are supported in IBM Integration Designer (3 of 3) . . . 4-45
Event sequencing . . . 4-46
Setting the event sequencing qualifier . . . 4-47
Event sequencing (ES) runtime . . . 4-48
Which components need event sequencing? . . . 4-49
Enable event sequencing: JMS exports . . . 4-50
Enable event sequencing: IBM MQ JMS and IBM MQ exports . . . 4-51
Enable event sequencing: EIS export . . . 4-52
esAdmin command-line interface . . . 4-53
Unit summary . . . 4-54
Review questions . . . 4-55
Review answers . . . 4-56
Exercise 3 . . . 4-57
Exercise objectives . . . 4-58
Components that are required for Exercise 3 . . . 4-59
Enable the event sequencing qualifier in Exercise 3 . . . 4-60
Course business scenario: What you work with in Ex 3 (1 of 2) . . . 4-61
Course business scenario: What you work with in Ex 3 (2 of 2) . . . 4-62

Unit 5. Fault handlers . . . 5-1
Unit objectives . . . 5-2
Exception handling in business processes . . . 5-3
Fault types . . . 5-5
Examine user-defined fault types with the business object editor . . . 5-6
Sources of user-defined faults . . . 5-7
Raise a fault with the throw activity . . . 5-8
Raise a fault from a Java snippet . . . 5-9
Return a fault by using the reply activity (1 of 2) . . . 5-11
Return a fault by using the reply activity (2 of 2) . . . 5-12
Manage faults with fault handlers . . . 5-13
Fault handling behavior . . . 5-14
Add a fault handler to a business process . . . 5-15
Catch a specific fault message . . . 5-16
Continue processing upon unhandled faults feature . . . 5-17
Expirations for long-running service invocations . . . 5-19
Generalized flow to model faults graphically (1 of 3) . . . 5-20
Generalized flow to model faults graphically (2 of 3) . . . 5-21
Generalized flow to model faults graphically (3 of 3) . . . 5-22

© Copyright IBM Corp. 2017

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

Catch Advanced Integration services (AIS) exceptions (1 of 2) 5-23
Catch Advanced Integration services (AIS) exceptions (2 of 2) 5-24
Unit summary 5-25
Review questions 5-26
Review answers 5-27
Exercise 4 5-28
Exercise objectives 5-29
Components that are required for Exercise 4 5-30
Create fault handlers to catch and throw faults in Exercise 4 5-32
Course business scenario: What you work with in Ex 4 5-33

Unit 6. WS-BPEL compensation and event handlers 6-1
Unit objectives 6-2
The need for compensation 6-3
Characteristics of transactions – ACID test 6-4
Business process execution modes: Long-running processes 6-5
Business process execution modes: Microflows 6-6
Compensation and execution mode 6-7
Compensation in long-running processes 6-9
When does compensation occur in a long-running process? 6-10
Compensation in a microflow 6-11
When does compensation occur in a microflow? 6-12
Compensation and fault handlers 6-13
Building blocks for compensation (1 of 2) 6-14
Building blocks for compensation (2 of 2) 6-15
Add a compensation operation to an invoke activity in microflows 6-17
Add a compensation handler to a long-running process 6-19
Trigger compensation explicitly 6-20
Default compensation handler 6-21
Default fault handler 6-22
Fault handlers and compensation handlers 6-23
Compensation and subprocesses (1 of 2) 6-24
Compensation and subprocesses (2 of 2) 6-25
Failure during the compensation 6-26
Reacting to changes outside of the business process 6-27
Why use event handlers? 6-28
Event handler characteristics 6-29
Adding an event handler to a scope 6-30
Message event handler 6-31
Adding and configuring an OnEvent handler 6-32
Timeout event handler 6-33
Specifying a date-based timeout value 6-34
Specifying a duration-based timeout value 6-35
Event handler lifecycle 6-36
Unit summary 6-37
Review questions 6-38
Review answers 6-39
Exercise 5 6-40
Exercise objectives 6-41
Components that are required for Exercise 5 6-42
Building specific blocks of the AccountVerification process in Exercise 5 6-43
Course business scenario: What you work with in Ex 5 6-44

Unit 7. Business state machines. 7-1
Unit objectives 7-2
Topics 7-3

© Copyright IBM Corp. 2017

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

7.1. Introduction to business state machines . . . 7-4
Introduction to business state machines . . . 7-5
Business state machines are service components . . . 7-6
States and state machines . . . 7-7
What is a business state machine? . . . 7-9
State machines are implemented in BPEL . . . 7-11
Business state machines versus business processes . . . 7-13
State machines in IBM Process Server (1 of 2) . . . 7-14
State machines in IBM Process Server (2 of 2) . . . 7-15
Business Process Choreographer Explorer . . . 7-16
7.2. Elements of state machines . . . 7-17
Elements of state machines . . . 7-18
Overview: Elements of a state machine (1 of 2) . . . 7-19
Overview: Elements of a state machine (2 of 2) . . . 7-20
Business state machine components: States . . . 7-21
Business state machine components: Transitions . . . 7-22
Business state machine components: Conditions . . . 7-23
Business state machine components: Timeouts . . . 7-24
Business state machine components: Actions . . . 7-25
Business state machine component flow at run time . . . 7-26
Composite states (1 of 2) . . . 7-27
Composite states (2 of 2) . . . 7-28
Unit summary . . . 7-29
Review questions . . . 7-30
Review answers . . . 7-31
Exercise 6 . . . 7-32
Exercise objectives . . . 7-33
Components that are required for Exercise 6 . . . 7-34
Explore SodaMachine business state machine in Exercise 6 . . . 7-35
Course business scenario: What you work with in Ex 6 . . . 7-36

Unit 8. Transactional behavior of SCA applications . . . 8-1
Unit objectives . . . 8-2
Topics . . . 8-3
8.1. Transactional behavior concepts . . . 8-4
Transactional behavior concepts . . . 8-5
Transaction concepts review . . . 8-6
Local versus global transactions . . . 8-7
Rationale for multiple transactions in a process . . . 8-8
Transaction behavior settings in activities . . . 8-9
Activity types and transaction boundaries . . . 8-10
Activities that commit and start new transactions . . . 8-11
Shifting transaction boundaries . . . 8-12
Activities that end (commit) transactions . . . 8-13
Fault handling and transactions . . . 8-14
Activities with no influence on transactions . . . 8-15
Transactional behavior: Settings for activities . . . 8-16
Transactional behavior: Commit before (1 of 3) . . . 8-17
Transactional behavior: Commit before (2 of 3) . . . 8-18
Transactional behavior: Commit before (3 of 3) . . . 8-19
Transactional behavior: Commit after (1 of 3) . . . 8-20
Transactional behavior: Commit after (2 of 3) . . . 8-21
Transactional behavior: Commit after (3 of 3) . . . 8-22
Transactional behavior: Participates (1 of 2) . . . 8-23
Transactional behavior: Participates (2 of 2) . . . 8-24
Transactional behavior: Requires own (1 of 4) . . . 8-25

© Copyright IBM Corp. 2017

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

Transactional behavior: Requires own (2 of 4) . . . 8-26
Transactional behavior: Requires own (3 of 4) . . . 8-27
Transactional behavior: Requires own (4 of 4) . . . 8-28
Transactions affecting parallel execution . . . 8-29
Long-running processes and parallelism . . . 8-30
Microflows and parallelism . . . 8-31
Uncertain transaction boundaries with participates transaction behavior . . . 8-32
8.2. Transactional behavior during failure . . . 8-33
Transactional behavior during failure . . . 8-34
Transaction handling during failure (1 of 5) . . . 8-35
Transaction handling during failure (2 of 5) . . . 8-36
Transaction handling during failure (3 of 5) . . . 8-37
Transaction handling during failure (4 of 5) . . . 8-38
Transaction handling during failure (5 of 5) . . . 8-39
Unit summary . . . 8-40
Review questions . . . 8-41
Review answers . . . 8-42
Exercise 7 . . . 8-43
Exercise objectives . . . 8-44
Components that are required for Exercise 7 . . . 8-45
Explore Create Account Solution diagram in Exercise 7 . . . 8-46
Course business scenario: What you work with in Ex 7 . . . 8-47

Unit 9. Business process flexibility . . . 9-1
Unit objectives . . . 9-2
Topics . . . 9-3
9.1. Selectors . . . 9-4
Selectors . . . 9-5
Selectors are a supporting service component . . . 9-6
Overview of selectors . . . 9-7
Selector examples . . . 9-8
Selector components . . . 9-9
Business value of selectors . . . 9-10
Selectors at run time . . . 9-11
Selector auditing . . . 9-13
9.2. Dynamic routing . . . 9-14
Dynamic routing . . . 9-15
Dynamic invocation overview (1 of 2) . . . 9-16
Dynamic invocation overview (2 of 2) . . . 9-17
Example: Dynamic invocation that uses the EndpointReference APIs with HTTP binding . . . 9-18
Dynamic endpoint selection in mediations (1 of 2) . . . 9-19
Dynamic endpoint selection in mediations (2 of 2) . . . 9-20
Endpoint lookup in mediations . . . 9-21
Dynamic invocation by using the SMO structure . . . 9-23
9.3. Repair process navigation . . . 9-25
Repair process navigation . . . 9-26
Repair and dynamicity features . . . 9-27
Jump between activities of a process . . . 9-28
Skip activities within a process . . . 9-29
Update variables of a process . . . 9-30
Repair and dynamicity features . . . 9-31
Unit summary . . . 9-32
Review questions . . . 9-33
Review answers . . . 9-34
Exercise 8 . . . 9-35
Exercise objectives . . . 9-36

© Copyright IBM Corp. 2017

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

Components that are required for Exercise 8 . . . 9-37
Create the CreditScoreProviderSelector component in Exercise 8 . . . 9-38
Course business scenario: What you work with in Ex 8 . . . 9-39

Unit 10. Relationships . . . 10-1
Unit objectives . . . 10-2
Topics . . . 10-3
10.1. Introduction to relationships . . . 10-4
Introduction to relationships . . . 10-5
Relationships are a supporting service . . . 10-6
Overview: Relationships . . . 10-7
Relationship components . . . 10-8
10.2. Dynamic relationships. . . . 10-9
Dynamic relationships . . . 10-10
Relationship types: Dynamic relationships . . . 10-11
Identity relationships in IBM Integration Designer (1 of 2) . . . 10-12
Identity relationships in IBM Integration Designer (2 of 2) . . . 10-13
Managed roles in identity relationships . . . 10-14
Example: Identity relationship . . . 10-15
10.3. Static relationships . . . 10-17
Static relationships . . . 10-18
Relationship types: Static relationship . . . 10-19
Static relationship: Defining the components . . . 10-20
Static relationship: Defining the instance data . . . 10-21
Static relationship: Using the relationship lookup (1 of 2) . . . 10-22
Static relationship: Using the relationship lookup (2 of 2) . . . 10-23
Static relationship: BO map implementation (1 of 2) . . . 10-24
Static relationship: BO map implementation (2 of 2) . . . 10-25
10.4. Runtime aspects of relationships . . . 10-26
Runtime aspects of relationships . . . 10-27
Runtime components of relationships . . . 10-28
Relationship data constituents (1 of 2) . . . 10-29
Relationship data constituents (2 of 2) . . . 10-30
Relationships: Calling context and verbs . . . 10-31
Overview: Calling contexts . . . 10-32
10.5. Managing relationships. . . . 10-33
Managing relationships . . . 10-34
Relationship Manager . . . 10-35
Instance creation through Relationship Manager . . . 10-36
Populating the relationship instance table by using database views . . . 10-37
Unit summary . . . 10-38
Review questions . . . 10-39
Review answers . . . 10-40
Exercise 9 . . . 10-41
Exercise objectives . . . 10-42
Components that are required for Exercise 9 . . . 10-43
Create the CountryCode static relationship in Exercise 9 . . . 10-44
Course business scenario: What you work with in Ex 9 (1 of 2) . . . 10-45
Course business scenario: What you work with in Ex 9 (2 of 2) . . . 10-46

Unit 11. Mediation primitives. . . . 11-1
Unit objectives . . . 11-2
Topics . . . 11-3
11.1. Miscellaneous mediation primitives . . . 11-4
Miscellaneous mediation primitives . . . 11-5
Data handler primitive . . . 11-6

© Copyright IBM Corp. 2017

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

Database lookup primitive . . . 11-8
Set message type primitive . . . 11-9
Custom mediation primitive . . . 11-10
Custom mediation: Error handling . . . 11-11
11.2. Message validation and logging primitives . . . 11-12
Message validation and logging primitives . . . 11-13
Message validator primitive . . . 11-14
Event emitter primitive . . . 11-15
Message logger primitive . . . 11-16
Trace primitive . . . 11-18
11.3. Starting services statically and aggregating messages . . . 11-19
Starting services statically and aggregating messages . . . 11-20
Callout node . . . 11-21
Service invoke primitive . . . 11-22
Service message object contexts . . . 11-23
Fan-out and fan-in primitives: Overview . . . 11-25
Fan-out primitive (1 of 2) . . . 11-26
Fan-out primitive (2 of 2) . . . 11-27
Fan-in primitive (1 of 2) . . . 11-28
Fan-in primitive (2 of 2) . . . 11-29
Aggregation and shared context . . . 11-30
SMO structure: Context shared . . . 11-31
SMO structure: Context fan-out . . . 11-32
Fan-out and fan-in: Aggregation example . . . 11-33
Unit summary . . . 11-35
Review questions . . . 11-36
Review answers . . . 11-37
Exercise 10 . . . 11-38
Exercise objectives . . . 11-39
Components that are required for Exercise 10 . . . 11-40
Implement the CreditScoreMediation in Exercise 10 . . . 11-41
Course business scenario: What you with in Ex 10 . . . 11-43

Unit 12. Mediation flow control . . . 12-1
Unit objectives . . . 12-2
Topics . . . 12-3
12.1. Flow control with mediation primitives . . . 12-4
Flow control with mediation primitives . . . 12-5
Mediation primitives: Routing (1 of 2) . . . 12-6
Mediation primitives: Routing (2 of 2) . . . 12-7
Flow order primitive . . . 12-8
Flow order primitive: Example . . . 12-9
Message filter primitive . . . 12-10
Message filter primitive: Usage examples . . . 12-11
Message filter primitive: Configuration . . . 12-12
Type filter primitive . . . 12-14
Mediation subflow: Review . . . 12-15
Mediation subflow: Implementation . . . 12-17
Mediation subflow: Restrictions . . . 12-18
12.2. Event sequencing . . . 12-19
Event sequencing . . . 12-20
Event sequencing . . . 12-21
Event sequencing: How it works . . . 12-22
Event sequencing: Defining the keys . . . 12-23
Event sequencing in IBM Process Server . . . 12-24
12.3. Starting services dynamically . . . 12-25

© Copyright IBM Corp. 2017

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

Starting services dynamically . . . 12-26
Introduction to dynamic message processing . . . 12-27
Mediation primitives: Dynamic routing . . . 12-28
WebSphere Service Registry and Repository . . . 12-29
WebSphere Service Registry and Repository Registry Definition . . . 12-31
Endpoint lookup primitive: Overview . . . 12-32
Endpoint lookup primitive: Actions . . . 12-34
UDDI endpoint lookup primitive: Overview . . . 12-35
Endpoint lookup component interaction . . . 12-36
Mediation policies and dynamic service interactions . . . 12-37
Endpoint lookup (ELP) promoted properties . . . 12-39
Gateway endpoint lookup primitive: Overview . . . 12-40
SLA check primitive: Overview . . . 12-41
Service level agreements . . . 12-42
Policy resolution primitive: Overview . . . 12-43
Interaction with mediation policy . . . 12-44
Unit summary . . . 12-46
Review questions . . . 12-47
Review answers . . . 12-48
Exercise 11 . . . 12-49
Exercise objectives . . . 12-50
Components that are required for Exercise 11 . . . 12-51
Implement the CreditScoreErrorMediation subflow in Ex 11 . . . 12-52
Course business scenario: What you work with in Ex 11 . . . 12-53

Unit 13. Security . . . 13-1
Unit objectives . . . 13-2
Topics . . . 13-3
13.1. Securing IBM Process Server runtime engine . . . 13-4
Securing IBM Process Server runtime engine . . . 13-5
IBM Business Process Manager: Users and groups management (1 of 2) . . . 13-6
IBM Business Process Manager: Users and groups management (2 of 2) . . . 13-7
IBM Process Server security overview . . . 13-8
Enabling administrative security . . . 13-9
Enable application security . . . 13-10
Enabling Java EE security . . . 13-11
Creating profiles with security (1 of 2) . . . 13-12
Creating profiles with security (2 of 2) . . . 13-13
Configuring a user account repository (1 of 2) . . . 13-14
Configuring a user account repository (2 of 2) . . . 13-16
Security domains . . . 13-17
Security domain configuration . . . 13-18
13.2. Application security overview . . . 13-19
Application security overview . . . 13-20
Elements of application level security: Authentication . . . 13-21
Elements of application level security: Authorization . . . 13-22
Elements of application level security: Data integrity and privacy (1 of 2) . . . 13-23
Elements of application level security: Data integrity and privacy (2 of 2) . . . 13-24
Supported authentication mechanisms . . . 13-25
Application security overview . . . 13-26
Authentication for web applications . . . 13-27
Authentication for Enterprise JavaBeans applications . . . 13-28
Role-based resource security . . . 13-30
System and resource roles . . . 13-31
Owner, writer, and reader resource roles . . . 13-32
13.3. SCA security qualifiers . . . 13-33

© Copyright IBM Corp. 2017

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

SCA security qualifiers . . . 13-34
Using security-related quality of service qualifiers . . . 13-35
Security permission qualifier (1 of 3) . . . 13-36
Security permission qualifier (2 of 3) . . . 13-37
Security permission qualifier (3 of 3) . . . 13-38
Security identity qualifier (1 of 4) . . . 13-39
Security identity qualifier (2 of 4) . . . 13-40
Security identity qualifier (3 of 4) . . . 13-41
Security identity qualifier (4 of 4) . . . 13-42
Security context propagation . . . 13-43
13.4. Securing business processes . . . 13-44
Securing business processes . . . 13-45
Securing business processes: Roles . . . 13-46
Work items: Roles to operation mapping for processes and activities . . . 13-48
Securing business processes: Create and start processes . . . 13-49
Securing business processes: Interact with processes . . . 13-51
Securing business processes: Administer processes . . . 13-53
Assigning administrative tasks to processes and activities . . . 13-55
Change process ownership . . . 13-56
Unit summary . . . 13-57
Review questions . . . 13-58
Review answers . . . 13-59
Exercise 12 . . . 13-60
Exercise objectives . . . 13-61
Components that are required for Exercise 12 . . . 13-62
Create users and groups in administrative console in Exercise 12 . . . 13-63
Manage application security using security qualifier in Ex 12 . . . 13-64
Create LTPA encryption using global security in Ex 12 (1 of 2) . . . 13-65
Create LTPA encryption using global security in Ex 12 (2 of 2) . . . 13-66
Test task security in Business Process Choreographer Explorer in Exercise 12 (1 of 2) . . . 13-67
Test task security in Business Process Choreographer Explorer in Exercise 12 (2 of 2) . . . 13-68
Course business scenario: What you work with in Ex 12 (1 of 2) . . . 13-69
Course business scenario: What you work with in Ex 12 (2 of 2) . . . 13-70

Unit 14. Governance of process applications . . . 14-1
Unit objectives . . . 14-2
Governance of process applications: Overview . . . 14-3
What is a governance process? . . . 14-4
Sample governance process application . . . 14-6
Release the governance process . . . 14-7
Enable governance for a process application . . . 14-8
Installing snapshots on IBM Process Server . . . 14-9
Status messages . . . 14-10
Triggered governance process (1 of 2) . . . 14-11
Triggered governance process (2 of 2) . . . 14-12
Unit summary . . . 14-13
Review questions . . . 14-14
Review answers . . . 14-15
Exercise 13 . . . 14-16
Exercise objectives . . . 14-17
Components that are required for Exercise 13 . . . 14-18
Test governance template in Process Center in Exercise 13 . . . 14-19
Course business scenario: What you work with in Ex 13 (1 of 2) . . . 14-20
Course business scenario: What you work with in Ex 13 (2 of 2) . . . 14-21
References . . . 14-22

© Copyright IBM Corp. 2017

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

Unit 15. Integrating with IBM Business Process Manager 15-1
Unit objectives 15-2
Integrating IBM BPM and Enterprise Content Management 15-3
IBM Business Process Manager and IBM MobileFirst 15-5
IBM Business Process Manager and IBM Operational Decision Manager (1 of 2) 15-7
IBM Business Process Manager and IBM Operational Decision Manager (2 of 2) 15-9
Building a decision service (1 of 2) 15-10
Building a decision service (2 of 2) 15-11
IBM Business Process Manager and IBM Business Monitor 15-12
About IBM Business Monitor 15-14
Advantages of IBM Business Monitor 15-15
Logical architecture for business activity monitoring 15-16
Creating monitor models in IBM Integration Designer 15-18
Generating a monitor model in IBM Integration Designer 15-19
Good practices when integrating IBM Business Monitor 15-20
IBM Business Process Manager and WebSphere Service Registry and Repository 15-21
Unit summary 15-23
Review questions 15-24
Review answers 15-25
Exercise 14 15-26
Exercise objectives 15-27
Components that are required for Exercise 14 15-28

Unit 16. Course summary 16-1
Unit objectives 16-2
Course objectives 16-3
Course objectives 16-4
Course objectives 16-5
To learn more on the subject 16-6
Identify other IBM Training courses 16-7
Earn an IBM Badge 16-8
Enhance your learning with IBM resources 16-9
Unit summary 16-10
Course completion 16-11

Appendix A. List of abbreviations A-1

Appendix B. Resource guide. B-1

© Copyright IBM Corp. 2017

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

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®	ClearCase®	DB™
DB2®	developerWorks®	FileNet®
IA®	IBM MobileFirst™	IMS™
PartnerWorld®	Passport Advantage®	Rational®
Redbooks®	Sametime®	WebSphere®

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

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

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

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

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

Evolution® is a trademark or registered trademark of Kenexa, an IBM Company.

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

© Copyright IBM Corp. 2017

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

Course description

Developing Applications in IBM Business Process Manager Advanced V8.5.7 - II

Duration: 5 days

Purpose

This course continues to build the skills and concepts that are learned in courses *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I* (WB860G) and *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I* (ZB860G). Those courses introduced the service-oriented architecture (SOA) approach to business process management (BPM), and covered how to build a basic process integration solution. In this course, you learn how to extend the functionality of that solution to accommodate more process integration scenarios.

Through presentations and hands-on lab exercises, you learn about the Service Component Architecture (SCA) programming model and runtime environment, XML (business object) support, Business Process Execution Language (BPEL) handlers, business state machines, and SCA transactions. You also learn about business process dynamicity and flexibility, relationships, and mediation primitives. Other topics include creating versions of SCA components, test environment capabilities, and approaches to securing SCA applications.

You use the IBM Process Center repository to add process model artifacts in a complete and integrated development solution. You learn how to apply governance to process applications.

In addition, you use IBM Integration Designer to design, develop, and test comprehensive process integration solutions that use the SCA programming model, the Service Data Object (SDO) data model, and the mediation flow programming model. You learn how to deploy these solutions to the IBM Process Server runtime environment, maintain them in the IBM Process Center repository, and test them in the IBM Integration Designer test environment.

The course also covers how to use the tools of IBM Business Process Manager for iterative (model-driven) application development. For example, you learn how to develop a process diagram in IBM Process Designer, store it in the IBM Process Center repository, and associate it with modules and libraries in IBM Integration Designer.

The lab environment for this course uses the Windows 2012 server R2 64-bit platform.

Audience

This course is designed for integration developers and other application development specialists.

Prerequisites

Before taking this course, you should successfully complete *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I* (WB860G) or *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I* (ZB860G). You should also have:

- Basic Java and Java 2 Platform, Enterprise Edition (Java EE) skills

© Copyright IBM Corp. 2017

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

- Experience or prior education on WebSphere Application Server, or experience with Rational Application Developer
- Basic Extensible Markup Language (XML) skills

Objectives

- Create and modify an application by using iterative, model-driven development
- Develop a business process diagram in IBM Process Designer
- Work with a process application snapshot in IBM Integration Designer
- Associate IBM Integration Designer artifacts with a process application snapshot
- Use artifacts from an IBM Integration Designer library in a business process diagram
- Implement and deploy SCA modules and libraries that contain version information
- Use the serviceDeploy tool to install versioned modules in an IBM Process Server profile
- Enable cross-component trace to follow the invocation sequences between SCA applications to diagnose and repair an application failure
- Implement the event sequencing quality of service qualifier in an application
- Implement a WebSphere MQ import in an application
- Implement a fault handler and compensation handler in a business process
- Examine and test a business state machine
- Examine the transaction propagation settings and transaction quality of service qualifier settings for an application
- Implement a selector component and dynamically navigate a business process instance at run time
- Implement a static (lookup) relationship
- Create a service aggregation mediation flow by using several prebuilt mediation primitives
- Create a reusable mediation subflow that contains error handling primitives
- Configure security quality of service qualifiers and implement role-based security for human tasks and business processes
- Apply governance to process applications
- Configure IBM Business Process Manager Advanced tools for integration with other applications, such as IBM Business Monitor

© Copyright IBM Corp. 2017

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

Agenda

Note

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

Day 1

(00:15) Course introduction
(01:15) Unit 1. Introduction to IBM Business Process Manager V8.5.7
(02:00) Exercise 1. Using iterative development to create applications
(01:00) Unit 2. Course business scenario
(01:30) Unit 3. Version control and testing of SCA applications
(01:00) Exercise 2. Version control for SCA applications

Day 2

(01:30) Unit 4. SCA bindings, the SCA runtime, and SCA quality of service qualifiers
(01:00) Exercise 3. Working with SCA bindings and qualifiers
(00:45) Unit 5. Fault handlers
(01:30) Exercise 4. Applying fault handlers
(00:45) Unit 6. WS-BPEL compensation and event handlers
(01:00) Exercise 5. Applying a compensation handler to WS-BPEL

Day 3

(01:00) Unit 7. Business state machines
(01:00) Exercise 6. Working with business state machines
(01:00) Unit 8. Transactional behavior of SCA applications
(01:00) Exercise 7. Defining transactional behavior in SCA applications
(01:00) Unit 9. Business process flexibility
(01:30) Exercise 8. Creating flexible business processes

Day 4

(01:00) Unit 10. Relationships
(01:00) Exercise 9. Working with static relationships
(01:30) Unit 11. Mediation primitives
(01:00) Exercise 10. Implementing a mediation flow
(01:00) Unit 12. Mediation flow control
(02:00) Exercise 11. Writing a generic error handler for IBM Process Server

© Copyright IBM Corp. 2017

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

Day 5

(01:15) Unit 13. Security
(01:30) Exercise 12. Implementing security
(00:45) Unit 14. Governance of process applications
(01:00) Exercise 13. Applying governance to process applications
(01:00) Unit 15. Integrating with IBM Business Process Manager
(01:00) Exercise 14. Integrating other applications with IBM Integration Designer
(00:15) Unit 16. Course summary

© Copyright IBM Corp. 2017

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

Unit 1. Introduction to IBM Business Process Manager V8.5.7

Estimated time

01:15

Overview

In this unit, you learn about the purpose, function, and business value of IBM Business Process Manager. The unit introduces the tools that are included with IBM Business Process Manager V8.5.7 Advanced edition.

How you will check your progress

- Checkpoint questions
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

How to check online for course material updates

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

Instructions

1. Enter this URL in your browser: http://ibm.biz/CloudEduCourses
2. On the wiki page, locate and click the **Course Information** category.
3. Find your course in the list and then click the link.
4. The wiki page displays information for the course. If an errata document is available, this page is where it is found.
5. If you want to download an attachment, such as an errata document, click the **Attachments** tab at the end of the page.

6. To save the file to your computer, click the document link and follow the dialog box prompts.

© Copyright IBM Corporation 2017

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

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe the concepts of business processes and business process management (BPM)
- Describe the IBM product editions

© Copyright IBM Corporation 2017

Figure 1-2. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Introduction to business process management
- IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 1-3. Topics

© Copyright IBM Corp. 2017

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

1.1. Introduction to business process management

© Copyright IBM Corp. 2017

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

IBM Training

Introduction to business process management

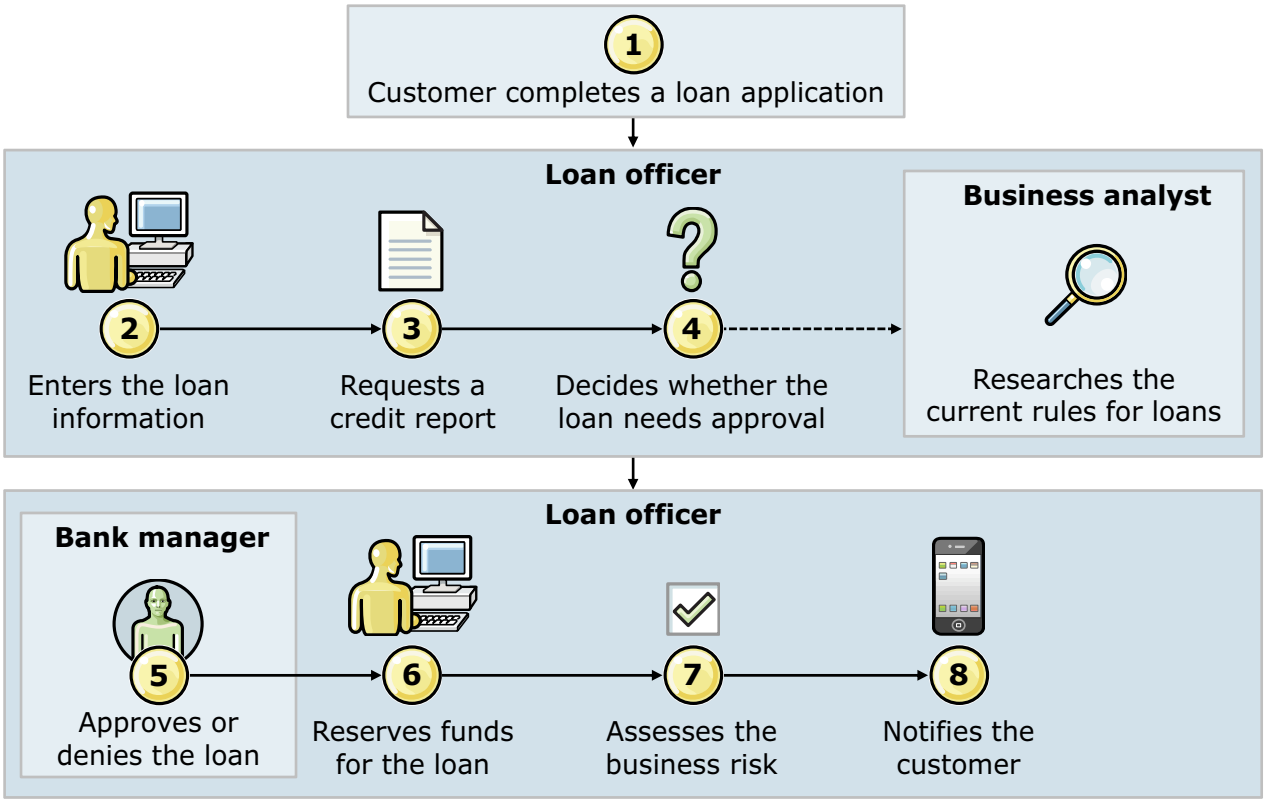
© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 1-4. Introduction to business process management

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

IBM Training

A typical manual system: Loan application



© Copyright IBM Corporation 2017

Figure 1-5. A typical manual system: Loan application

1. Customer fills out a loan application.
2. Loan officer enters the relevant data into the system.
3. Loan officer requests a credit report and receives a fax with the information.
4. Loan officer decides that this loan needs a review by a manager before it can be approved or rejected. Business analyst might need to change the rules frequently and have the developers change the system.
5. Bank manager approves or denies the loan.
6. Loan officer reserves the funds in the system.
7. Loan officer sends an email to check government watch lists to minimize risk.
8. Finally, loan officer notifies customer that the loan was approved.

© Copyright IBM Corp. 2017

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

IBM Training

Challenges with the old system

- Takes too long to process loan applications
- Paper-based human interaction in loan processing is error prone
- Many different technologies are involved (legacy, application servers, email applications)
- Integrating people, processes, and information is difficult
- Manual work is needed to “undo” work that is done when a process failure occurs
- Difficult to propagate and react to business rule changes

© Copyright IBM Corporation 2017

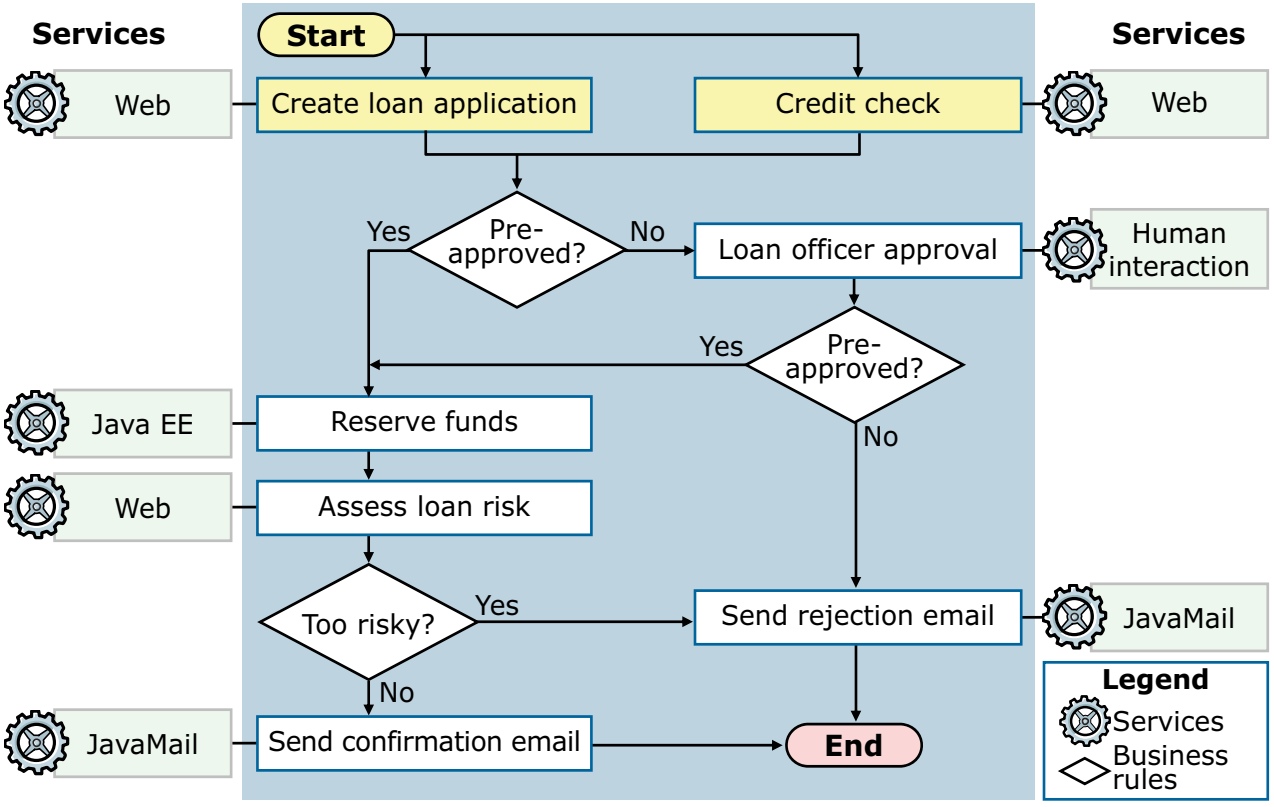
Figure 1-6. Challenges with the old system

© Copyright IBM Corp. 2017

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

IBM Training

Loan application as an automated business process



© Copyright IBM Corporation 2017

Figure 1-7. Loan application as an automated business process

© Copyright IBM Corp. 2017

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

IBM Training

Services and service-oriented architecture

- A **service** in SOA is:
  - A self-contained entity that has a distinct business function
  - Based on open standards
  - A building block for composite applications that share information within and beyond the enterprise to include customers, suppliers, and partners
- **Service-oriented architecture** (SOA) is an architectural approach to building distributed systems by using service interactions
  - SOA transforms the IT infrastructure into flexible, reusable, standards-based services
  - SOA separates interface technology from implementation
  - Knowledge of the underlying implementation details is not required to develop composite applications
- The key to business agility is to reuse and combine standardized components (services) to address changing business priorities

Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-8. Services and service-oriented architecture

The short definition of a service is a “repeatable business task”. In SOA, clients inside or outside the enterprise can invoke services. Data is exchanged between services by using well-defined interactions (messages and operations) based on open standards.

One way to provide an existing application as a service is to create an SOA gateway that converts the interfaces of the existing application into SOA-compliant interfaces. Creating this gateway involves no disruption to the application coding, so regression testing of the application is not necessary. New SOA interfaces need development and testing resources, and interruptions can occur when the new interfaces are deployed. After deployment, the application can be used through either the existing interfaces or the new SOA interfaces.

You can remodel the application into reusable SOA components and then combine those components into an SOA modular solution. This combination provides a more flexible application that can make better reuse of business logic; it benefits from SOA techniques and other modern programming practices. This way of providing existing applications as services has more effect on the existing application and its use, and is most applicable to core systems.

© Copyright IBM Corp. 2017

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

IBM Training

Evolution of enterprise integration toward SOA

- Architectures enable flexibility and reuse
- Integrate applications by using **loosely coupled** and **well-defined** interactions
  - Hide the implementation details from the service-invoking application
  - Provide a common service definition independent of a specific technology

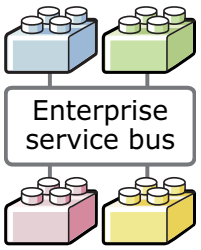
Point-to-point connectivity

- Custom point-to-point integration between technology-aware bridges and application components
- **Point-to-point systems are brittle**

EAI hub-and-spoke

- Enterprise application integration uses broker-specific adapters, providing connectivity to a wide variety of applications and environments
- **EAI is less brittle than point-to-point**

SOI with ESB



- Service-oriented integration uses service interactions and an ESB to integrate applications
- **SOI is the least brittle and is loosely coupled**

Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-9. Evolution of enterprise integration toward SOA

Point-to-point connectivity works well for direct application-to-application connection. It supports high throughput and is comparatively simple to create and deploy, but works best when you have a fairly small number of connection points that change infrequently or do not change.

Custom, point-to-point integration is expensive to maintain and expensive for new business requirements. Point-to-point integration is brittle; any change to an endpoint is likely to break it. Changes to internal components (operating systems, IP addresses, host names) can also break it.

Enterprise application integration is a hub-and-spoke integration design. This system relies on adapters to connect applications and data sources to a central hub. This design works well when you have a larger number of endpoints and when they change more frequently. When you add an endpoint or change how the endpoint connects, you need only to change one of the adapters rather than the connections to all the other endpoints. The downside of this is that it does not support high throughput well, and the communications standards are proprietary and less interoperable.

Enterprise application integration is less brittle. Changes to endpoints might require some changes, but they do not ripple through the entire infrastructure as in point-to-point integration. Changes to internal components might force changes to other components because internal broker components are typically tightly coupled. Initial costs are higher than point-to-point, but maintenance efforts are much smaller.

© Copyright IBM Corp. 2017

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

SOA blends the best of all these concepts into one new architecture, giving users capabilities that they never had before. Service-oriented integration (SOI) is the least brittle. SOI, if implemented by using web services, has standard interfaces and is much cheaper for new business requirements. If the exposed interface (Web Services Description Language) does not change, the endpoint can be swapped with no integration effect. Changes to internal components have no effect because even the internal components are loosely coupled.

Enterprises, large and small, have implemented SOA for nearly 10 years, and it is now the standard method that is used for communication. SOA builds upon previous integration techniques but does not replace them. It is important to recognize that SOA is not the end of the road either. It is the next step in the evolution of flexible infrastructures. The desire to make IT more flexible is not new. Indeed, it is as old as the IT industry itself. Each of these integration techniques has its place and is appropriate for handling certain situations. IBM has experience and history in all of these approaches.

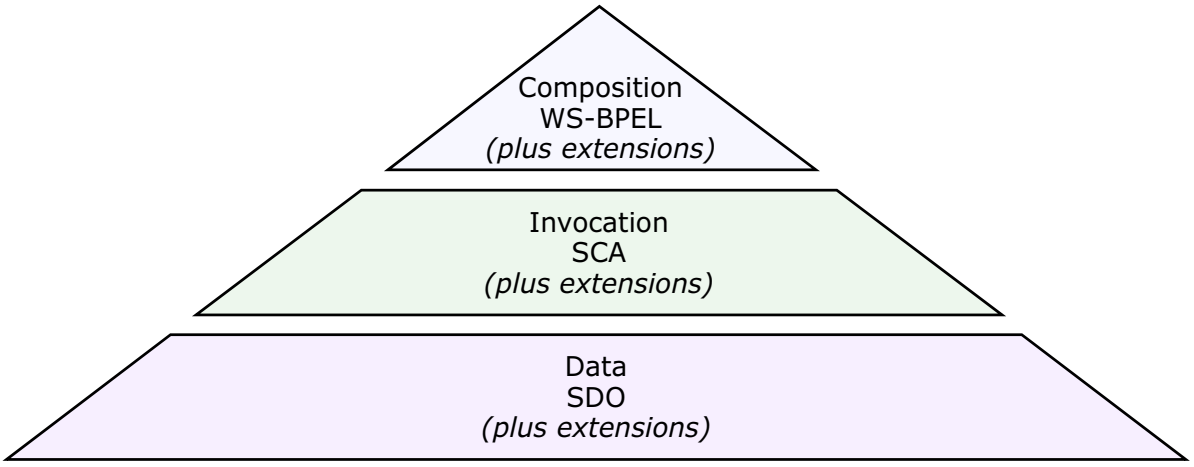
© Copyright IBM Corp. 2017

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

IBM Training

SOA model: Open standards with IBM extensions

- Data manipulation, service invocation, and process choreography are using simplified, open standards
  - Web Services Business Process Execution Language (WS-BPEL) is used to choreograph service interactions
  - Service Component Architecture is a programming framework that standardizes service invocation by isolating service implementation
  - Service Data Objects provide a standard mechanism for working with data



© Copyright IBM Corporation 2017

Figure 1-10. SOA model: Open standards with IBM extensions

© Copyright IBM Corp. 2017

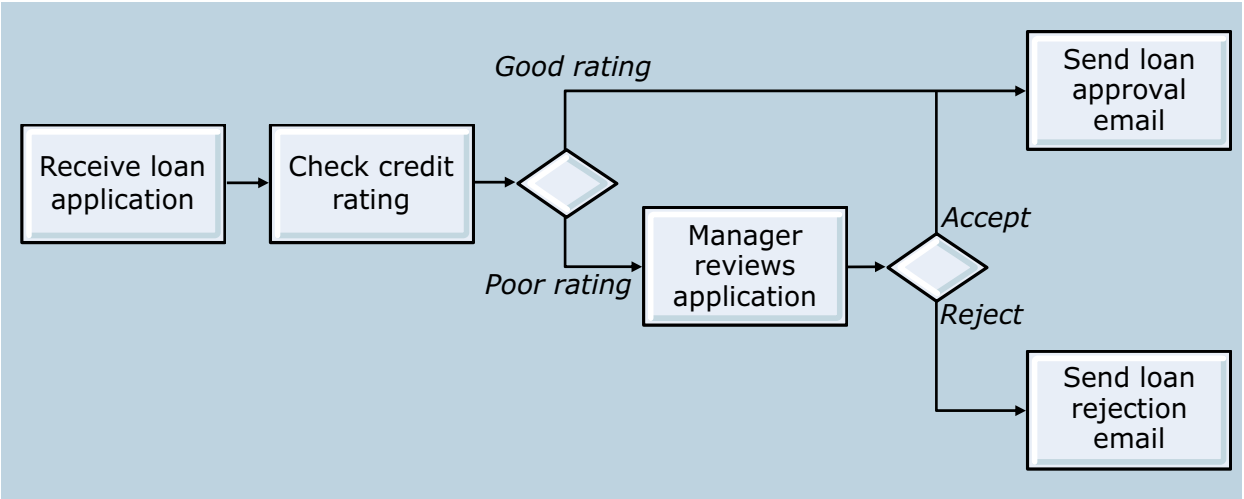
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Business processes support business goals

- A business process is a set of linked activities that create value by transforming an input into a more valuable output
  - Both input and output can be artifacts or information, or both
  - Human actors, machines, or both can do the transformation

Example:



© Copyright IBM Corporation 2017

Figure 1-11. Business processes support business goals

A *business process* is a collection of service interactions and activities that are run to fulfill a business need.

A business process defines the potential execution order of services.

The business process defines how to coordinate interactions between a process instance and its partners, and specifies how errors are handled.

In the context of a business process, partners define parties that interact with the process.

© Copyright IBM Corp. 2017

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

IBM Training

Business process management

- What is BPM?
  - BPM is not a product or technology
  - BPM is a comprehensive management approach to managing and improving the efficiency and effectiveness of business processes across the enterprise
- When to use it?
  - When complex processes are done manually or span multiple systems
  - Variations in process cannot guarantee the flow
  - Rework or updates to processes are error prone
- Why use BPM?
  - Business agility
  - End-to-end process visibility
  - Compliance
  - Efficiency
  - Governance

© Copyright IBM Corporation 2017

Figure 1-12. Business Process Management

© Copyright IBM Corp. 2017

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

IBM Training

Business process management and IBM BPM solution

- Business process management is a systematic approach to manage and improve business processes for organizations
  - Describes activities and events that manage, improve, and optimize a business process
  - The goal of BPM is to make business processes more effective, more efficient, and more capable of adapting to a rapidly changing environment
- IBM Business Process Manager solution identifies the following capabilities for a successful agile BPM strategy:
  - Dynamic business processes and models
  - Flexible IT infrastructure that is based on service-oriented architecture (SOA)
  - Efficient team collaboration

© Copyright IBM Corporation 2017

Figure 1-13. Business process management and IBM BPM solution

© Copyright IBM Corp. 2017

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

1.2. IBM Business Process Manager

© Copyright IBM Corp. 2017

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

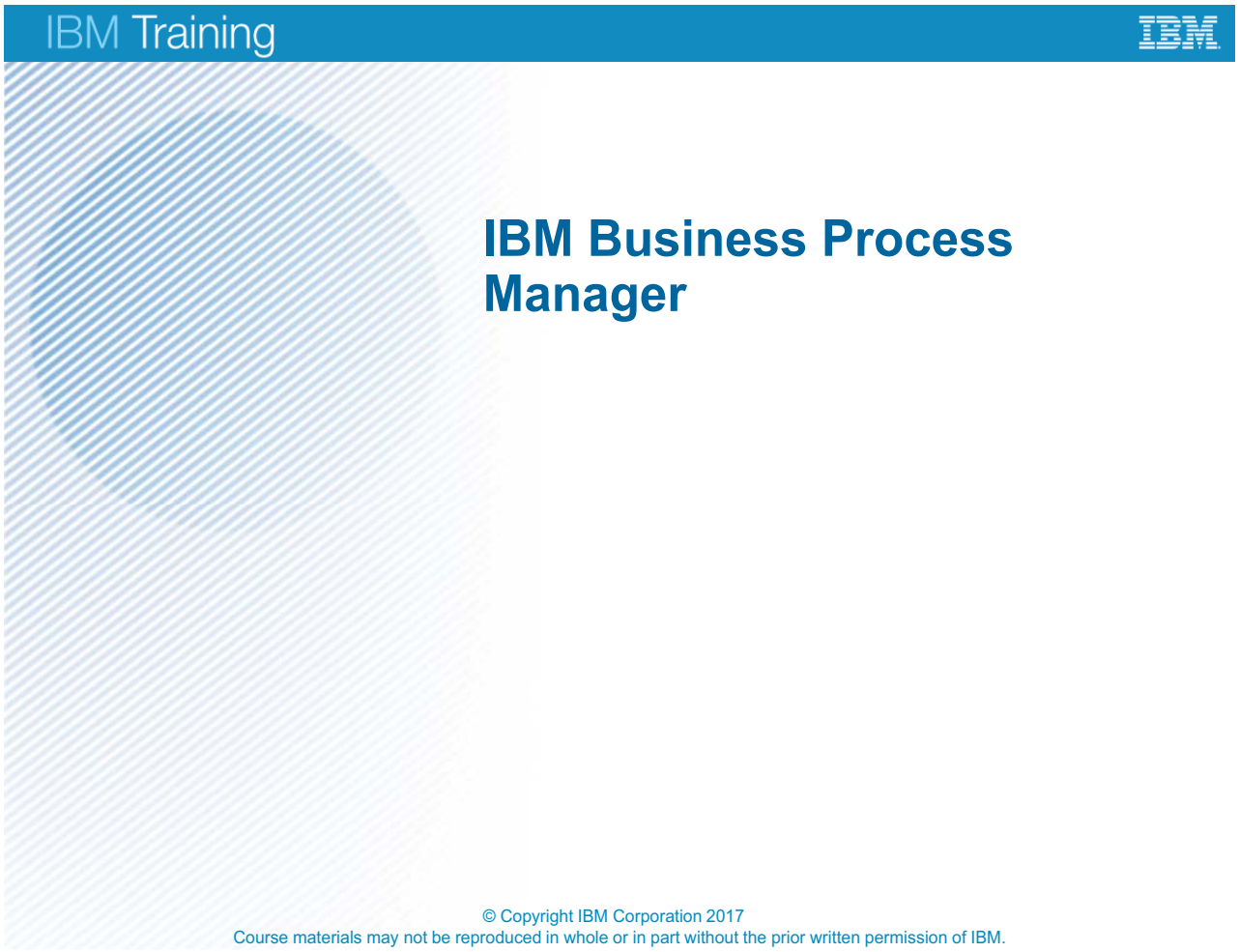


Figure 1-14. IBM Business Process Manager

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager

- IBM Business Process Manager for creating your business processes
  - Enables the development and management of business processes
  - Can be configured to support various levels of complexity and integration between IBM BPM components
  - An integrated runtime for all business processes, services, and enterprise applications
  - Tools for developers, administrators, and users
- Components of IBM Business Process Manager
  - IBM Process Server: The runtime operating system
  - IBM Process Center: A unified BPM asset repository
  - IBM Integration Designer (available in Advanced edition only): An authoring environment for developing services and self-contained enterprise applications
  - IBM Process Designer: An authoring environment for developing process models

Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-15. IBM Business Process Manager

IBM Business Process Manager gives you the capability to model and run your business processes. With IBM Business Process Manager, you get to choose the types of models and services you want to create, and the type of development environment you want to use. For example, you decide whether you want to model the business processes by using Business Process Model and Notation (BPMN) or Business Process Execution Language (BPEL). The integrated runtime supports both BPMN and BPEL (Advanced edition of the product only) for business processes, and supports services and enterprise applications.

The components of IBM Business Process Manager include:

- **IBM Process Server**: The runtime platform
- **IBM Process Center**: A unified IBM BPM asset repository
- **IBM Integration Designer** (Advanced edition of the product only): An authoring environment for developing services and self-contained enterprise applications
- **IBM Process Designer**: An authoring environment for developing process models

© Copyright IBM Corp. 2017

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

IBM Training

Business integration roles in IBM BPM development

User role	Role definition
Integration developer	• Some basic programming experience (loops, conditionals, and string manipulation) • Expects tools to simplify and abstract advanced IT implementation details
Application developer (IT developer)	• Focus is on the development of application-specific business logic (for example, EJBs, POJOs, and COBOL) for components and services that a business integration solution uses
Administrator	• Focus is on administration, management, and maintenance of a deployed business solution

© Copyright IBM Corporation 2017

Figure 1-16. Business integration roles in IBM BPM development

The IBM process management tools span the development cycle, allowing increased productivity and communication between different user roles. The definitions of the business integration user roles do not assume a one-to-one relationship. A single person can do multiple roles. For example, an application developer can also take the role of integration developer. It is also common for user roles to be involved in several phases of the application development cycle.

© Copyright IBM Corp. 2017

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

IBM Training

Three editions of IBM Business Process Manager

- IBM BPM Express
  - Limited number of users
- IBM BPM Standard
  - A comprehensive IBM BPM platform
  - Can be used for typical business process management solutions
- IBM BPM Advanced
  - Includes all the features of the IBM BPM Standard version and more advanced integration and connectivity capabilities
- IBM Business Process Manager on Cloud is now available
  - Provides the same IBM BPM capabilities on the cloud

Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-17. Three editions of IBM Business Process Manager

IBM Business Process Manager Standard is used for typical business process management projects that require a deeper business user engagement and IT collaboration through the process improvement lifecycle. IBM Business Process Manager Standard is ideal for multi-project improvement programs that focus on workflow and productivity, scaling easily from initial project to enterprise-wide programs.

BPM on Cloud is discussed in more detail later in this unit. BPM on Cloud is covered in more detail in the appendix at the end of the student book.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager Standard (1 of 2)

- IBM Process Designer
- IBM Process Center
  - Able to version BPMN processes
  - Clustering supported
- IBM Process Server
  - Run BPMN processes
  - Run monitoring support
  - No BPEL, SCA, or ESB support
- Includes basic system integration support
- Focus on improved workflow and productivity

© Copyright IBM Corporation 2017

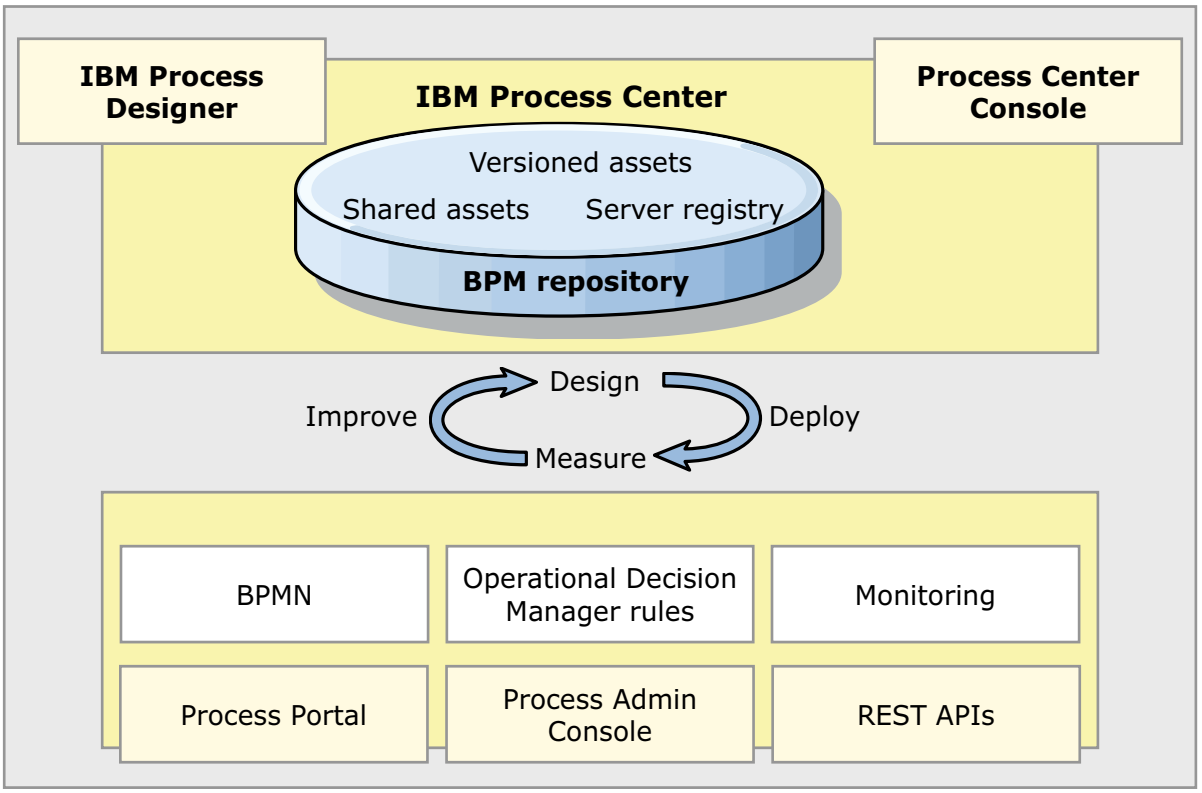
Figure 1-18. IBM Business Process Manager Standard (1 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager Standard (2 of 2)



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-19. IBM Business Process Manager Standard (2 of 2)

IBM Business Process Manager Standard provides new levels of interoperability between IBM software products.

IBM Business Process Manager Standard uses a shared model for development artifacts that are authored in the IBM Process Designer. It has one common repository and a single representation of the solution. The IBM BPM component that is called the IBM Process Center realizes this shared model. The IBM Process Center repository is implemented as tables within a database (commonly DB2).

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager Advanced (1 of 2)

- Includes all the tools, features, and capabilities of IBM BPM Standard

And

- IBM Integration Designer
  - An authoring environment that is used for creating processes, including BPEL with human tasks, SCA components, and services
- More capabilities
  - Built in ESB, transaction support, integration adapters, and case management

© Copyright IBM Corporation 2017

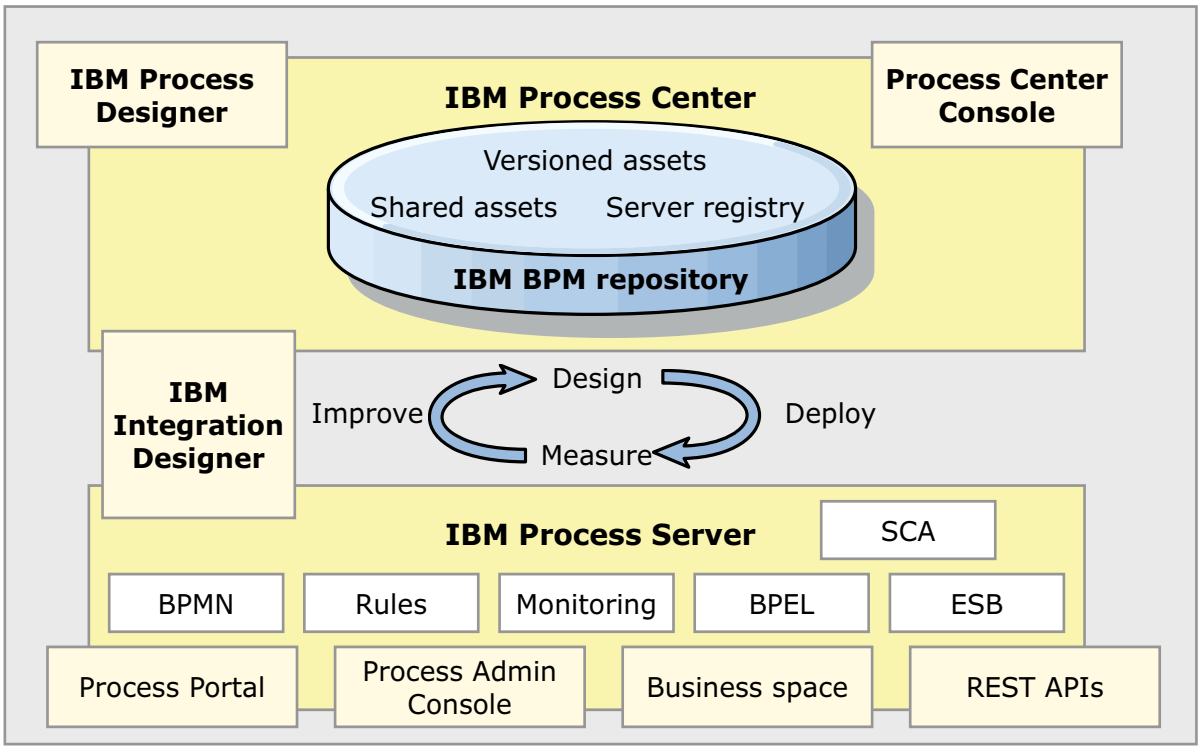
Figure 1-20. IBM Business Process Manager Advanced (1 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager Advanced (2 of 2)



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-21. IBM Business Process Manager Advanced (2 of 2)

IBM Business Process Manager Advanced provides new levels of interoperability between IBM software. It supports high-volume automation and extensive system integration.

IBM Business Process Manager Advanced uses a shared model for development artifacts that are authored in either IBM Process Designer or IBM Integration Designer. It has one common repository and a single representation of the solution.

The IBM BPM component that is called the IBM Process Center realizes this shared model. The IBM Process Center repository is implemented as tables within a database (commonly DB2).

© Copyright IBM Corp. 2017

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

IBM Training

Major tools in IBM Business Process Manager Advanced

- IBM Process Designer: An authoring environment that is used for creating process models in BPMN and BPD formats
- IBM Process Center: Includes a repository for all processes, services, and other assets
- IBM Process Server: Includes a runtime environment for supporting process models and services
- IBM Integration Designer: An authoring environment that is used for creating processes, including BPEL with human tasks, SCA components, and services
- Process Portal: Business process users interact with this tool to complete tasks and processes

 © Copyright IBM Corporation 2017

Figure 1-22. Major tools in IBM Business Process Manager Advanced

IBM Process Designer is an authoring environment that is used to create process models that contain automated and human tasks that are developed with the Business Process Model and Notation (BPMN) and Business Process Definition (BPD) formats.

IBM Integration Designer is an authoring environment that is used to create process models and advanced service implementations, including mediations, Service Component Architecture (SCA) modules, business rules, and Business Process Execution Language (BPEL) with human tasks.

IBM Process Center includes a repository for all processes, services, and other assets that are created in the authoring environments. IBM Process Server provides a single runtime environment for supporting process models, service orchestration, and integration capabilities.

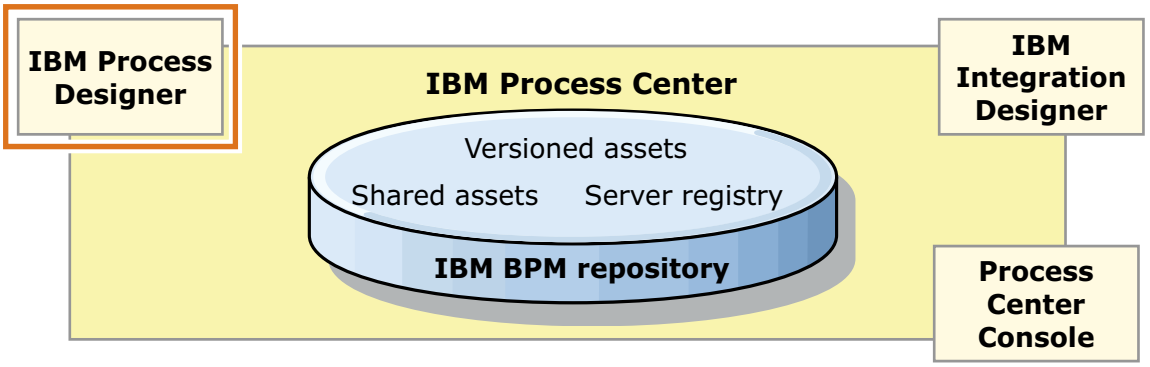
© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Designer

- Tool to develop and manage business processes
- Model, simulate, and inspect business processes
- Uses Business Process Model and Notation (BPMN) and business process definitions (BPD)



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-23. IBM Process Designer

IBM Process Designer is an Eclipse-based tool that business process authors use. It offers capabilities to model and implement business processes as process applications. IBM Process Designer includes tools, the Process Inspector, and the Process Optimizer, for interacting with processes on the Process Center Server (playback server) or a Process Server deployment target.

A process is the major unit of logic in IBM Business Process Manager. It is the container for all components of a process definition, including services, activities, and gateways; timer, message, and exception events; and sequence lines, rules, and variables. When you model a process, you create a reusable business process definition (BPD).

All Process Designer projects are contained in process applications. You store those process applications and associated artifacts in the Process Center repository.

Toolkits are containers that store library items (for example, BPDs) for reuse by process applications or other toolkits. Process applications can share library items from one or more toolkits, and toolkits can share library items from other toolkits.

Process applications that are developed in Process Designer can run any time on the Process Center server or can be saved to a snapshot and deployed on the Process Server. The same is true of services that are developed in Integration Designer and associated with process applications.

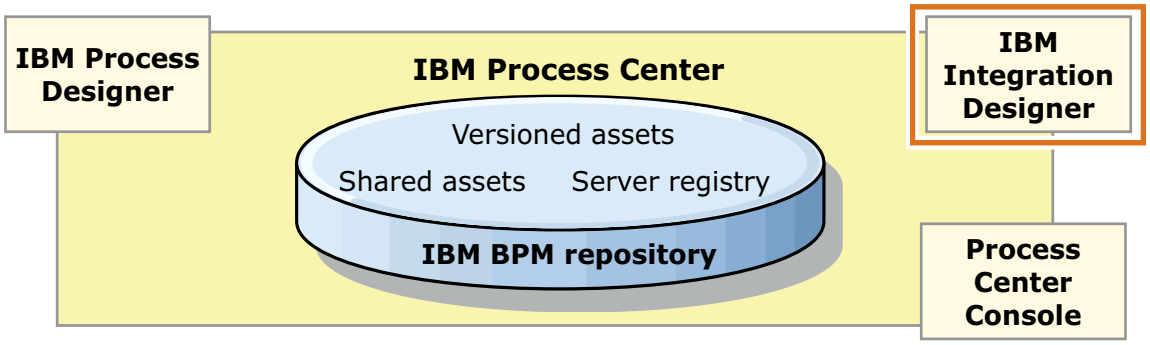
© Copyright IBM Corp. 2017

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

IBM Training

IBM Integration Designer

- **Available with Advanced edition of the product only**
- Development tool for building SCA-based integration applications
- Provides a visual development environment for developing, assembling, testing, deploying, and managing integration modules and mediation modules
- Uses Service Component Architecture (SCA) modules and libraries, and Advanced Integration services (AIS)



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-24. IBM Integration Designer

IBM Integration Designer is an Eclipse-based tool that IT developers use. IBM Integration Designer is used to author complex integrations and fully automated processes that support process applications that are designed in the Process Designer.

It incorporates a fully integrated testing environment with test cases and test suites. Using IBM Integration Designer, IT developers build reusable SOA services, orchestrate those services, and access traditional systems.

The artifacts that are produced in IBM Integration Designer include:

- SCA components that contain one or more modules and libraries and are deployable as EAR files
- Advanced Integration services that are traditional BPEL processes

The SCA modules and libraries that are created with IBM Integration Designer can be associated with a process application by using the Process Center.

The artifacts that are produced in Integration Designer can be used as services by processes that are created in Process Designer. In such cases, they are deployed with the process application.

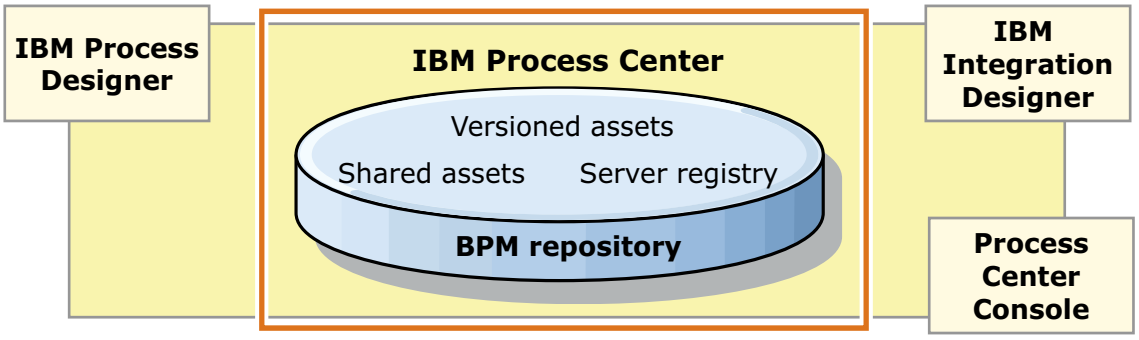
© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Center

- Repository for all Business Process Manager assets
- Lifecycle management and deployment of all applications
- Includes execution environment for development and testing
- Accessible from IBM Process Designer and from IBM Integration Designer
- Web interface that uses IBM Process Center Console
- Includes Process Center server and the Performance Data Warehouse server



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-25. IBM Process Center

You can use the IBM Process Center repository to share business process management assets with other users who are developing process applications and toolkits. These assets include process applications, reusable toolkits, monitor models, and other assets. It also can manage dependencies, versions, and deployment to servers.

The repository also maintains a registry of the process servers in the environment. The Process Center is involved in the modeling and development of process applications, running the applications during initial testing, and deploying the applications to test and production servers.

The IBM Process Center includes two servers, the Process Center server and the Performance Data Warehouse server. These servers allow developers who are working in Process Designer to run their process applications and store performance data for testing and playback during development efforts. Performance Data Warehouse retrieves tracked data from Process Server or Process Center server at regular intervals.

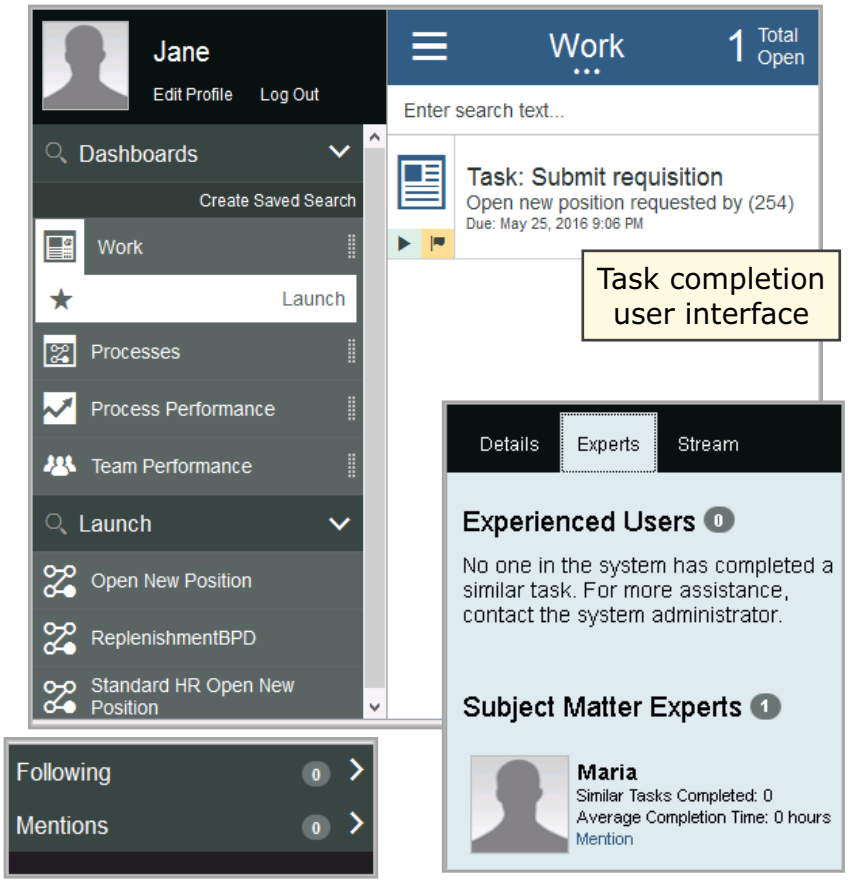
© Copyright IBM Corp. 2017

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

IBM Training

Process Portal

- User interface for participating in processes
- Social features for collaboration
  - Following
  - Mentions
  - Experts
- Streams
  - A conversation around a business process instance
- Create ad hoc reports



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-26. Process Portal

Process Portal provides an interface for you to work on your assigned tasks and collaborate with others to complete your work efficiently. In addition, if you have the associated permission, you can use dashboards to view the performance of individuals, teams, and processes. Process Portal is available with all editions of IBM Business Process Manager for participating in processes.

New tasks are automatically added to your list of tasks. If you set your user preferences to receive notifications when you are assigned new tasks, your list of tasks is automatically refreshed. If you are working on a task, the refresh is delayed until you finish your work.

When you open a task that is assigned to a group, you automatically become the owner of the task. By default, you receive a message that confirms that the task is assigned to you. You can turn off these messages from the confirmation window or by changing your user preferences. If you open a task by mistake, you can reassign it back to the designated group.

Using Process Portal, you can start processes, post comments about the process and its tasks, follow the work and comments on a process, view the audit data, and determine and modify the process path.

- **Collaborating with other users to complete a task**: While working on a task in Process Portal, you might decide that the most effective way to complete the work is to have someone work with you directly on the task. You can send a request to one or more users for them to collaborate with you.

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

You can receive notifications directly in Process Portal when an action occurs that affects you and your work. For example, a notification is received when a new task is assigned to you or your user group, or you are mentioned in a post on the process. You can use settings in your Process Portal preferences to control which notifications you see.

You can receive a notification in the following situations:

- A new task is assigned to you or your user group.
- You are mentioned in a post on a process that you are participating in.
- You are invited to collaborate on a task.

Regardless of your preferences settings, the **@Mentions** tab shows all the posts on the process that mention you and all your collaboration invitations.

You can use the **Stream** tab to see at a glance the updates to work and the posts that are specific processes. You can subscribe to the stream for specific processes by following the process.

- **Getting help from experts to complete a task**: While working on a task in Process Portal, you can see the experts who are defined for it. An expert is a team member with related task expertise who can help you with completing a task. You can get help from an expert in one of the following ways:
  - Ask an expert a question. Click **@Mentions** and compose your message. Your post is displayed in the list of posts in the **Stream** tab.
  - Work with an expert directly on your task by clicking **Collaborate**. A collaboration invitation is sent to the expert.
  - Start an instant message session with the expert. If IBM Sametime Connect is configured on your system and the expert is online, click **Chat**.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Server (1 of 2)

- IBM Process Server enables simple and flexible execution of standards-based business process management solutions
  - Implements a WS-BPEL compliant process engine that runs complex business process automation securely, consistently, and with transactional integrity by using advanced human workflow, business rules, system-to-system, and B2B capabilities
  - Enables building of composite integration applications
  - Provides high performance and quality of service with fault tolerance and error-detection capability
  - Includes a prebuilt Web 2.0 business user client (Business Space)
- IBM Process Server provides business value
  - Completes the lifecycle of business process management as the runtime engine for deployed applications
  - Helps maximize reuse of existing IT assets
  - Complete set of IBM Process Server capabilities is included only in the Advanced edition of the product
- Built on top of WebSphere Application Server
  - WebSphere Application Server offers a fast, flexible, and secure Java application server runtime environment

 © Copyright IBM Corporation 2017

Figure 1-27. IBM Process Server (1 of 2)

IBM Process Server is the runtime environment that makes SOA come alive. You can think of it as the platform or infrastructure that makes assembling an SOA solution possible. IBM Process Server provides the capabilities necessary to link together services in a meaningful way.

As installed, IBM Process Server does not provide you a turnkey system any more than buying a Java EE environment would. IBM Process Server is:

- Middleware
- An environment
- A group of functions and capabilities that make your life dramatically easier when your challenge or role is the development of SOA-based solutions

Throughout the remainder of this course, as the capabilities of IBM Process Server are illustrated, ask yourself: “How can this task be achieved without IBM Process Server?” In each case, the solution is easier with IBM Process Server than without. In this course, you examine a number of functions and capabilities. Some of them are applicable in every project; some might be applicable in only a few projects.

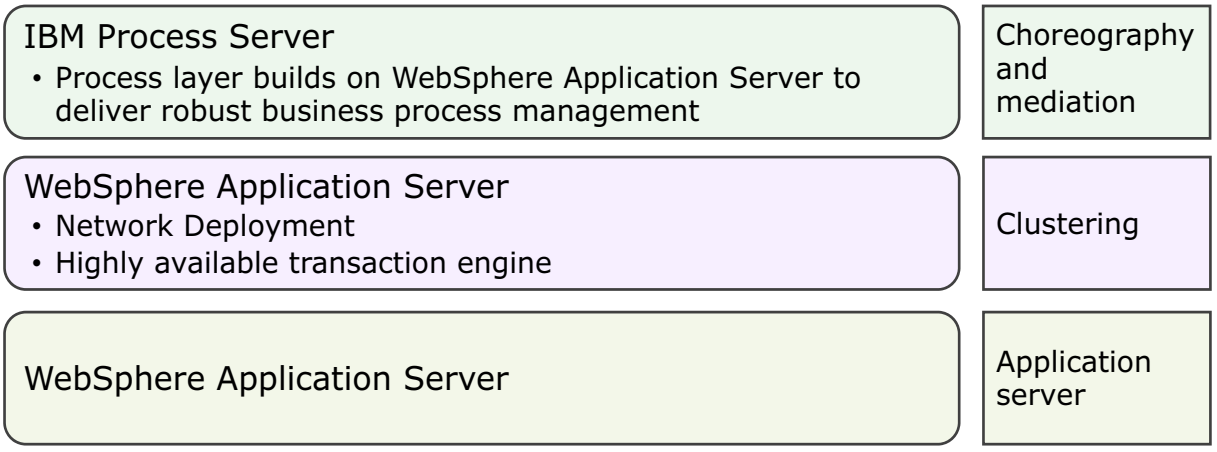
© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Server (2 of 2)

- WebSphere Application Server and Network Deployment provide high availability, workload management, and qualities of service
- IBM Process Server builds adds business process management functions and provides a communication infrastructure for integrating services, applications, and data



© Copyright IBM Corporation 2017

Figure 1-28. IBM Process Server (2 of 2)

At the lowest level, IBM Process Server is based on the WebSphere Application Server product. WebSphere Application Server implements the Java EE standard and allows applications that are written on top of it to be portable and efficient. To IBM, Java EE is a platform-neutral operating system that provides all the richness of functions that are needed to build and run applications. Users of Java EE can divest themselves from concerns of security, transaction support, resource management, and much more, and leave those functions to the Java EE environment. In this way, programmers can focus on the required business functions.

Although IBM Process Server is implemented on top of WebSphere Application Server, Java EE skills are not required for designing and implementing BPM solutions in IBM Process Server. IBM Process Server provides a higher level of abstraction, hiding its own implementation details.

Each layer encapsulates and builds on the lower layer. Everything begins with the application server. The higher abstraction layers indirectly use the application server or WebSphere Application Server Network Deployment for security, user registry, transactions, scalability, clustering, high availability, failover, platform messaging, and automated deployment.

IBM Process Server adds the capabilities of business process development, choreography, and mediations.

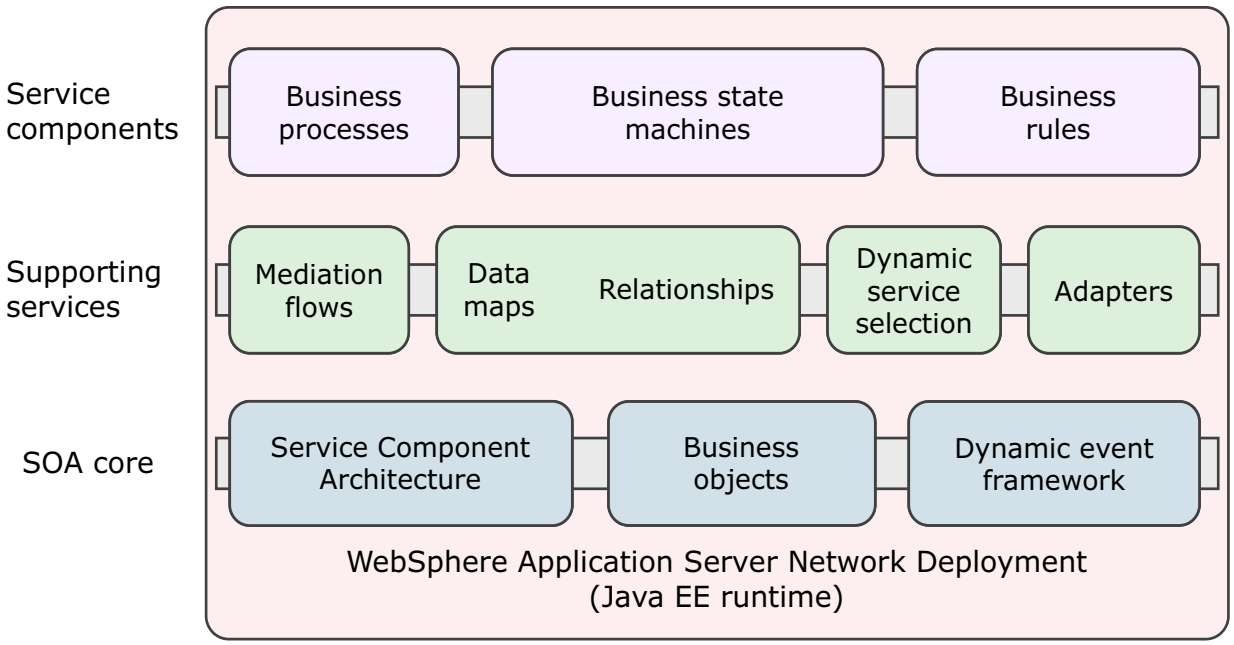
© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Server components

- IBM Process Server adds process management components to the WebSphere platform



© Copyright IBM Corporation 2017

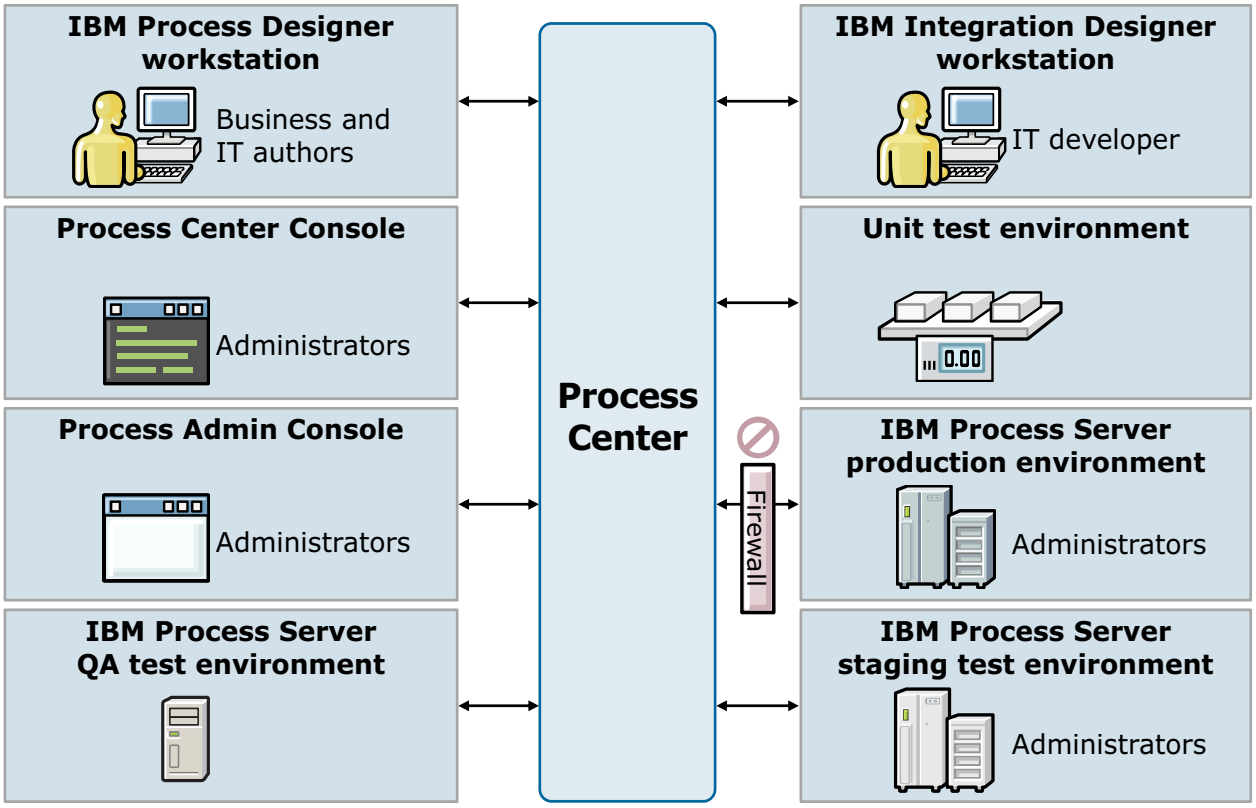
Figure 1-29. IBM Process Server components

© Copyright IBM Corp. 2017

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

IBM Training

User roles and tools in a typical deployment scenario



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-30. User roles and tools in a typical deployment scenario

The diagram displays a typical development and deployment topology.

A unit test environment with IBM Integration Designer is installed in a UTE environment. In this mode, the unit tester can test SCA modules locally in the IBM Process Server running inside the UTE or test them directly on the Process Center. Either way is acceptable, depending on the requirements.

The IBM Process Designer workstation communicates directly with the Process Center Console in the center.

It has independent QA, staging, and production environments, each with its own full stand-alone IBM Process Server connected to the central IBM Process Center. Artifacts are being published and synchronized back and forth between all the environments while the IBM Process Center is managing the central repository.

This scenario is just a sample, and the topology can be modified to suit the requirements of the organization.

© Copyright IBM Corp. 2017

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

IBM Training

Introduction to IBM BPM on Cloud

- Enterprise-grade IBM BPM 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
  - As of 2015, over 25 data centers are available worldwide
- Managed by IBM
- Artifacts that are created with IBM BPM on Cloud are compatible with IBM BPM on-premises product
- Free 30-day trial available (see Appendix A)

Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-31. Introduction to IBM BPM on Cloud

IBM Business Process Manager on Cloud is a subscription-based business process management (BPM) cloud service.

IBM Business Process Manager on Cloud provides a full lifecycle IBM BPM environment that includes development, test, and production – with tools and runtime for process design, execution, monitoring, and optimization. It is designed to enable business users to get started with process improvement quickly without the need to build and maintain an IT infrastructure.

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

IBM Training

BPM 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 2017

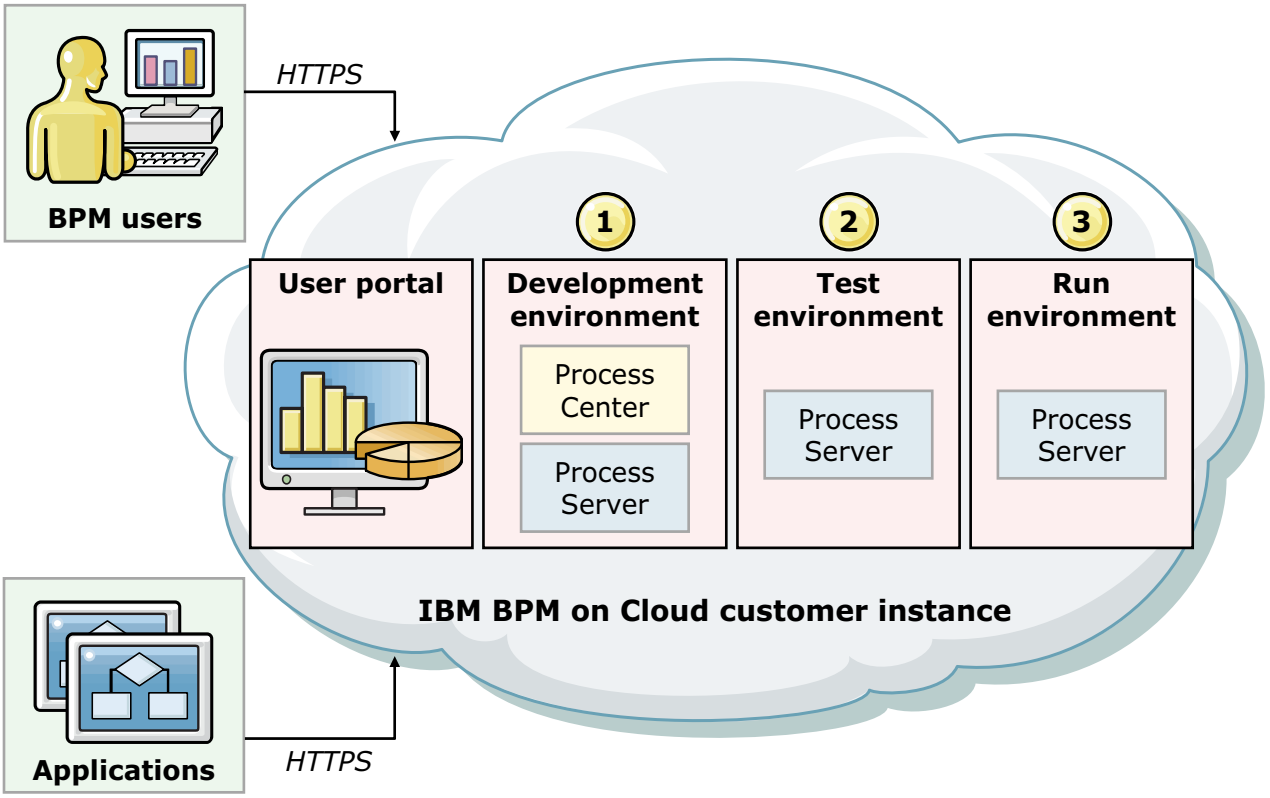
Figure 1-32. BPM on Cloud customer focus: Manage and automate decisions

© Copyright IBM Corp. 2017

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

IBM Training

IBM BPM on Cloud: Three runtime environments



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-33. IBM BPM on Cloud: Three runtime environments

IBM BPM on Cloud provides three runtime environments for decision management:

- Development
- Test
- Run

In this diagram:

- **IBM BPM 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.

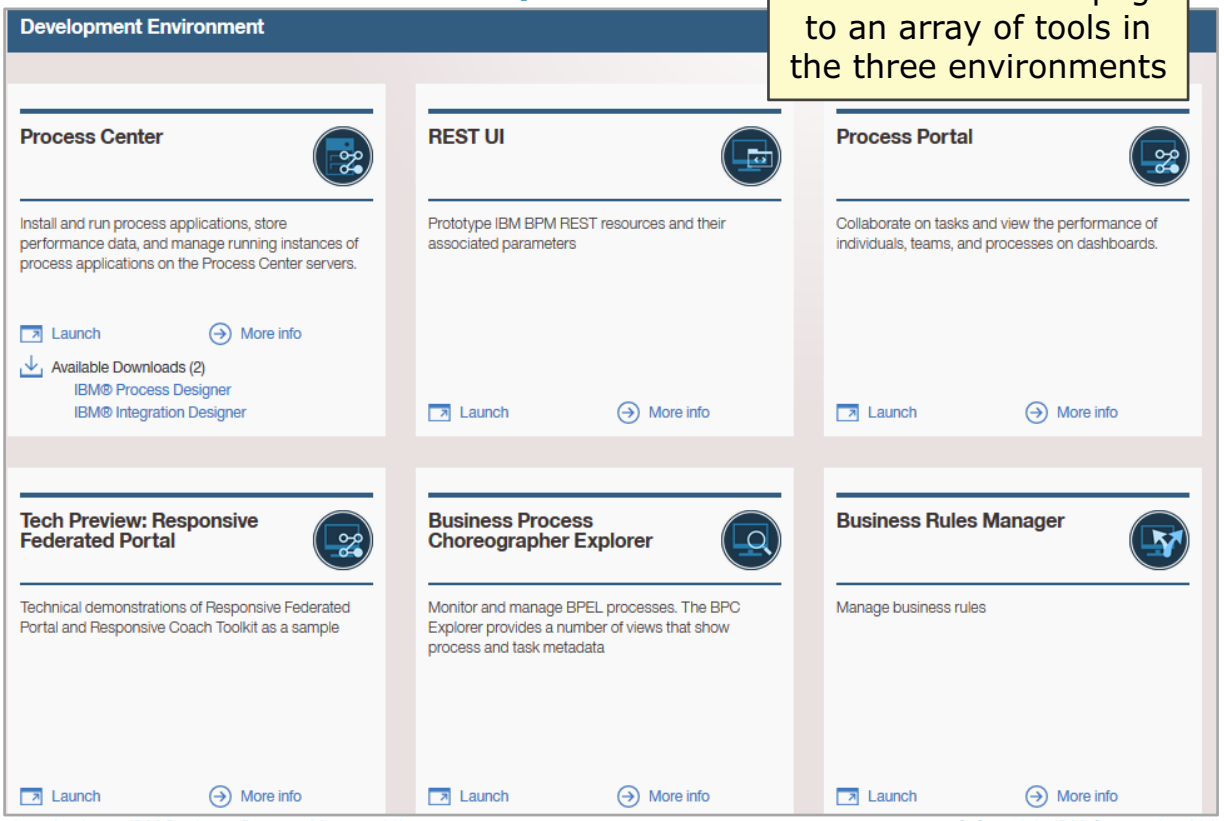
© Copyright IBM Corp. 2017

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

IBM Training

IBM BPM on Cloud user portal

Access from home page to an array of tools in the three environments



Introduction to IBM Business Process Manager V8.5.7 © Copyright IBM Corporation 2017

Figure 1-34. IBM BPM on Cloud user portal

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

IBM Training

Unit summary

- Describe the concepts of business processes and business process management (BPM)
- Describe the IBM product editions

© Copyright IBM Corporation 2017

Figure 1-35. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: Service-oriented integration is loosely coupled.
2. True or False: The integration developer role requires expertise in programming languages such as Java and C++.
3. True or False: IBM Integration Designer is included in IBM BPM Standard.

© Copyright IBM Corporation 2011, 2016

Figure 1-36. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: Service-oriented integration is loosely coupled.
   The answer is True. Service-oriented integration is loosely coupled and less brittle. Standard (WSDL) interfaces allow effective reuse and interoperability, which separates interface technology from implementation. Knowledge of the underlying implementation details is not required to develop composite applications.
2. True or False: The integration developer role requires expertise in programming languages such as Java and C++.
   The answer is False. The integration developer role requires only basic programming experience (loops, conditionals, and string manipulation).
3. True or False: IBM Integration Designer is included in IBM BPM Standard.
   The answer is False. IBM Integration Designer is not included in IBM BPM Standard. It is included only in the IBM BPM Advanced version.

© Copyright IBM Corporation 2011, 2016

Figure 1-37. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

References

- IBM Business Process Manager product documentation:
  - http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbpm.main.doc/kc-homepage-bpm.html
- BPM – Business Process Management:
  - http://www.ibm.com/software/info/bpm/
- IBM Business Process Manager home page:
  - http://www.ibm.com/software/products/us/en/business-process-manager-advanced
- IBM Business Process Manager on developerWorks:
  - http://www.ibm.com/developerworks/websphere/zones/bpm/

© Copyright IBM Corporation 2017

Figure 1-38. References

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Using iterative development to create applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 1-39. Exercise 1

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

IBM Training

Exercise objectives

- Examine the various paths for the Account Verification process in IBM Integration Designer
- Create a business process diagram in IBM Process Designer
- Build an implementation for an activity in a business process diagram
- Use the IBM Process Center Playback Server to test the business process diagram
- Replace a Java component with a business process diagram
- Use IBM Integration Designer to test the imported business process diagram

Introduction to IBM Business Process Manager V8.5.7

© Copyright IBM Corporation 2016

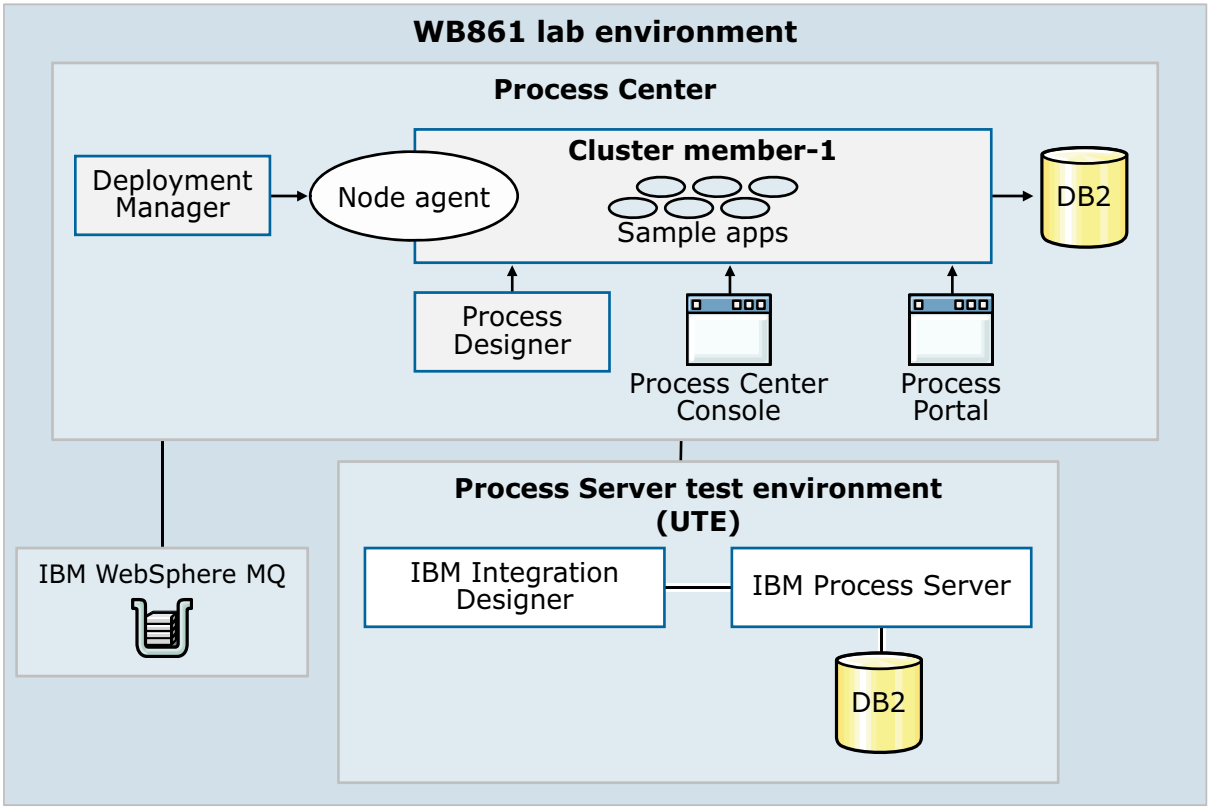
Figure 1-40. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

WB861 lab environment



© Copyright IBM Corporation 2017

Figure 1-41. WB861 lab environment

© Copyright IBM Corp. 2017

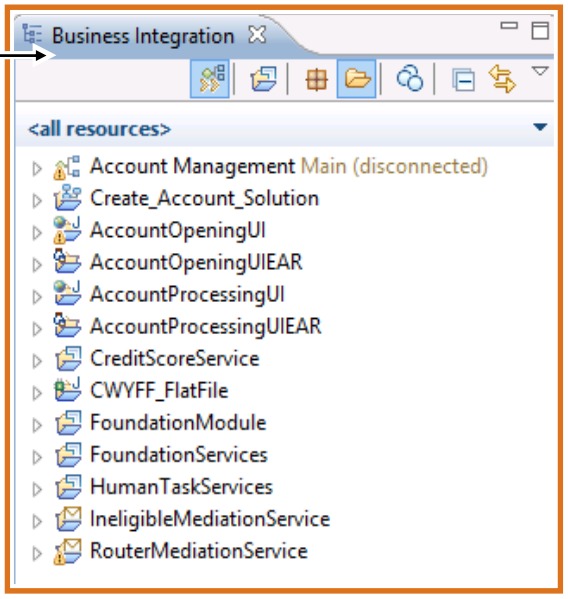
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 1 (1 of 2)

Prebuilt components that are imported in the lab:

1. **Account Management** process application
2. **AccountVerification** BPEL process



 © Copyright IBM Corporation 2017

Figure 1-42. Components that are required for Exercise 1 (1 of 2)

The Process Center includes a repository for all processes, services, and other assets that are created in IBM Process Designer and IBM Integration Designer. In this exercise, the four basic paths through the process are explored and tested:

1. WebSphere Adapter for Flat Files is used to archive the applications that are ineligible.
2. Eligible applications for customers that are deemed to be a low credit risk are automatically approved.
3. Eligible applications for customers that are deemed to be a high credit risk require supplemental documentation and manual approval or denial.
4. Eligible applications for customers that are deemed to be a medium credit risk also require supplemental documentation and manual approval or denial.

A predefined test case (with corresponding test data) is used to test each path through the process.

- An application for customer **AbcCo** is used to test the ineligible path (path 1).
- An application for customer **IBM** is used to test the low credit risk path (path 2).
- An application for customer **TestCo** is used to test the high credit risk path (path 3).
- An application for customer **ACME** is used to test the medium credit risk path (path 4).

© Copyright IBM Corp. 2017

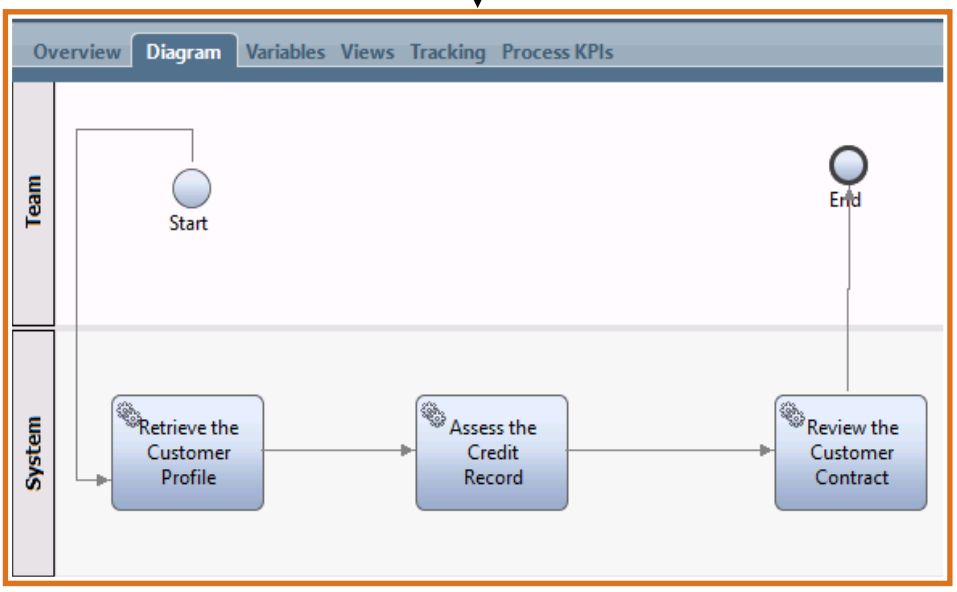
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 1 (2 of 2)

New components that you created in the lab:

1. Determine Applicant Eligibility BPD



© Copyright IBM Corporation 2017

Figure 1-43. Components that are required for Exercise 1 (2 of 2)

In the BPEL process, arbitrary Java code determines the eligibility of a customer account. The Java code is written in the **Determine Applicant Eligibility** component. This code populates the request with sample data, which depends upon the value of the customer company name.

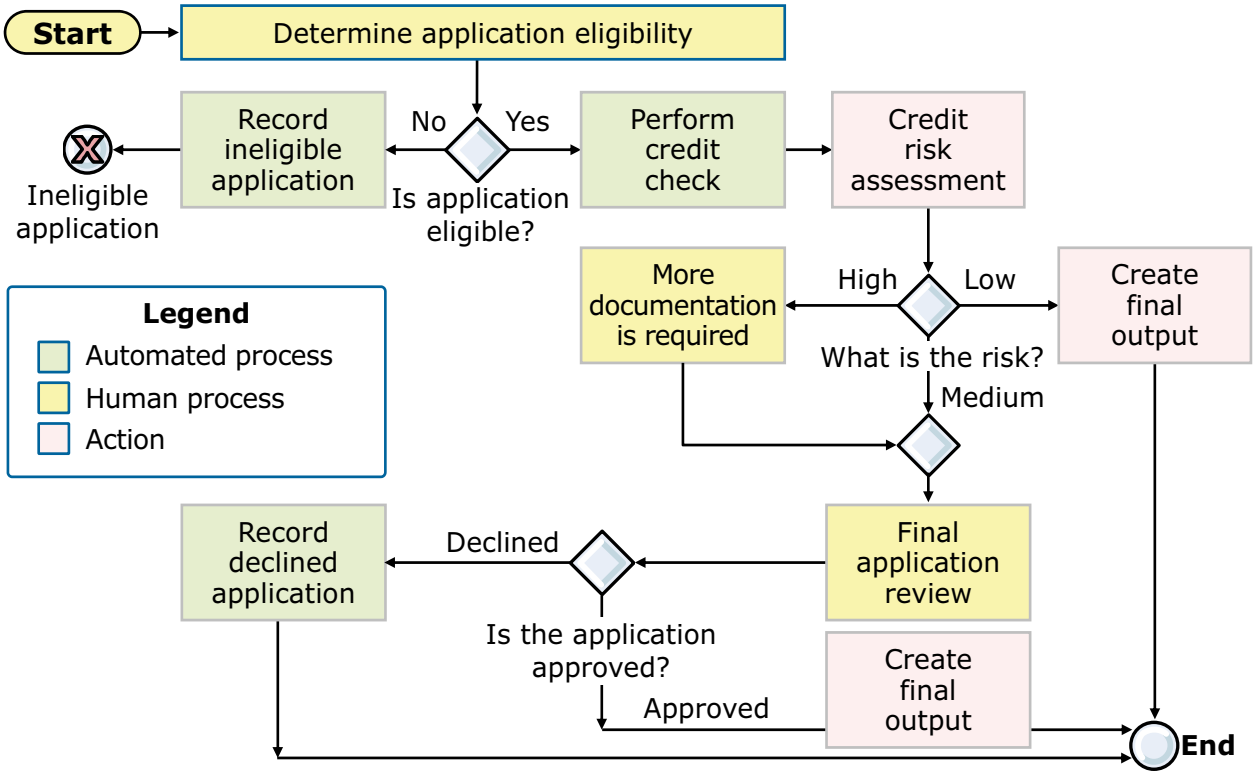
Now you replace this Java component with a business process built-in IBM Process Designer. You still use sample data, but the data is extracted from a business process diagram, not from a Java SCA component. In this section of the exercise, you create a business process to contain the DetermineApplicantEligibility logic. This process determines whether a customer application is eligible based on the company name.

© Copyright IBM Corp. 2017

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

IBM Training

Account verification process: Overview



© Copyright IBM Corporation 2017

Figure 1-44. Account verification process: Overview

You work with the Process designer tool, which is used to create process models, services, and other assets within process applications. You go through the various steps of the BPD and run the process to see the results.

The following scenario illustrates this concept.

Account Verification Process scenario

- When the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and end the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is performed against the customer's credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.
- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.
- If the customer is determined to be high risk, the customer must submit more documentation; then, the customer must seek final approval for the application.

© Copyright IBM Corp. 2017

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

- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

© Copyright IBM Corp. 2017

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

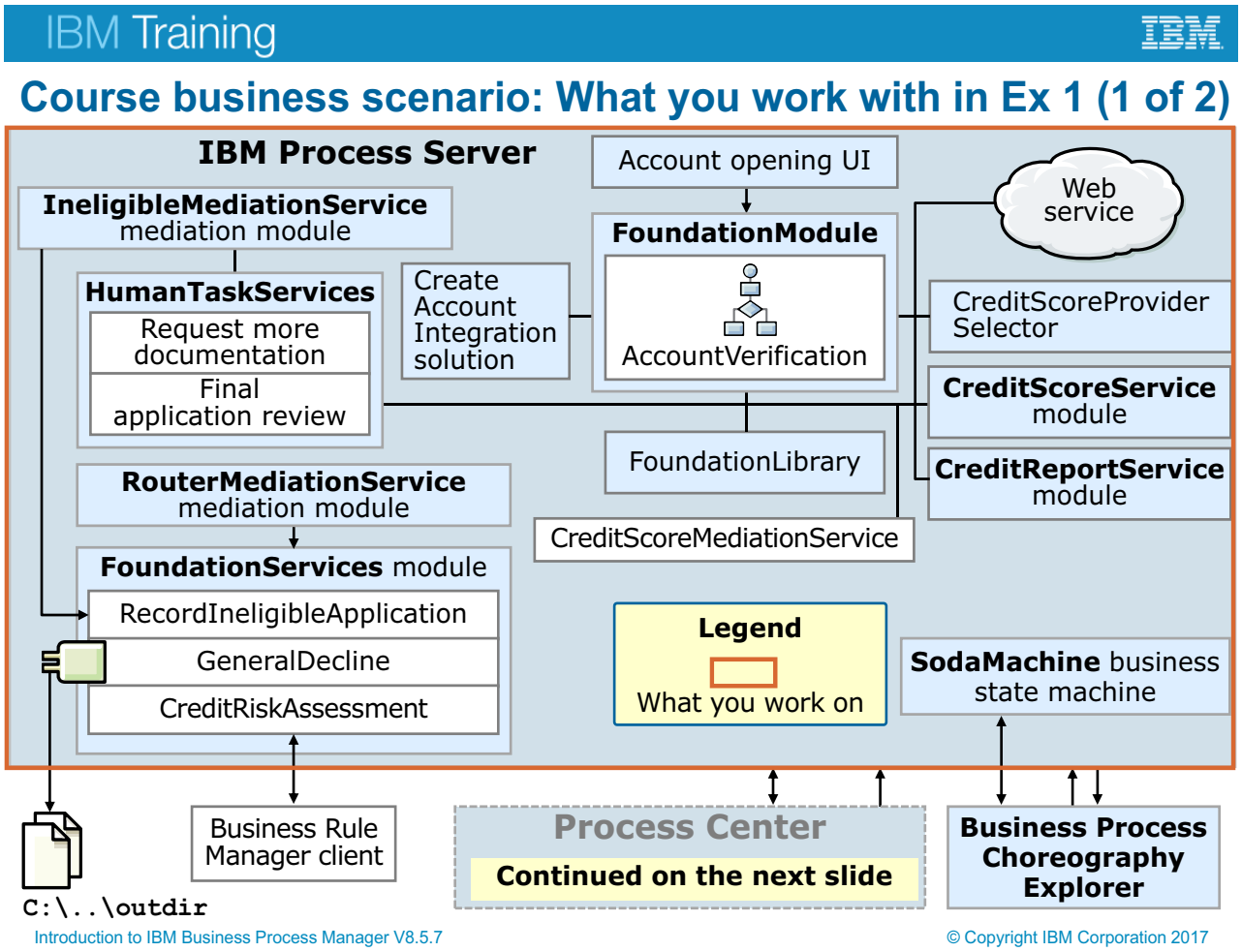


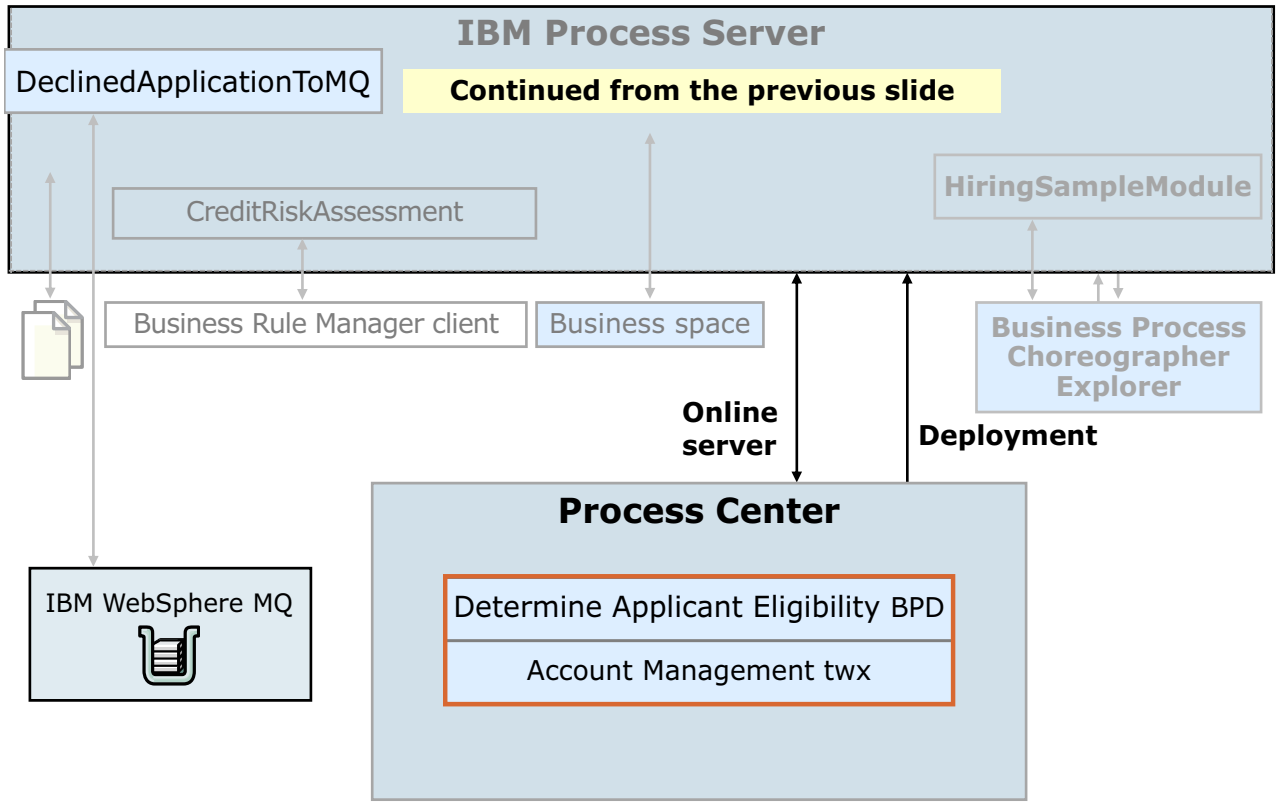
Figure 1-45. Course business scenario: What you work with in Ex 1 (1 of 2)

In this lab, you work with both the IBM Integration Designer and IBM Process Center.

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

IBM Training

Course business scenario: What you work with in Ex 1 (2 of 2)



© Copyright IBM Corporation 2017

Figure 1-46. Course business scenario: What you work with in Ex 1 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM Process Center.

© Copyright IBM Corp. 2017

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

Unit 2. Course business scenario

Estimated time

01:00

Overview

This unit describes the purpose, function, and business value of each lab exercise. It also describes the course lab scenario and the components that you use and build in the course labs.

How you will check your progress

- Checkpoint questions

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe the key components that you build and assemble in this course
- Explain the benefits of each lab exercise in this course

© Copyright IBM Corporation 2017

Figure 2-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Some important points

- This unit is an overview of what you create and work with in each exercise of this course
- It lists the key components that are already built for you at the start of each exercise
- It lists the key components that you build in each exercise
- The details of each component are not explained in this unit, but in the exercise

© Copyright IBM Corporation 2017

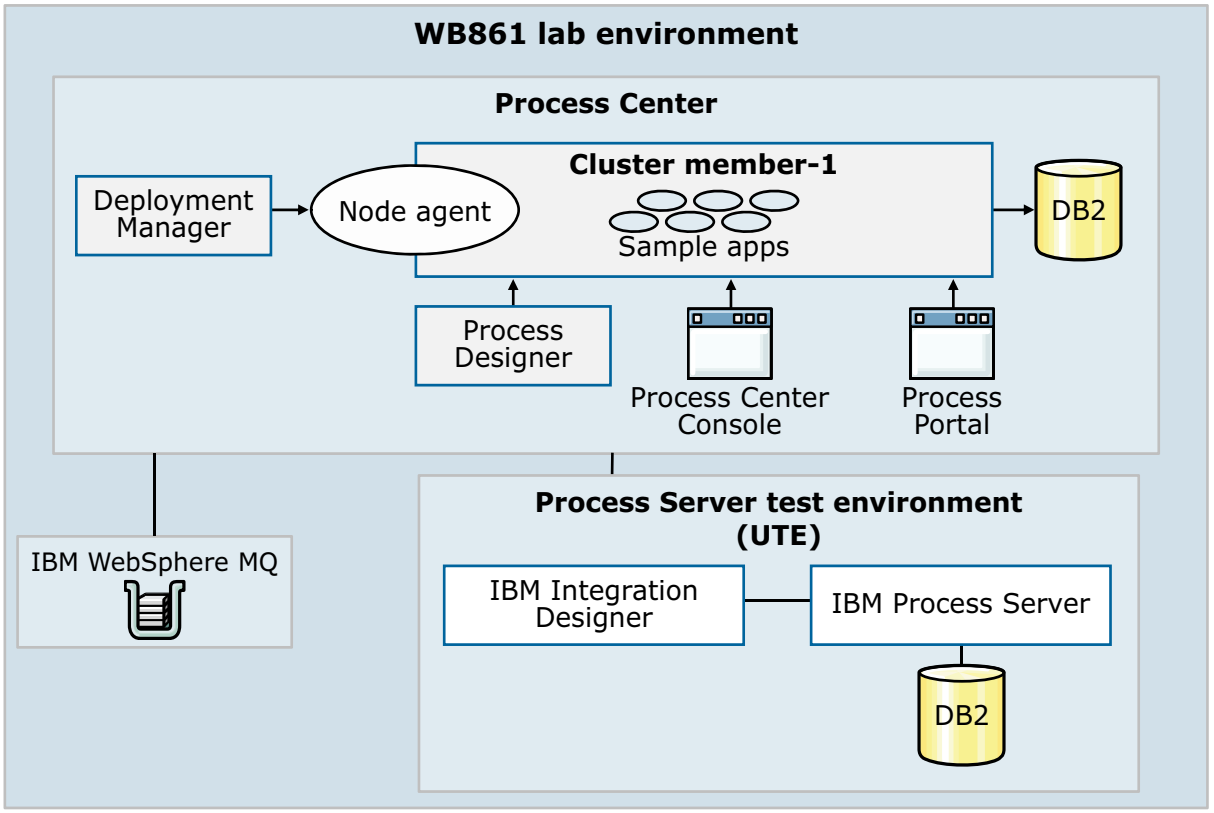
Figure 2-2. Some important points

© Copyright IBM Corp. 2017

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

IBM Training

WB861 lab environment



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-3. WB861 lab environment

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

IBM Training

Exercise 1

Using iterative development to create applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-4. Exercise 1

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

IBM Training

What are some key tasks that you did in Exercise 1?

- Tested the different paths for the Account Verification process in IBM Integration Designer
- Implemented a version of the Account Verification process in IBM Process Designer and then tested it

© Copyright IBM Corporation 2017

Figure 2-5. What are some key tasks that you did in Exercise 1?

© Copyright IBM Corp. 2017

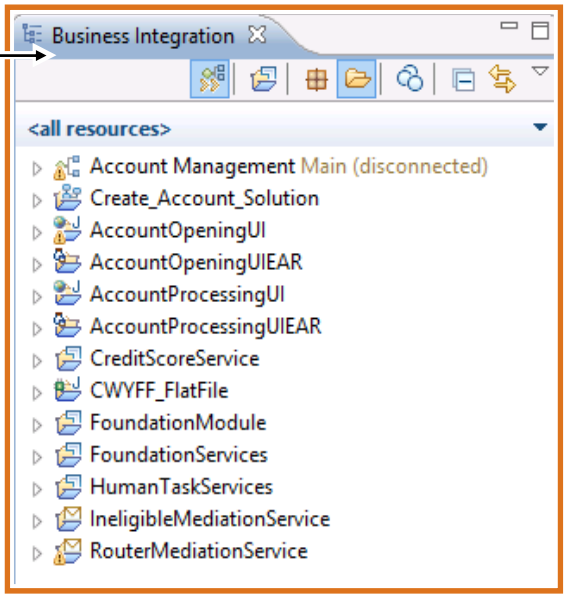
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 1 (1 of 2)

Prebuilt components that are imported in the lab:

1. **Account Management** process application
2. **AccountVerification** BPEL process



© Copyright IBM Corporation 2017

Figure 2-6. Components that are required for Exercise 1 (1 of 2)

The Process Center includes a repository for all processes, services, and other assets that are created in IBM Process Designer and IBM Integration Designer. In this exercise, The four basic paths through the process are explored and tested:

1. WebSphere Adapter for Flat Files is used to archive the applications that are ineligible.
2. Eligible applications for customers that are deemed to be a low credit risk are automatically approved.
3. Eligible applications for customers that are deemed to be a high credit risk require supplemental documentation and manual approval or denial.
4. Eligible applications for customers that are deemed to be a medium credit risk also require supplemental documentation and manual approval or denial.

A predefined test case (with corresponding test data) is used to test each path through the process.

- An application for customer **AbcCo** is used to test the ineligible path (path 1).
- An application for customer **IBM** is used to test the low credit risk path (path 2).
- An application for customer **TestCo** is used to test the high credit risk path (path 3).
- An application for customer **ACME** is used to test the medium credit risk path (path 4).

© Copyright IBM Corp. 2017

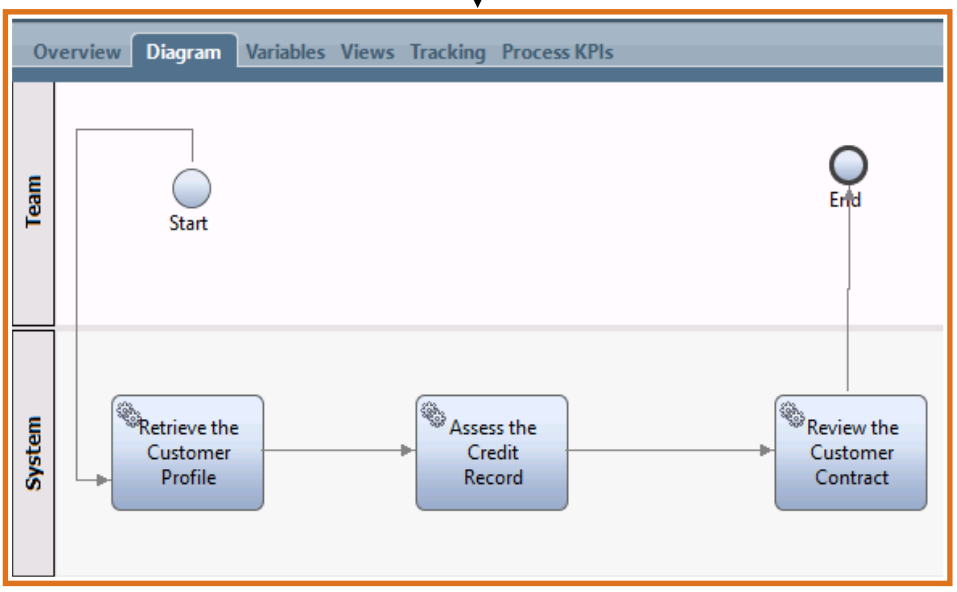
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 1 (2 of 2)

New components that you created in the lab:

1. Determine Applicant Eligibility BPD



© Copyright IBM Corporation 2017

Figure 2-7. Components that are required for Exercise 1 (2 of 2)

In the BPEL process, arbitrary Java code determined the eligibility of a customer account. The Java code is written in the **Determine Applicant Eligibility** component. This code populates the request with sample data, which depends upon the value of the customer company name.

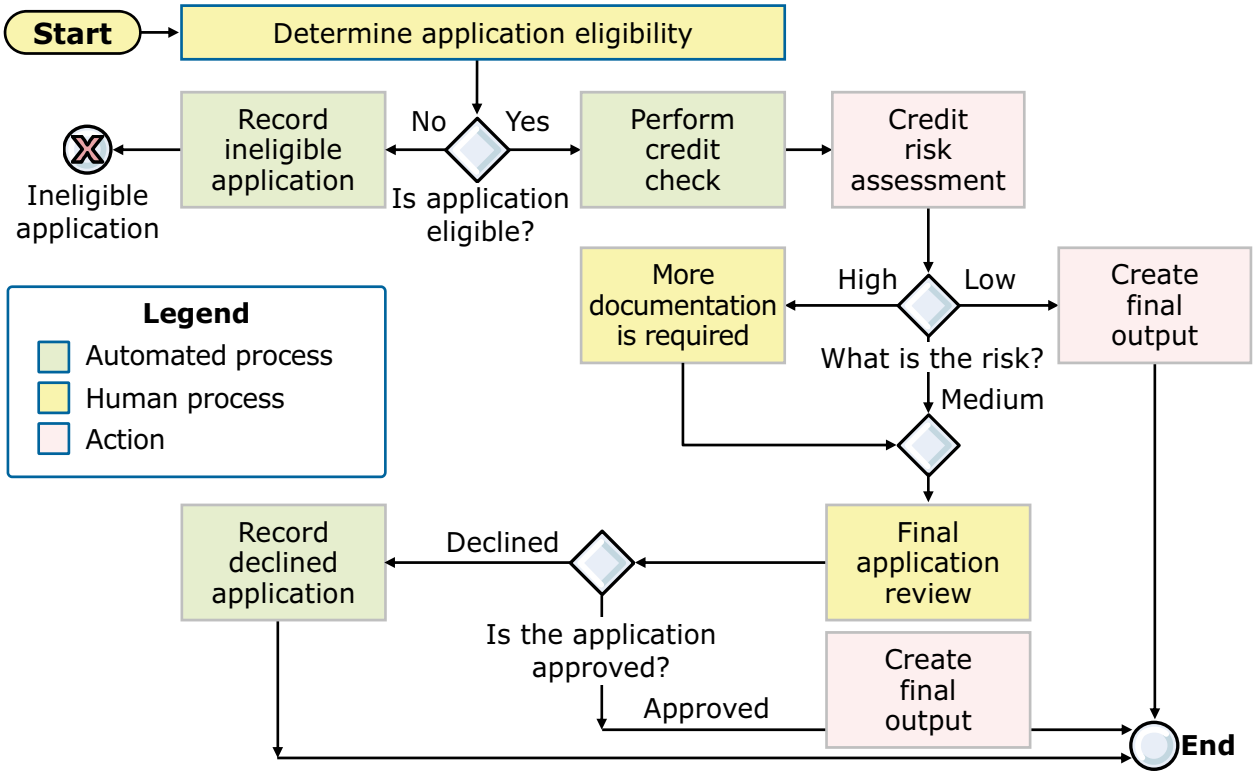
Now, you replace this Java component with a business process built-in IBM Process Designer. You still use sample data, but the data is extracted from a business process diagram, not from a Java SCA component. In this section of the exercise, you create a business process to contain the DetermineApplicantEligibility logic. This process determines whether a customer application is eligible based on the company name.

© Copyright IBM Corp. 2017

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

IBM Training

Account verification process: Overview



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-8. Account verification process: Overview

You work with the Process designer tool, which is used to create process models, services, and other assets within process applications. You go through the various steps of the BPD and run the process to see the results.

The following scenario illustrates this concept.

Account Verification Process scenario

- When the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and end the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is performed against the customer's credit check.
- If the customer is determined to be low risk, the application is automatically approved. An output message is generated for the client, and the process is complete.
- If the customer is determined to be medium risk, the customer must seek final approval from an authorized figure for the application.
- If the customer is determined to be high risk, the customer must submit more documentation; then, the customer must seek final approval for the application.

© Copyright IBM Corp. 2017

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

- If the application is approved, generate an output message for the customer. The process is complete.
- If the application is denied, record the declined application. The process is complete.

© Copyright IBM Corp. 2017

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

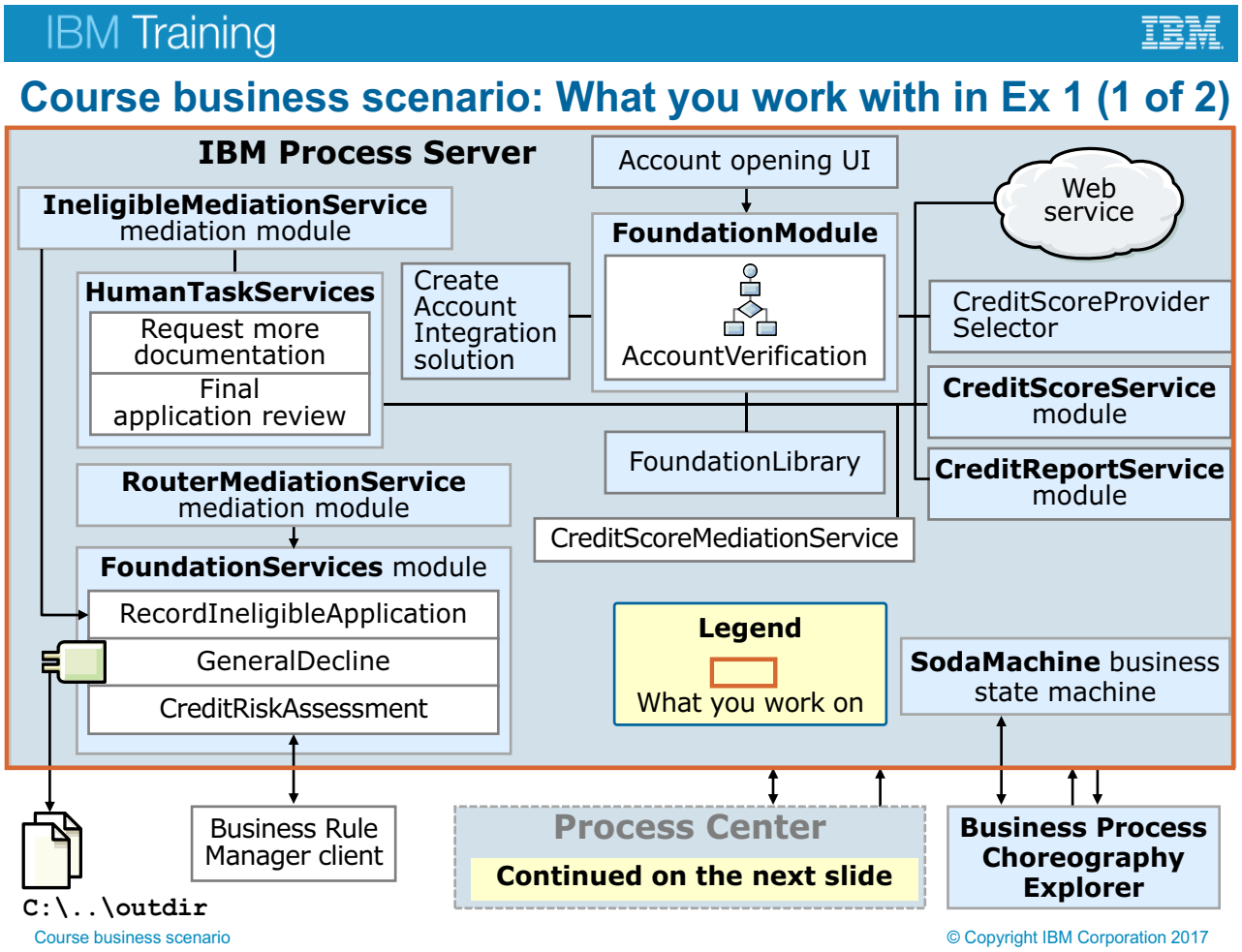


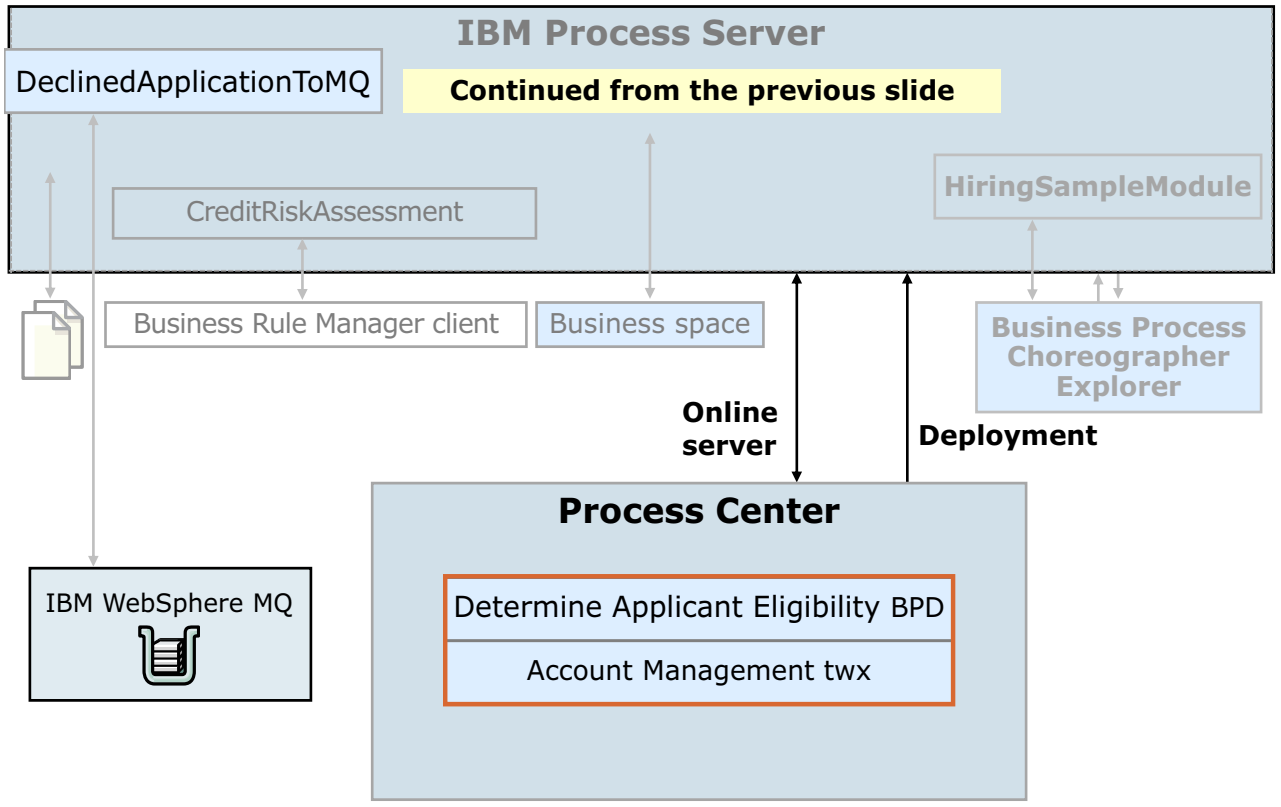
Figure 2-9. Course business scenario: What you work with in Ex 1 (1 of 2)

In this lab, you work with both the IBM Integration Designer and IBM Process Center.

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

IBM Training

Course business scenario: What you work with in Ex 1 (2 of 2)



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-10. Course business scenario: What you work with in Ex 1 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM Process Center.

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 2

Version control for SCA applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-11. Exercise 2

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

IBM Training

What are some key tasks that you do in Exercise 2?

- Create multiple versions of an Integration module and deploy the EAR file to IBM Process Server by using the service deployment tool

© Copyright IBM Corporation 2017

Figure 2-12. What are some key tasks that you do in Exercise 2?

© Copyright IBM Corp. 2017

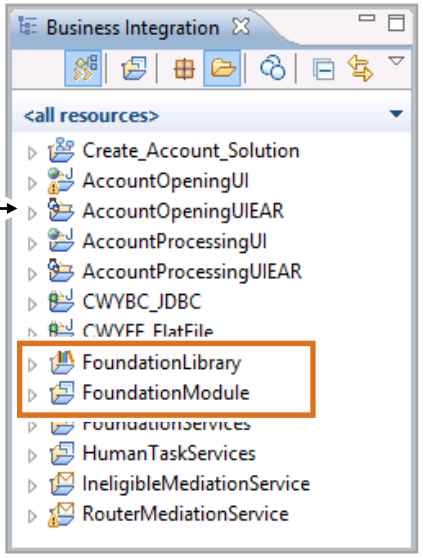
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **FoundationLibrary**



New components that you create in the lab:

1. FoundationModule_v1.0 ear
2. FoundationModule_v2.0 ear

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-13. Components that are required for Exercise 2

Version control is the ability for the runtime environment to identify snapshots in the lifecycle of a solution or service and to be able to concurrently run multiple snapshots at the same time. Version control is beneficial in a production environment for a number of reasons. Version control allows the simultaneous deployment of multiple versions of an SCA module to the same deployment environment.

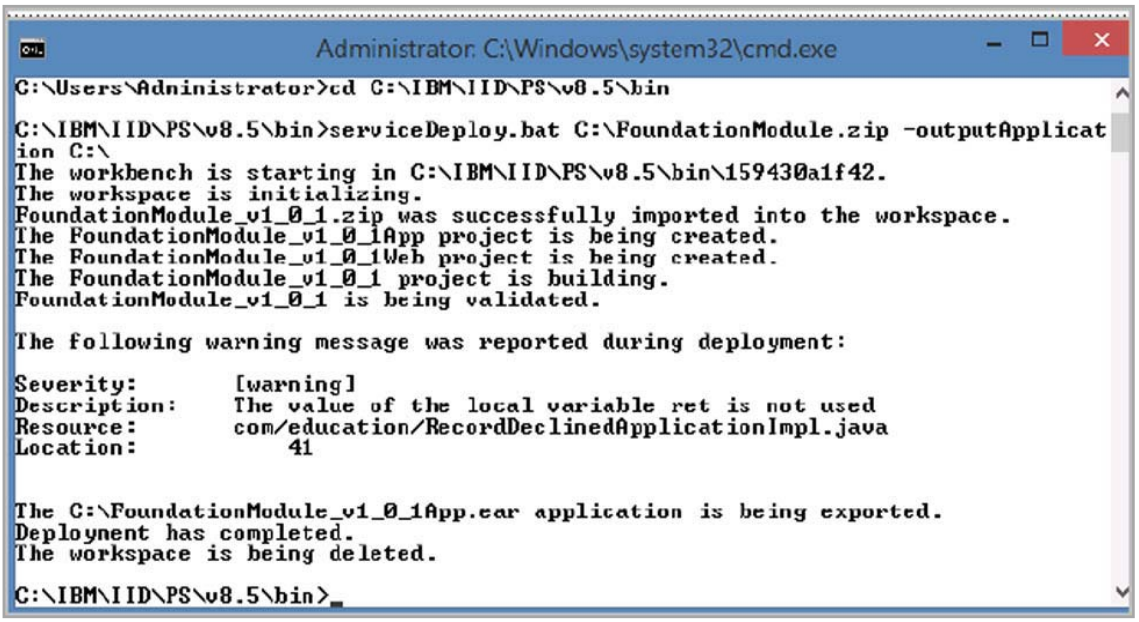
In this exercise, you create versioned modules and libraries in IBM Integration Designer for future deployment to a production environment. These versions are for use on the server at run time, not for distinguishing levels of completion in a development environment. The stored data consists of the version number (value), the version provider, and the library dependency.

© Copyright IBM Corp. 2017

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

IBM Training

Run the serviceDeploy command to create a version-aware EAR file in Exercise 2



Course business scenario © Copyright IBM Corporation 2017

Figure 2-14. Run the serviceDeploy command to create a version-aware EAR file in Exercise 2

In this section, you deploy the module to the server and verify that the version information is included. Versioned modules can be exported for command-line service deployment and deployed with the serviceDeploy command to create a version-aware EAR file. Versioned modules that are exported as EAR files are not version-aware.

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

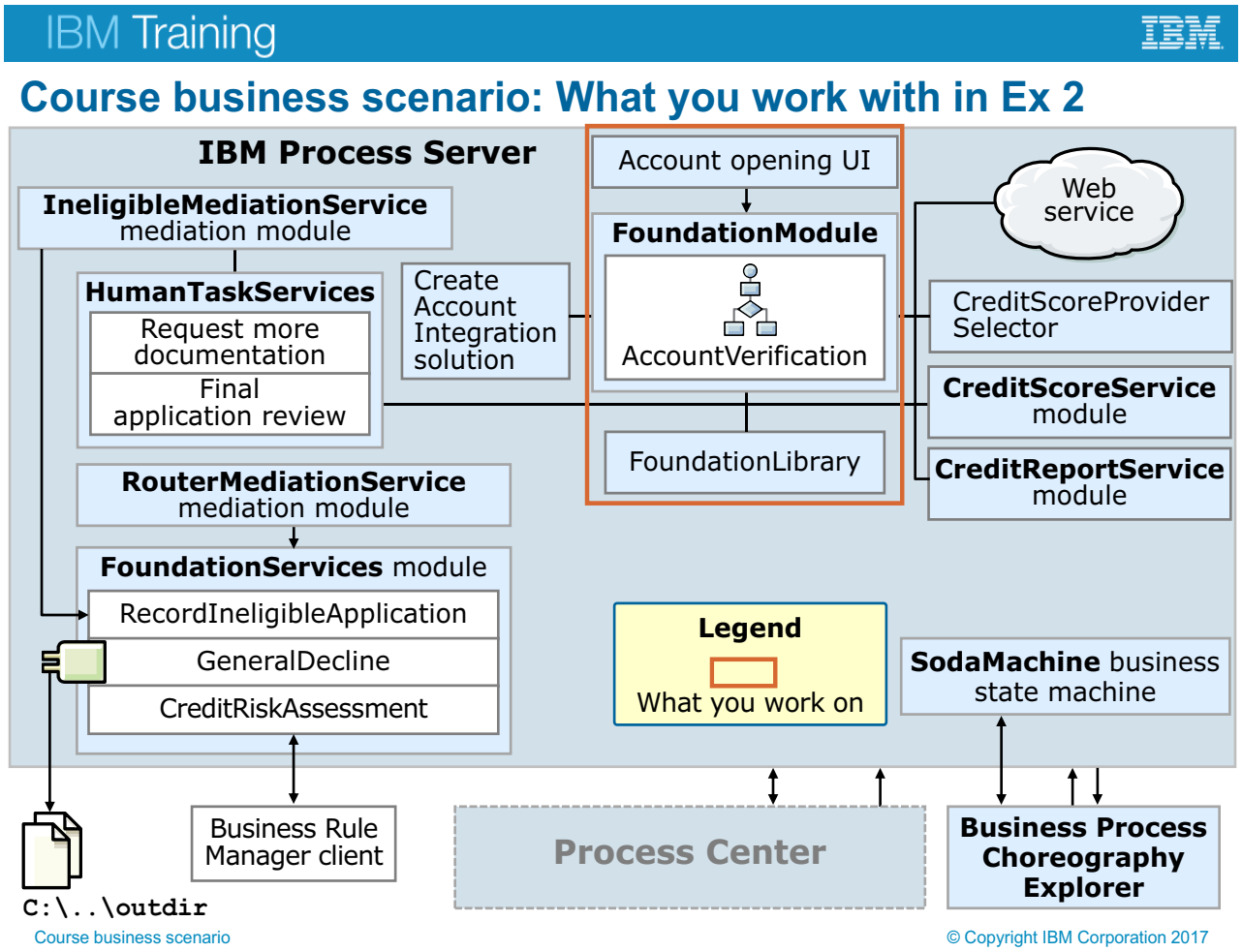


Figure 2-15. Course business scenario: What you work with in Ex 2

In this lab, you work with the IBM Process Server test environment.

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 3

Working with SCA bindings and qualifiers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-16. Exercise 3

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

IBM Training

What are some key tasks that you do in Exercise 3?

- Create and implement a WebSphere MQ import binding in IBM Integration Designer and test with a Flat File adapter
- Enable event sequencing qualifier on a Flat File adapter

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-17. What are some key tasks that you do in Exercise 3?

Messaging providers are equipped with application programming interfaces (APIs), which allow applications to use the services of the messaging provider. With IBM Integration Designer, you can use a number of messaging service providers, both commercially available and custom-written with the use of resource adapters. A resource adapter uses the APIs associated with a messaging provider to allow messages to be passed between the application and the messaging provider.

One such commercially available messaging provider is WebSphere MQ, a family of products that support high-throughput, highly reliable asynchronous messaging services across disparate applications, systems, and platforms. WebSphere MQ provides two sets of APIs for great flexibility in interacting with it from an application: the WebSphere MQ JMS interface and the queuing (MQI).

In this exercise, from an IBM Process Server application, you interact with WebSphere MQ messaging by using WebSphere MQ with the use of a resource adapter, which connects the application and WebSphere MQ through JMS messaging.

Event sequencing is one of the qualities of service (QoS) qualifiers that IBM Process Server supports. With event sequencing, you can define one or more “keys” that are used to group messages. When event sequencing is enabled, and two or more messages that have the same key value are received at an interface, those messages are prevented from being processed simultaneously.

© Copyright IBM Corp. 2017

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

It is important to note here that the event sequencing does ***not*** explicitly force a specific processing order based on the value of the sequencing keys. It also does not physically store or reorder messages based on sequencing key values. It prevents only messages that have the same key value as another from being processed at the same time.

For example, suppose that five messages are received, and their sequencing keys have the values 4, 5, 2, 4, 5. The first three messages (4, 5, 2) are eligible to be processed simultaneously, but the second “4” message cannot be processed until the first “4” message is processed. Similarly, the second “5” message cannot be processed at the same time as the first “5” message; it can be processed only after the first “5” message processing is complete.

Event sequencing can be enabled only for asynchronous calls, and only on interfaces. In this exercise, you enable event sequencing on a flat file adapter.

© Copyright IBM Corp. 2017

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

IBM Training

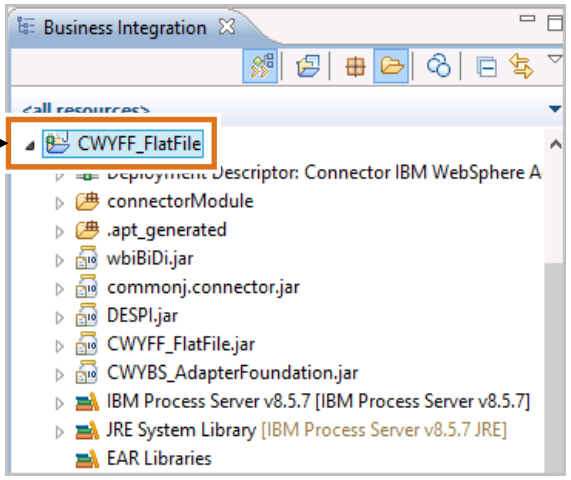
Components that are required for Exercise 3

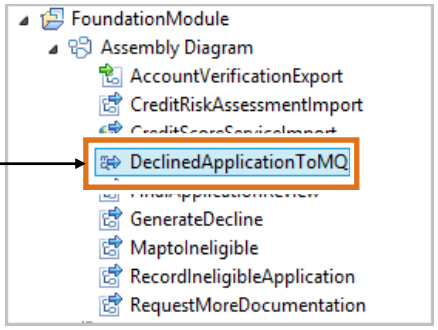
Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **AccountVerification BPEL**

New components that you create in the lab:

1. **DeclinedApplicationToMQ component**
2. **CWYFF_FlatFile adapter component**





© Copyright IBM Corporation 2017

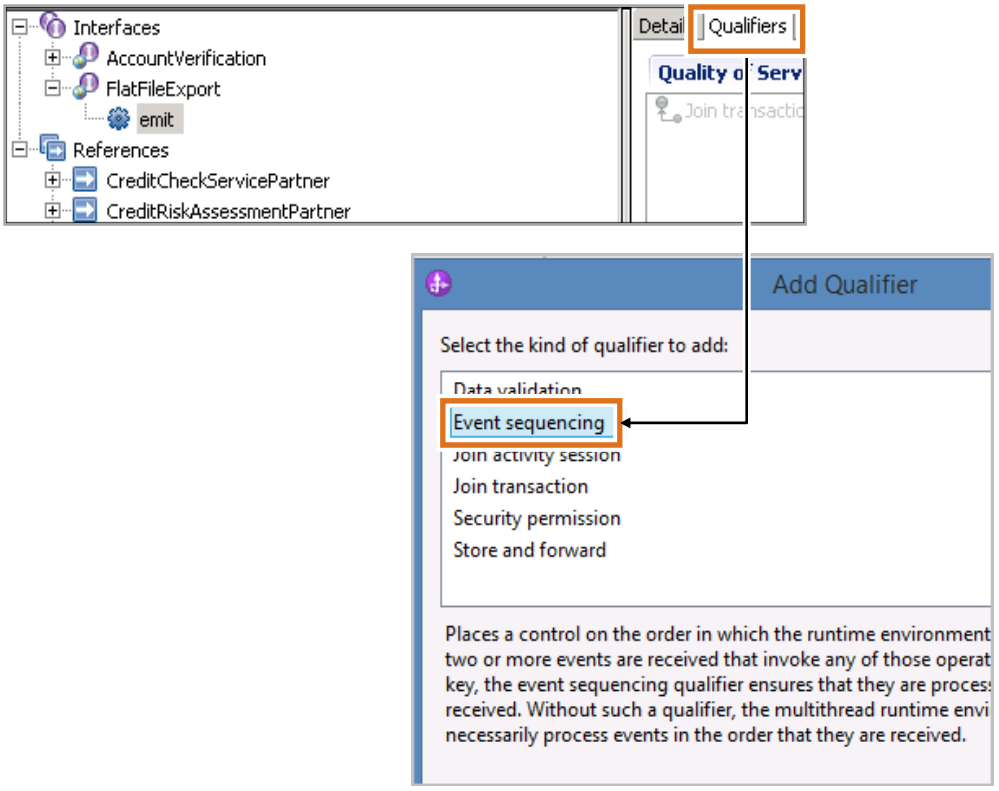
Figure 2-18. Components that are required for Exercise 3

© Copyright IBM Corp. 2017

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

IBM Training

Enable the event sequencing qualifier in Exercise 3



Course business scenario © Copyright IBM Corporation 2017

Figure 2-19. Enable the event sequencing qualifier in Exercise 3

© Copyright IBM Corp. 2017

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

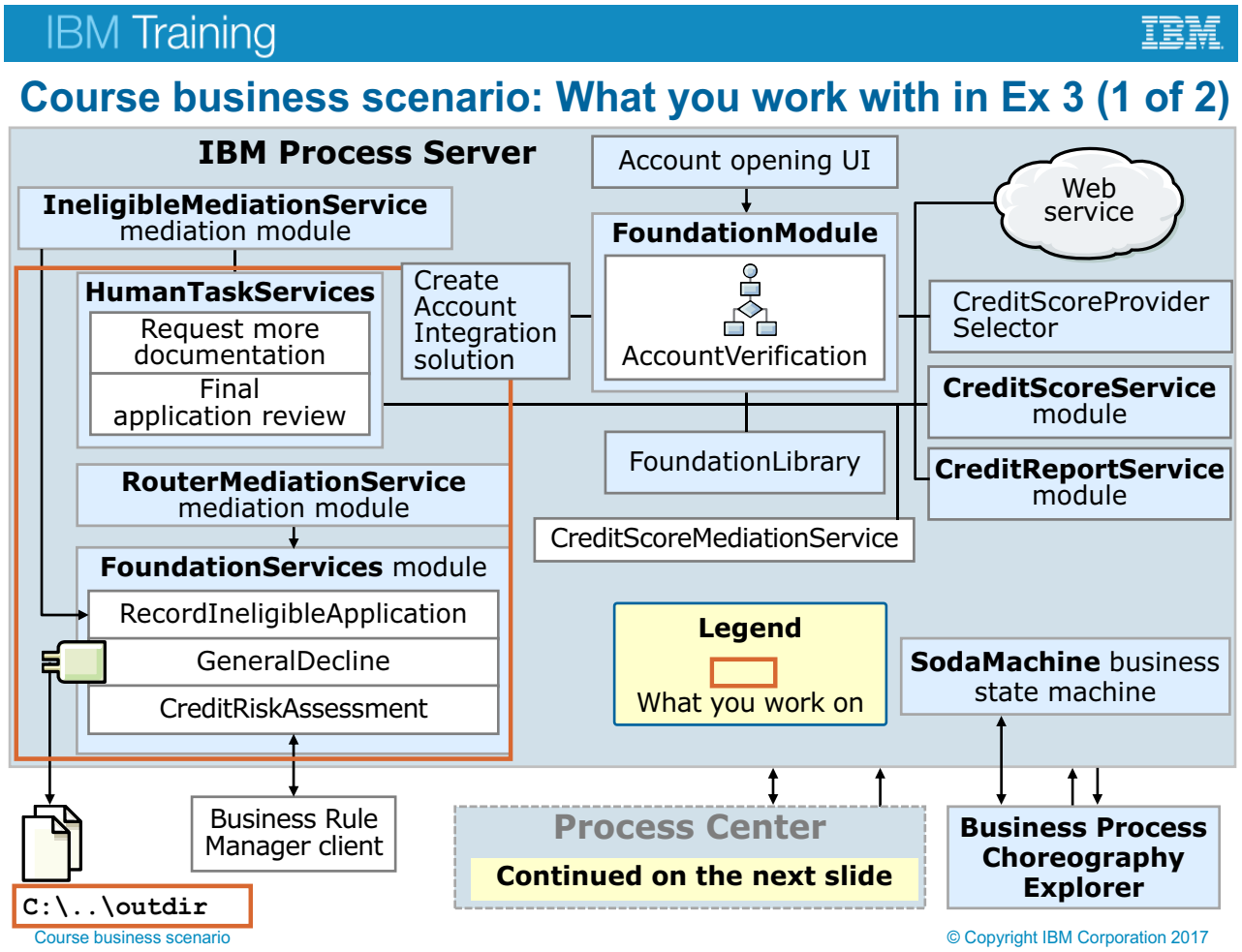


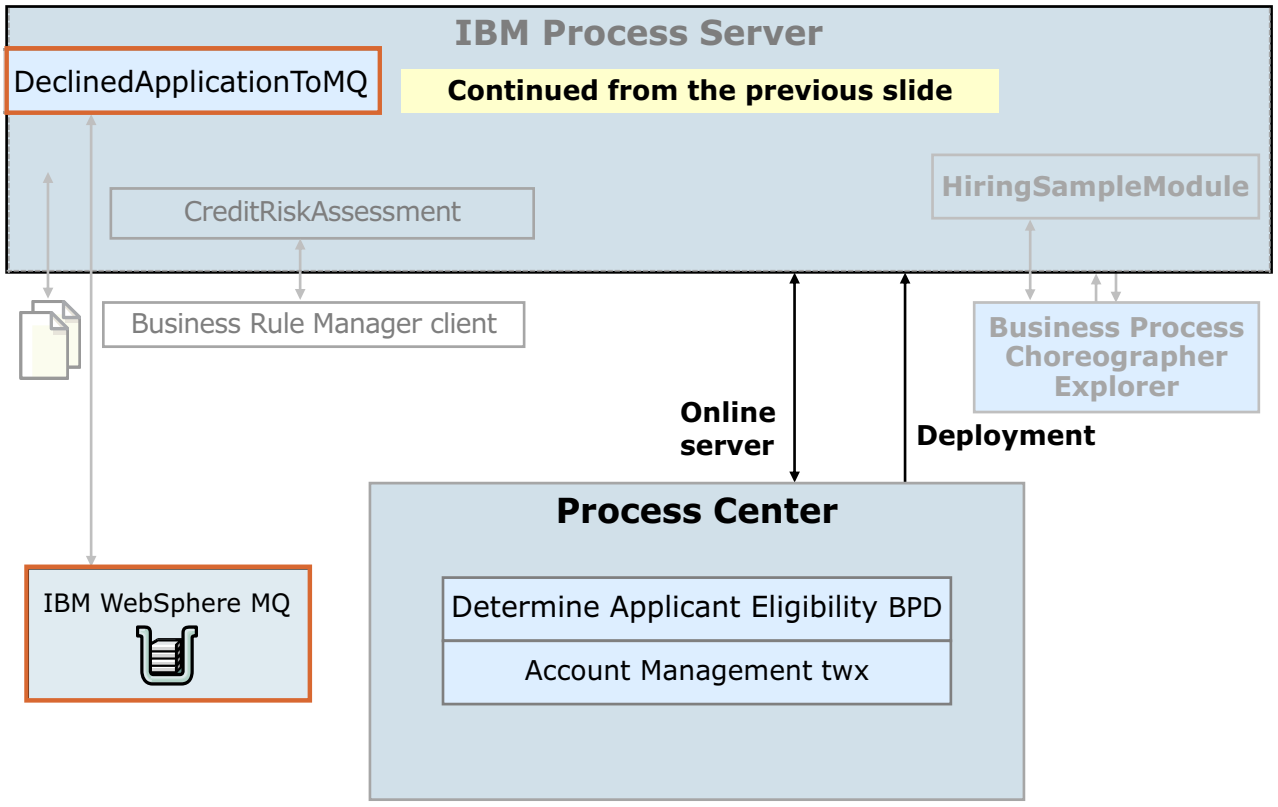
Figure 2-20. Course business scenario: What you work with in Ex 3 (1 of 2)

In this lab, you work with IBM Integration Designer.

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

IBM Training

Course business scenario: What you work with in Ex 3 (2 of 2)



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-21. Course business scenario: What you work with in Ex 3 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM WebSphere MQ. You do not work with Process Center.

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 4

Applying fault handlers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-22. Exercise 4

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

IBM Training

What are some key tasks that you do in Exercise 4?

- In business processes, some exceptions can be predicted and handled by defining fault handlers within the BPEL process
  - Fault handlers can also be defined to handle unexpected, arbitrary faults
- This exercise shows you how to implement fault handlers in a business process

© Copyright IBM Corporation 2017

Figure 2-23. What are some key tasks that you do in Exercise 4?

© Copyright IBM Corp. 2017

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

IBM Training

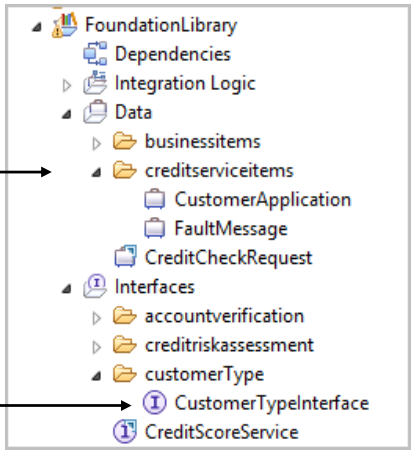
Components that are required for Exercise 4

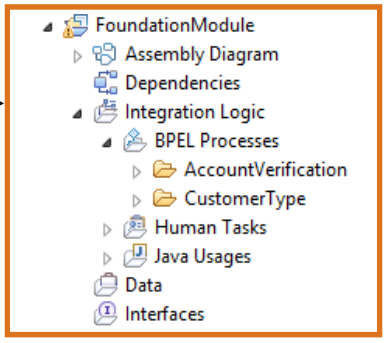
Prebuilt components that are imported in the lab:

1. **CustomerApplication** business object
2. **FoundationLibrary** library
3. **FoundationModule** module
4. **AccountVerification** process

New components that you create in the lab:

1. **CustomerApplication** business object
2. **FaultMessage** business object
3. **CustomerTypeInterface** interface
4. **CustomerType** BPEL process
5. **IsValidType** boolean variable
6. **FaultMessage** variable of FaultMessage data type
7. **CustomerTypePartner** partner reference
8. Several fault handlers





Course business scenario

© Copyright IBM Corporation 2017

Figure 2-24. Components that are required for Exercise 4

The following business objects are used in this exercise:

- **CustomerApplication**: The simple (non-hierarchical) business object that describes the characteristics of a particular customer application record. This generic business object (GBO) is instantiated from the user input through the human task.

The following interfaces are used along with the **CustomerApplication** business object:

- **CustomerApplicationDelivery**: Contains a single one-way operation that passes a single part, named **inputCustomerApplication** of type **CustomerApplication**, on the request. This interface is used on the definition of the business process and exposed as part of the exported SCA binding, which the other SCA components call.
- **CustomerApplicationRequest**: Contains a single request/response operation that passes a single part, named **inputCustomerApplication** of type **CustomerApplication**, on the request; and a single part, named **outputCustomerApplication** of type **CustomerApplication**, on the response. With the use of this interface, it is expected that any changes that are made to **CustomerApplication** on the receiving end are passed back to the process. This interface is used to define reference partners in the business process. Using the same interface simplifies the data exchange between the processes.

© Copyright IBM Corp. 2017

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

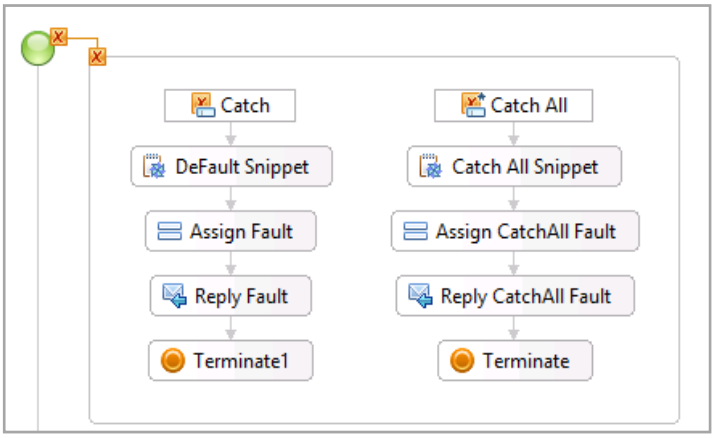
You also build a new **FaultMessage** data type to encapsulate two string attributes, **FaultID** and **FaultData**, which are used to provide more information to the **AccountVerification** business process that catches the returned faults.

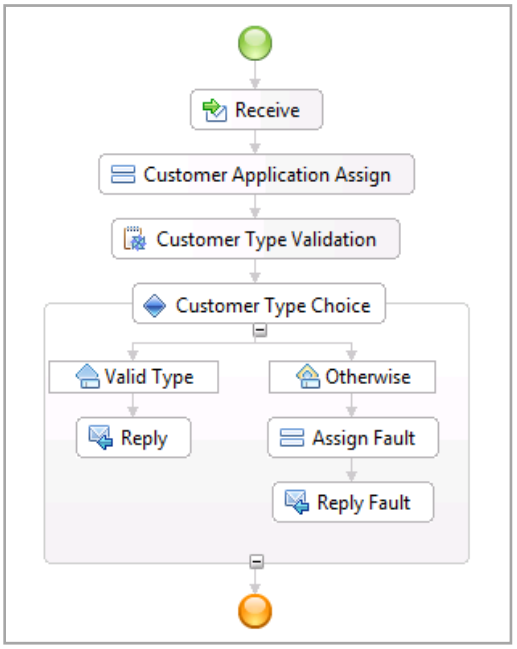
© Copyright IBM Corp. 2017

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

IBM Training

Create fault handlers to catch and throw faults in Exercise 4





© Copyright IBM Corporation 2017

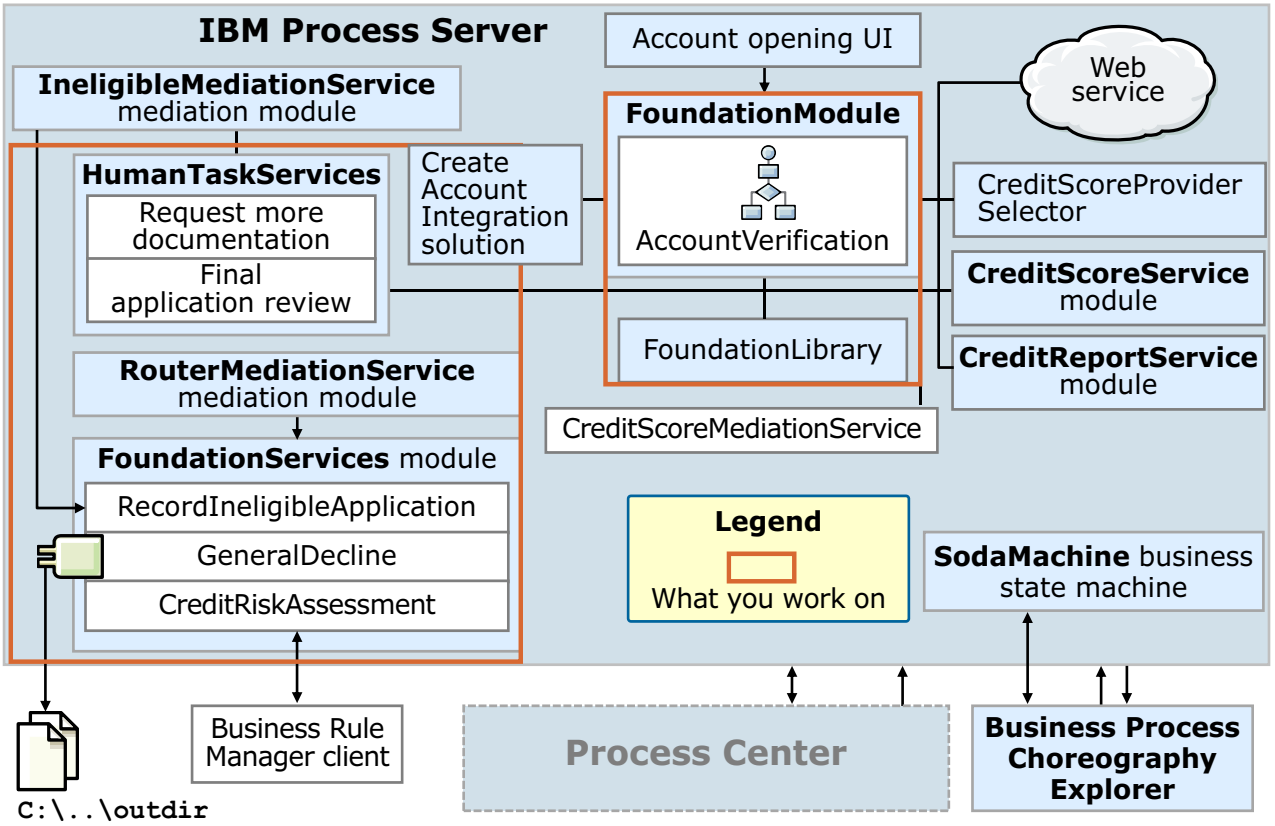
Figure 2-25. Create fault handlers to catch and throw faults in Exercise 4

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 4



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-26. Course business scenario: What you work with in Ex 4

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 5

Applying a compensation handler to WS-BPEL

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-27. Exercise 5

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

IBM Training

What are some key tasks that you do in Exercise 5?

- In this exercise, you implement compensation handlers in a business process
  - Since you cannot predict exactly when a particular fault might happen, you need to define a sequence of activities to react to such events, and possibly undo the activities that are already committed
  - The compensation handlers can handle these types of scenarios

© Copyright IBM Corporation 2017

Figure 2-28. What are some key tasks that you do in Exercise 5?

© Copyright IBM Corp. 2017

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

IBM Training

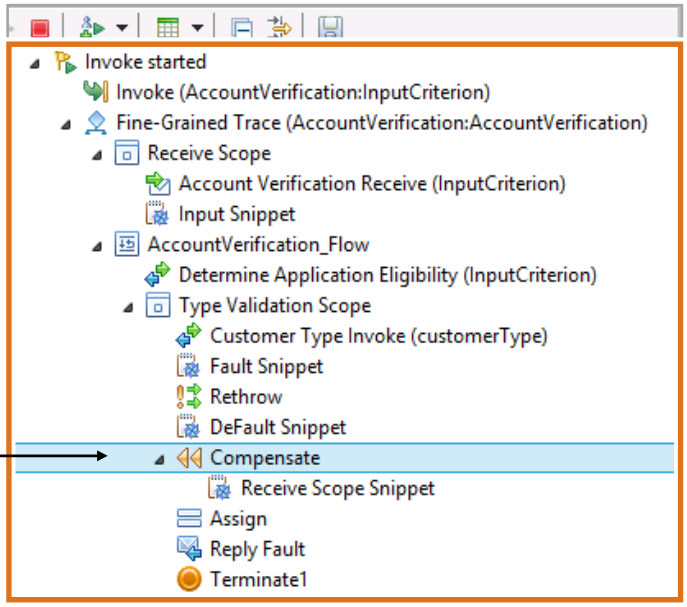
Components that are required for Exercise 5

Prebuilt components that are imported in the lab:

1. **AccountVerification BPEL**
2. **Default snippet**

New components that you create in the lab:

1. **Default Compensate** activity
2. **Compensation handler** for **Receive Scope** activity
3. **Compensation handler** for **Type Validation Scope** activity
4. **Compensation handler** for **Customer Type Scope** activity
5. **Receive Scope** activity
6. **Type Validation Scope snippet**
7. **Input Snippet**
8. **Status Choice** activity



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-29. Components that are required for Exercise 5

In a long-running process, each transaction is committed individually. If the process fails, a compensate activity specifies how to deal with each transaction that needs to be compensated. The compensation handler of a scope, or a process, triggers the compensation.

AccountVerification is already checking the customer type information through a service call to the **CustomerType** process. You now add logic to the **AccountVerification** process to also check the customer status and raise a fault for a detected inactive customer status setting. The logic checks the status of the customer. If the customer is active, no further action is necessary (other than a status message to the server logs or console).

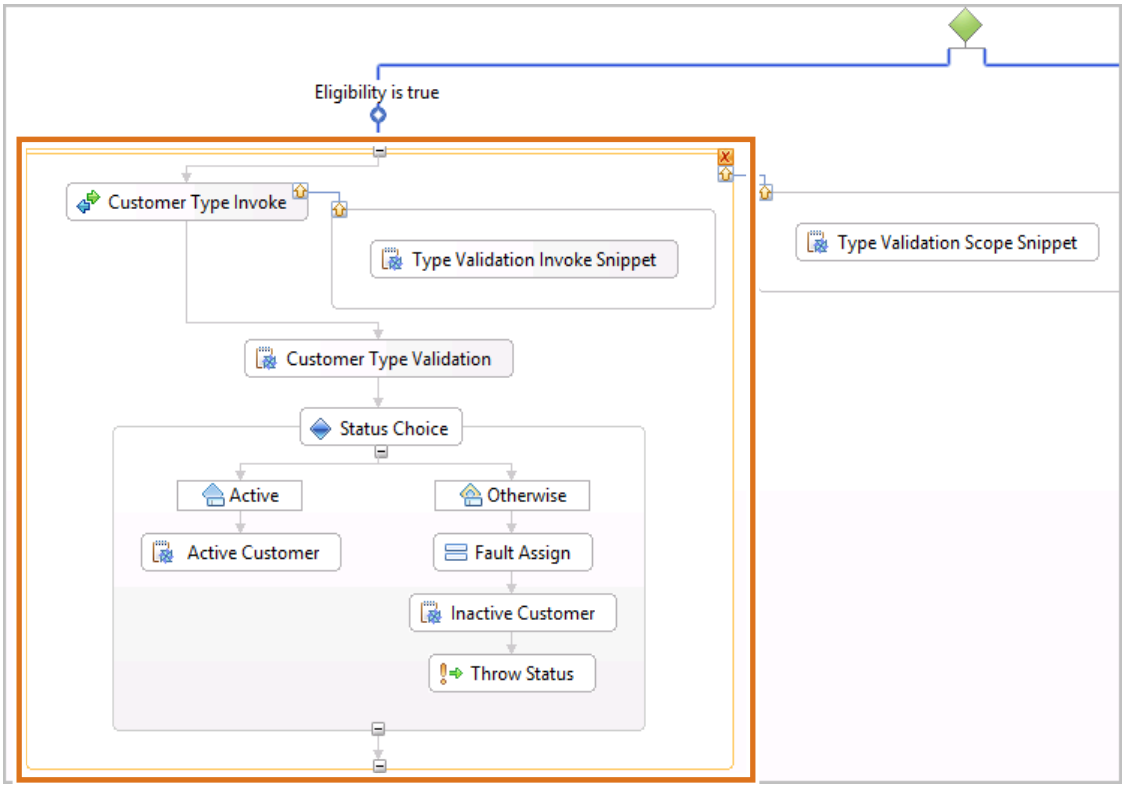
For inactive customer status, a fault is thrown.

© Copyright IBM Corp. 2017

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

IBM Training

Building specific blocks of the AccountVerification process in Exercise 5



Course business scenario

© Copyright IBM Corporation 2017

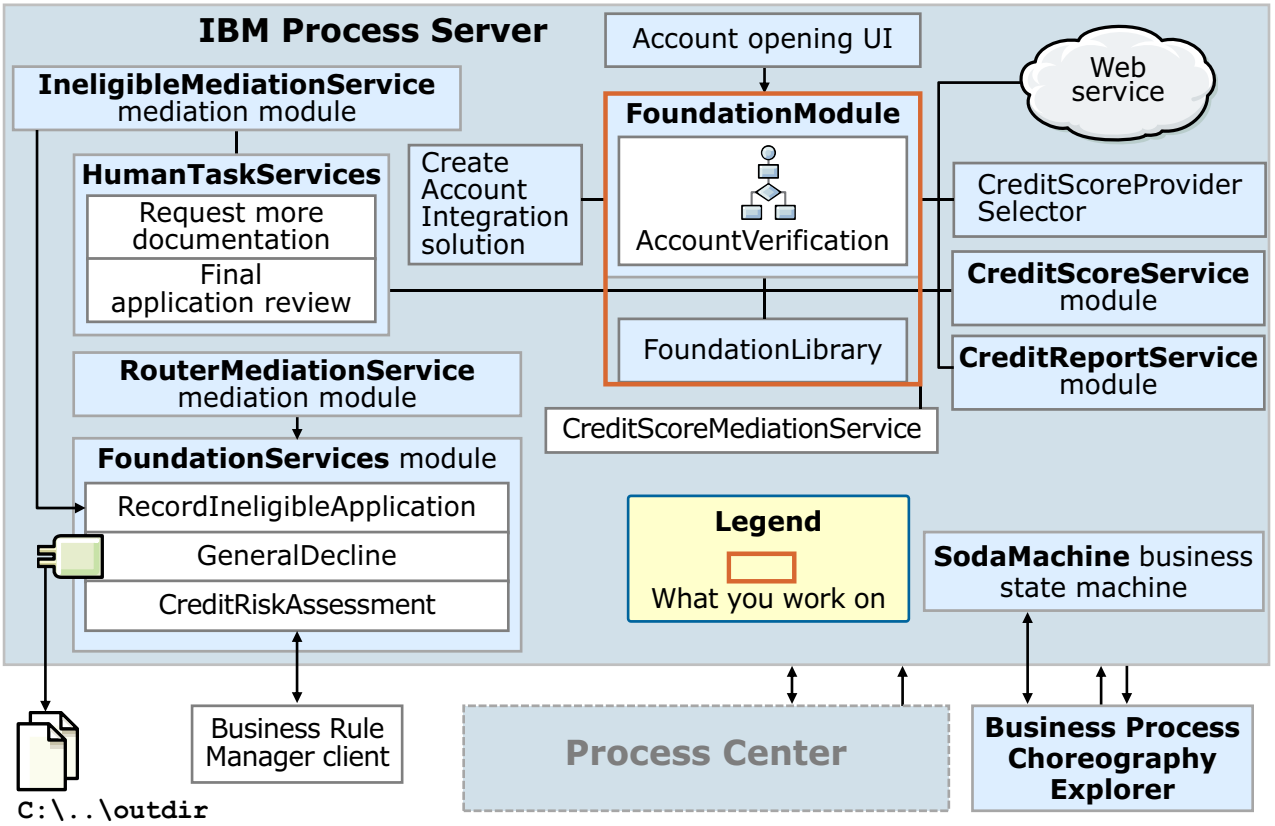
Figure 2-30. Building specific blocks of the AccountVerification process in Exercise 5

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 5



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-31. Course business scenario: What you work with in Ex 5

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 6

Working with business state machines

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-32. Exercise 6

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

IBM Training

What are some key tasks that you do in Exercise 6?

- Explore the building blocks of a SodaMachine sample with the business state machine editor
- Test the SodaMachine state machine with Business Process Choreographer Explorer

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-33. What are some key tasks that you do in Exercise 6?

A business state machine is an event-driven business application in which external operations trigger changes that guide the state machine from one discrete mode to another. Each mode is an individual state, and this mode determines what activities and operations can occur. The state machines are service components, which represent business processes that are based on states and events instead of a sequential business process model.

In this exercise, you create business state machines in IBM Integration Designer by using the business state machine editor. You use the SodaMachine sample to demonstrate a business state machine that is used to control the purchase of a can of soda from a vending machine.

© Copyright IBM Corp. 2017

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

IBM Training

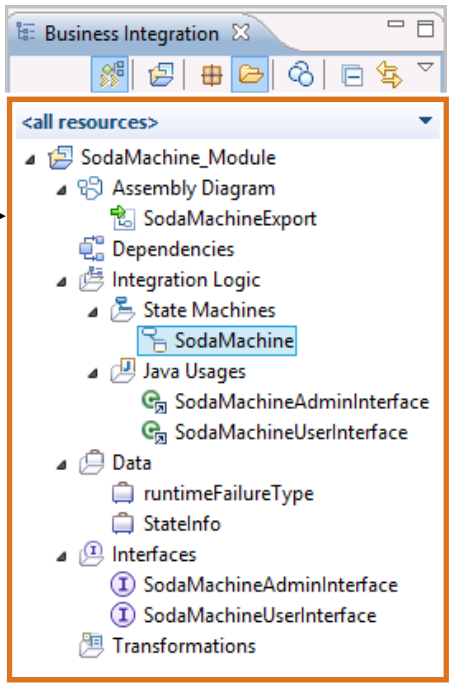
Components that are required for Exercise 6

Prebuilt components that are imported in the lab:

1. **SodaMachine BSM**

New components that you create in the lab:

None



© Copyright IBM Corporation 2017

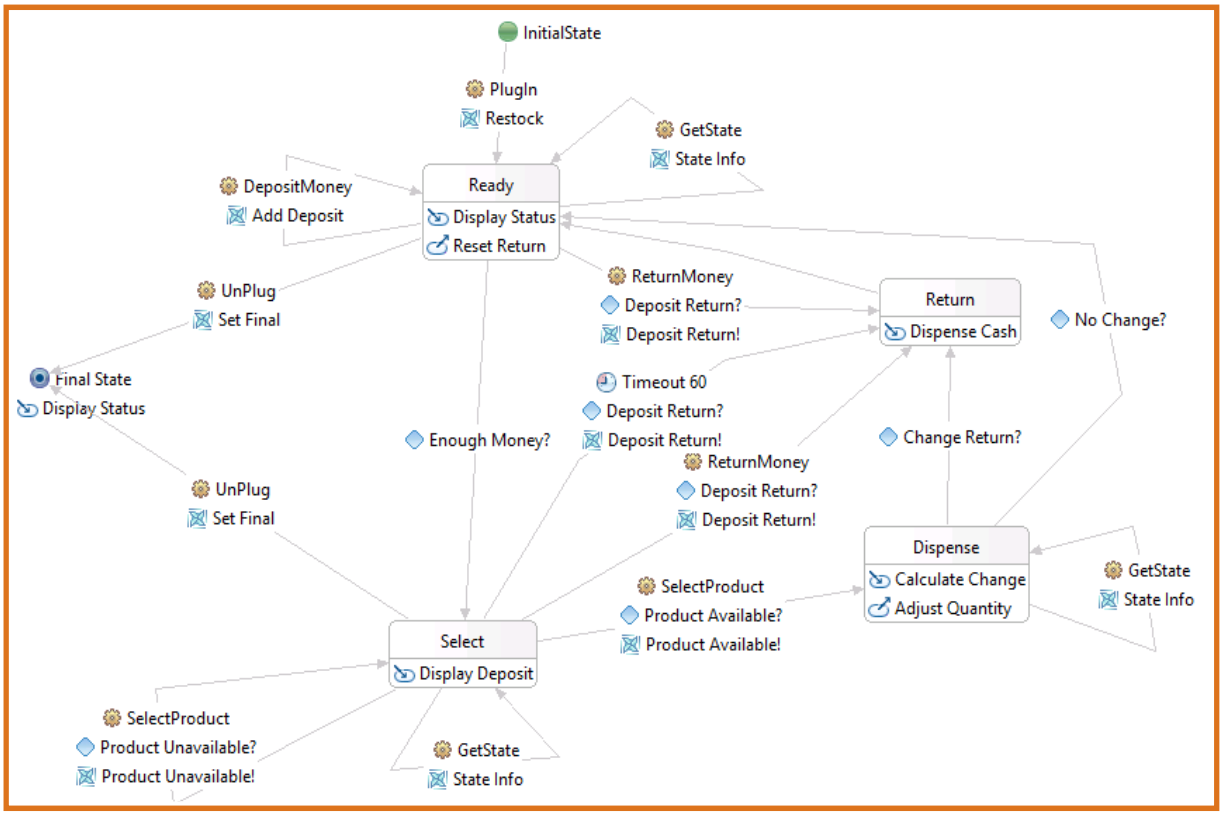
Figure 2-34. Components that are required for Exercise 6

© Copyright IBM Corp. 2017

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

IBM Training

Explore SodaMachine business state machine in Exercise 6



© Copyright IBM Corporation 2017

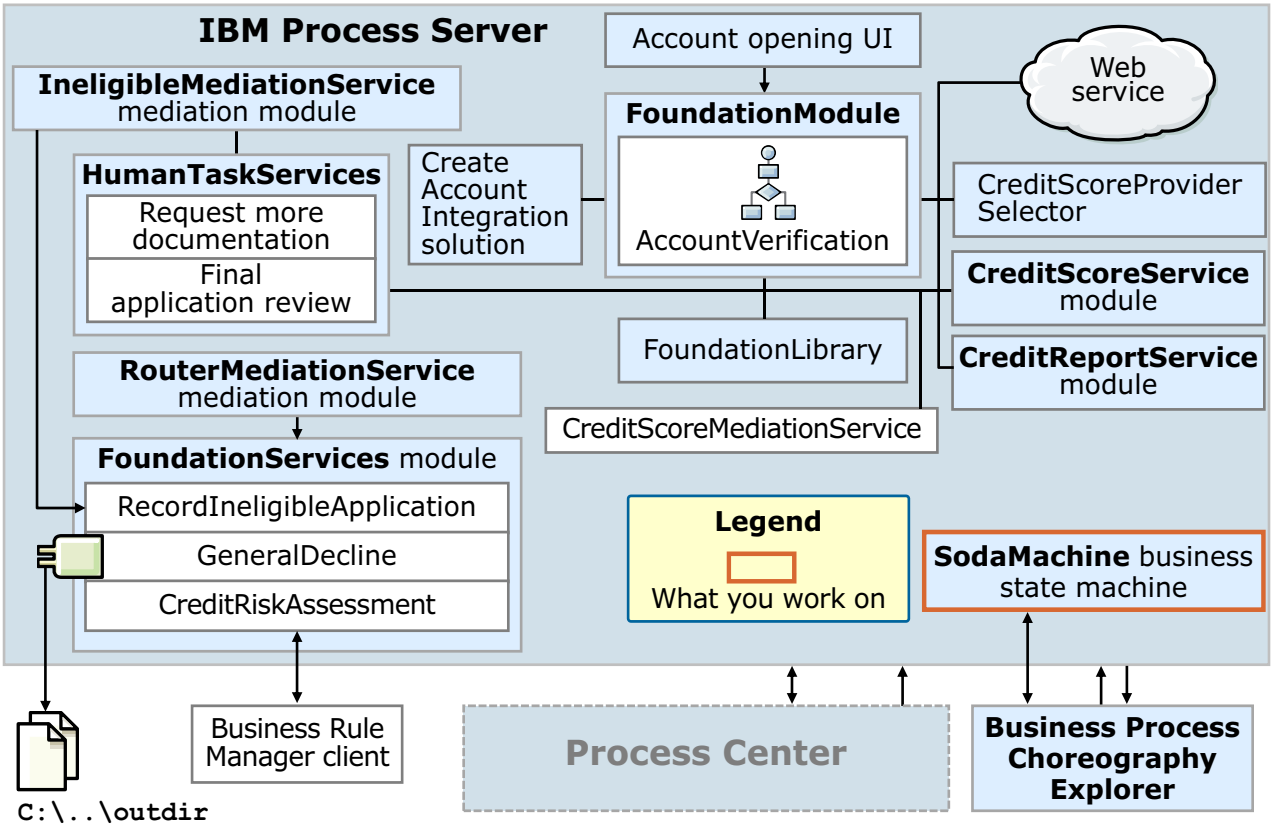
Figure 2-35. Explore SodaMachine business state machine in Exercise 6

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 6



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-36. Course business scenario: What you work with in Ex 6

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 7

Defining transactional behavior in SCA applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-37. Exercise 7

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

What are some key tasks that you do in Exercise 7?

- In this exercise, you view the transaction settings in the Account Opening end-to-end lab scenario
  - You view the transaction propagation settings in the integration solution diagram of the scenario, and you view propagation settings for an individual module with the assembly diagram
- After viewing the transaction propagation settings, you examine the transaction boundaries around activities in a BPEL process

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-38. What are some key tasks that you do in Exercise 7?

When you establish a transaction, you want to have confidence that all the operations commit successfully or roll back completely. You must ensure that a transaction can propagate to all components that must participate in the unit of work. Quality of service qualifiers and invocation style control the propagation of transactions in a module. A number of conditions must exist for a transaction to propagate successfully:

- A transaction qualifier must be set on the implementation, and the qualifier must specify a global transaction.
- The implementation must be started synchronously.
- The component must join a propagated transaction.
- The transaction cannot be suspended on any of the participating references.
- The preferred interaction style on a participating interface cannot influence the invocation style to become asynchronous.

In this exercise, you examine transaction settings and set transaction-related quality of service qualifiers. Setting up transactions in an SOA environment can be an intricate exercise. The assembly editor, BPEL editor, and integration solution diagram provide aids to make it easier to understand the way transactions are processed. However, you must understand the mechanics of transactions before you try to create them or alter the default settings.

© Copyright IBM Corp. 2017

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

IBM Training

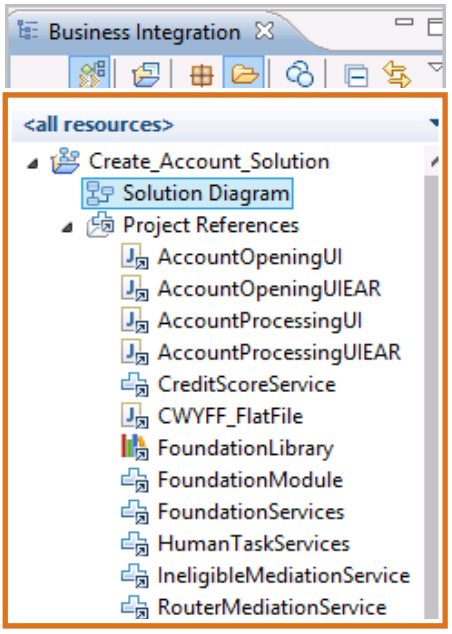
Components that are required for Exercise 7

Prebuilt components that are imported in the lab:

1. **Create_Account_Solution** solution diagram

New components that you create in the lab:

None



© Copyright IBM Corporation 2017

Figure 2-39. Components that are required for Exercise 7

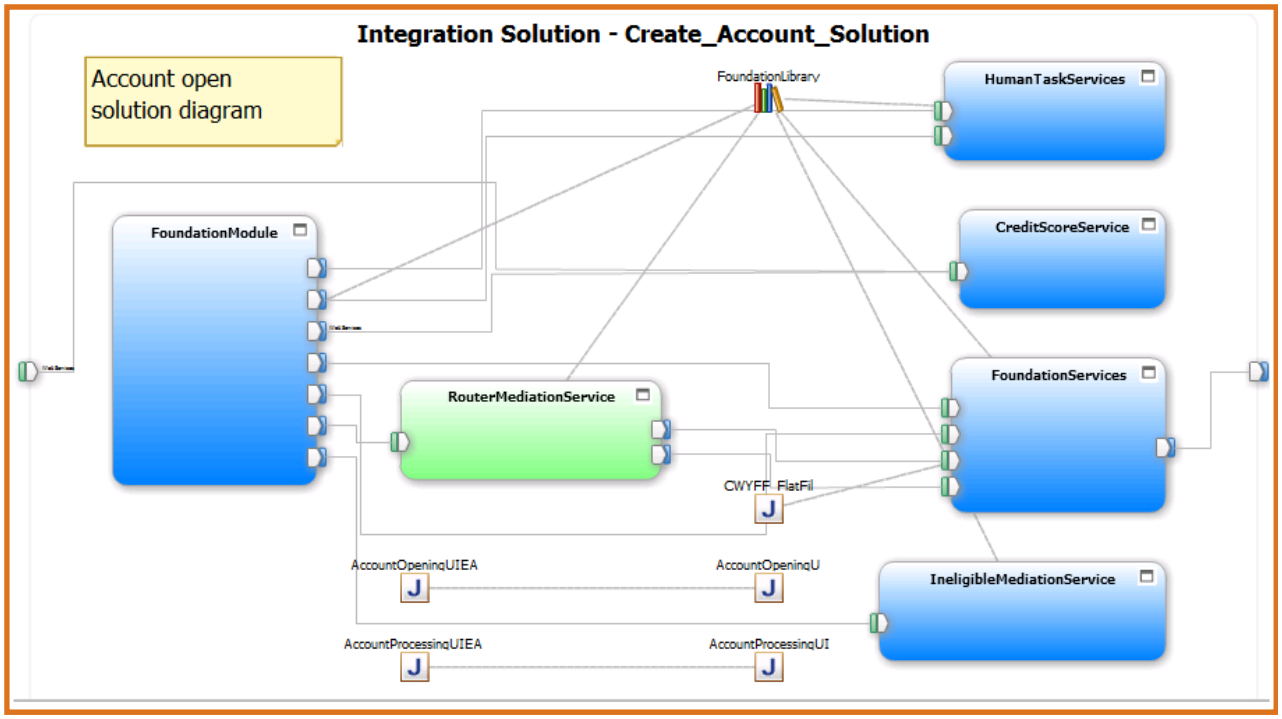
You view transaction settings in applications that are developed in IBM Integration Designer in several ways. You can view transaction propagation settings in the assembly editor and in integration solution diagrams. In this portion of the exercise, you examine transaction settings in the account verification lab scenario.

© Copyright IBM Corp. 2017

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

IBM Training

Explore Create Account Solution diagram in Exercise 7



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-40. Explore Create Account Solution diagram in Exercise 7

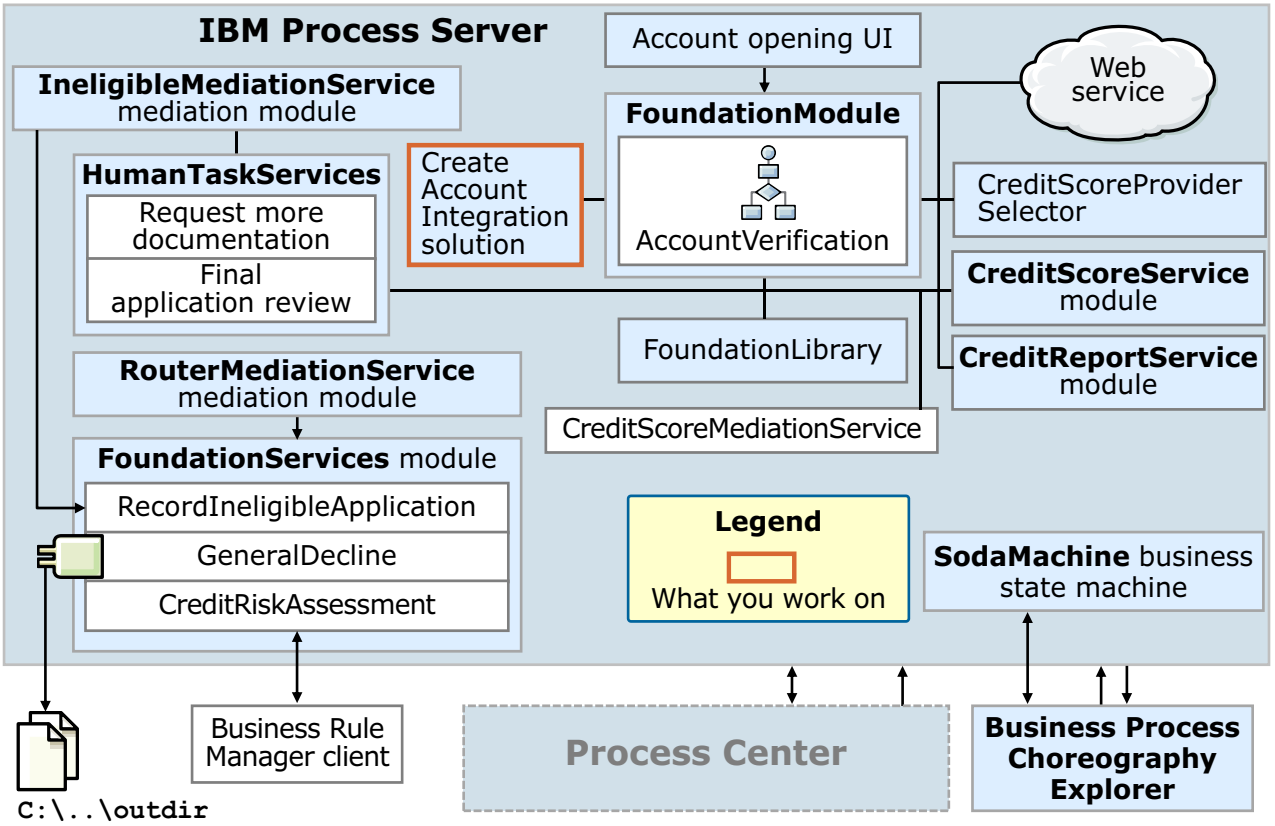
In the integration solution editor, you view and highlight the transaction path between components in a single module or between the components in all of the modules that make up your integration solution. You also show and highlight the transaction path from a specific component to the other components with which it interacts.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 7



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-41. Course business scenario: What you work with in Ex 7

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 8

Creating flexible business processes

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-42. Exercise 8

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

IBM Training

What are some key tasks that you do in Exercise 8?

- Use the Business Process Choreographer Explorer client to dynamically change the execution path of a running process instance
- Implement a selector component to choose between multiple service invocations

© Copyright IBM Corporation 2017

Figure 2-43. What are some key tasks that you do in Exercise 8?

© Copyright IBM Corp. 2017

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

IBM Training

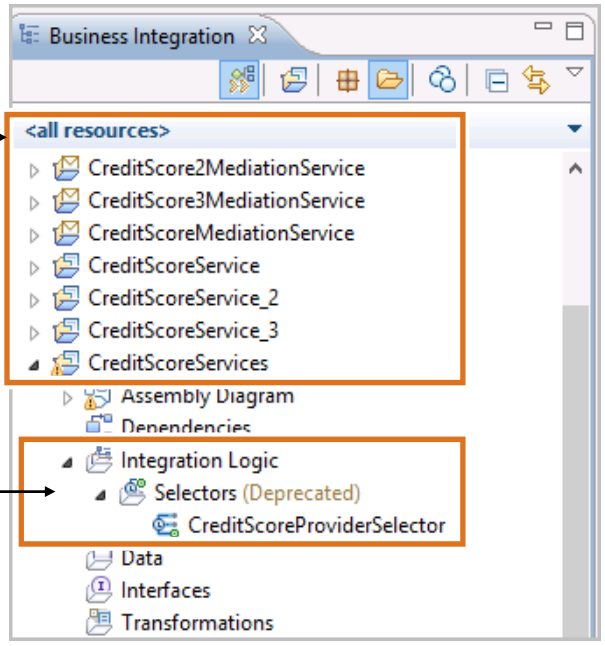
Components that are required for Exercise 8

Prebuilt components that are imported in the lab:

1. **CreditScore2mediationService**
2. **CreditScore3mediationService**
3. **CreditScoreService**
4. **CreditScoreService_2**
5. **CreditScoreService_3**
6. **CreditScoreServices**

New components that you create in the lab:

1. **CreditScoreMediationService** mediation subflow
2. **CreditScoreProviderSelector**



© Copyright IBM Corporation 2017

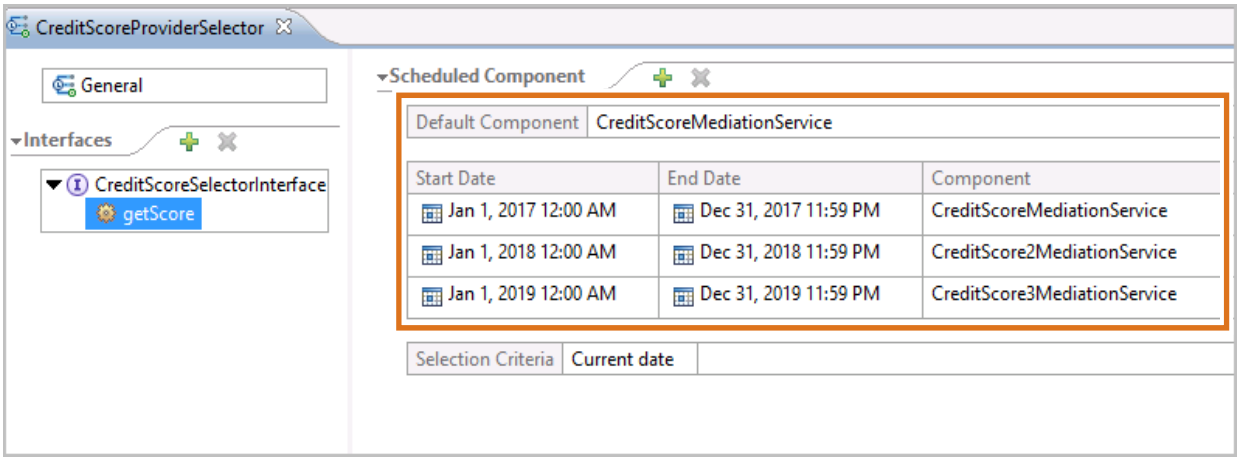
Figure 2-44. Components that are required for Exercise 8

© Copyright IBM Corp. 2017

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

IBM Training

Create the CreditScoreProviderSelector component in Exercise 8



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-45. Create the CreditScoreProviderSelector component in Exercise 8

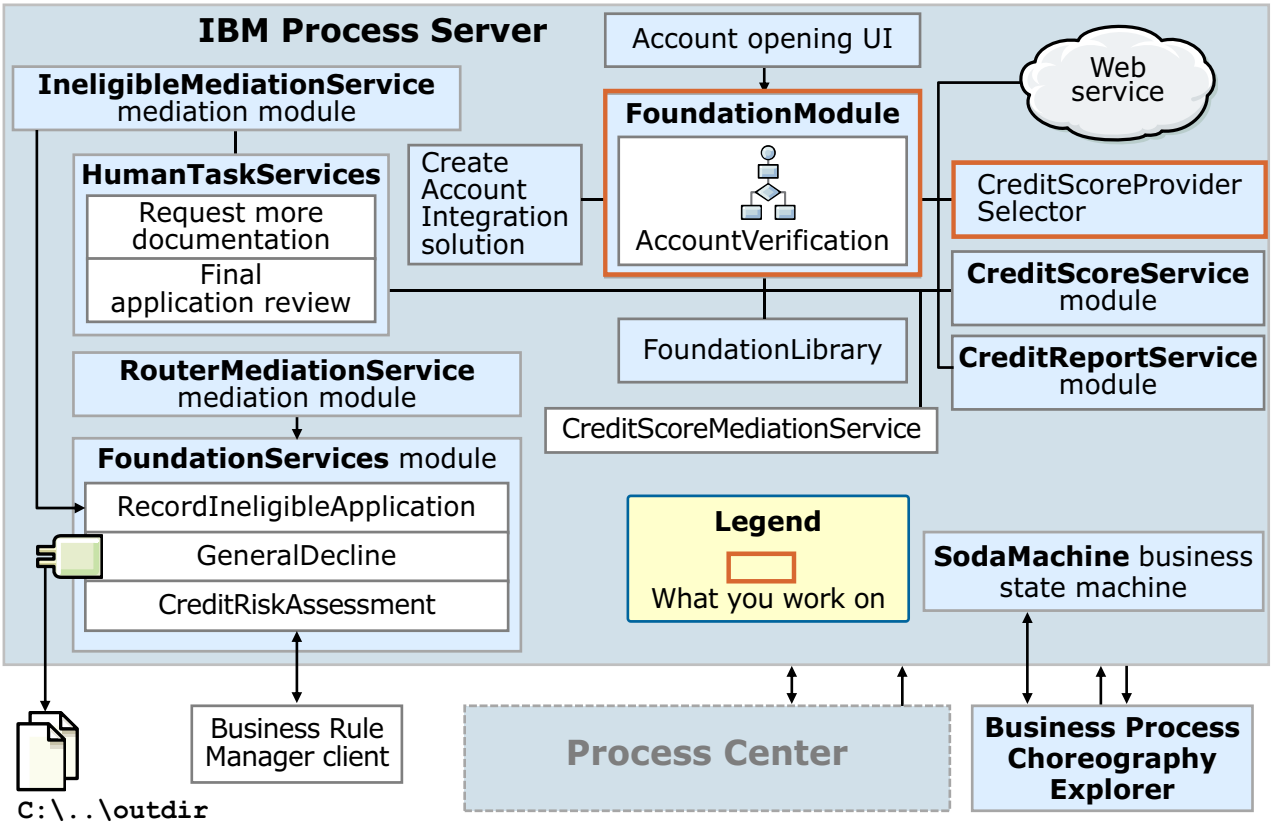
In the exercise, after creating the implementation, you test the application by using **companyName** IBM. Depending on when this test is run, the resulting credit score is either 9, 10, or 11. Recall from the selector component criteria that three possible mediation services can be invoked. **CreditScoreMediationService** is invoked between **1/1/17** and **12/31/17** and returns a **credit** score of 11. **CreditScore2MediationService** is invoked between **1/1/18** and **12/31/18** and returns a **credit score** of 10. **CreditScore3MediationService** is invoked between **1/1/19** and **12/31/19** and returns a **credit score** of 9. The result varies depending on the year in which the service is invoked. Any year other than 2017, 2018, or 2019 results in invoking the **Default Component**, which is **CreditScoreMediationService**, in which case the credit score is always 11.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 8



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-46. Course business scenario: What you work with in Ex 8

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 9

Working with static relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-47. Exercise 9

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

IBM Training

What are some key tasks that you do in Exercise 9?

- Implement and test a static relationship to map corresponding values between business object fields

© Copyright IBM Corporation 2017

Figure 2-48. What are some key tasks that you do in Exercise 9?

When you integrate multiple applications, it is not unusual to represent commonly used data elements in multiple formats. For example, one system might represent state or province names by using two-letter abbreviations, while other systems might represent the same values by using numeric representations. Rather than writing application logic to do these conversions, you can create a relationship in IBM Integration Designer that can be invoked from a data map.

In this exercise, you modify the WebSphere MQ binding exercise to include a static relationship. In this case, it is determined that the customer service application, which receives the “declined application” messages sent to it through WebSphere MQ, uses a numeric code to represent the Country field in the credit application. However, the sending application uses a character string to represent the Country field. You create the static relationship, integrate it into the existing application, and then verify that the relationship works as expected by testing it.

© Copyright IBM Corp. 2017

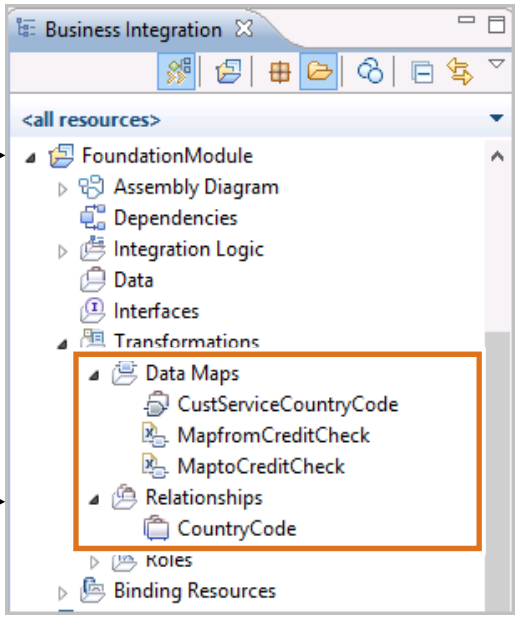
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 9

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CustomerApplication** business object
3. **AccountVerification** BPEL



New components that you create in the lab:

1. **CountryCode** static relationship
2. **CustServiceCountryCode** BO map

© Copyright IBM Corporation 2017

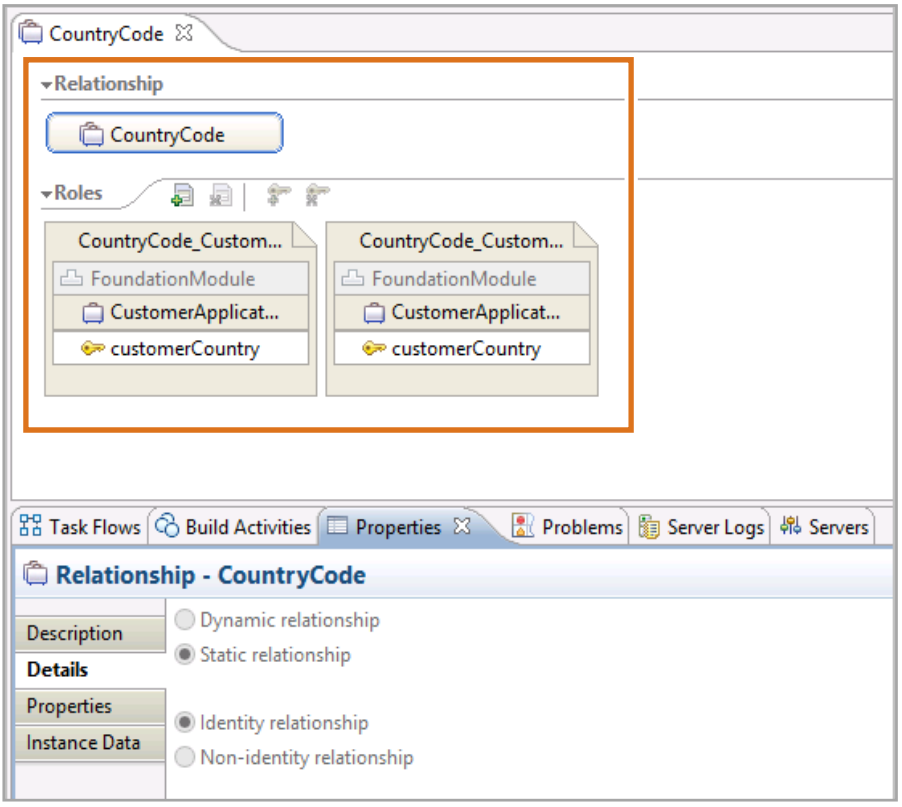
Figure 2-49. Components that are required for Exercise 9

© Copyright IBM Corp. 2017

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

IBM Training

Create the CountryCode static relationship in Exercise 9



© Copyright IBM Corporation 2017

Figure 2-50. Create the CountryCode static relationship in Exercise 9

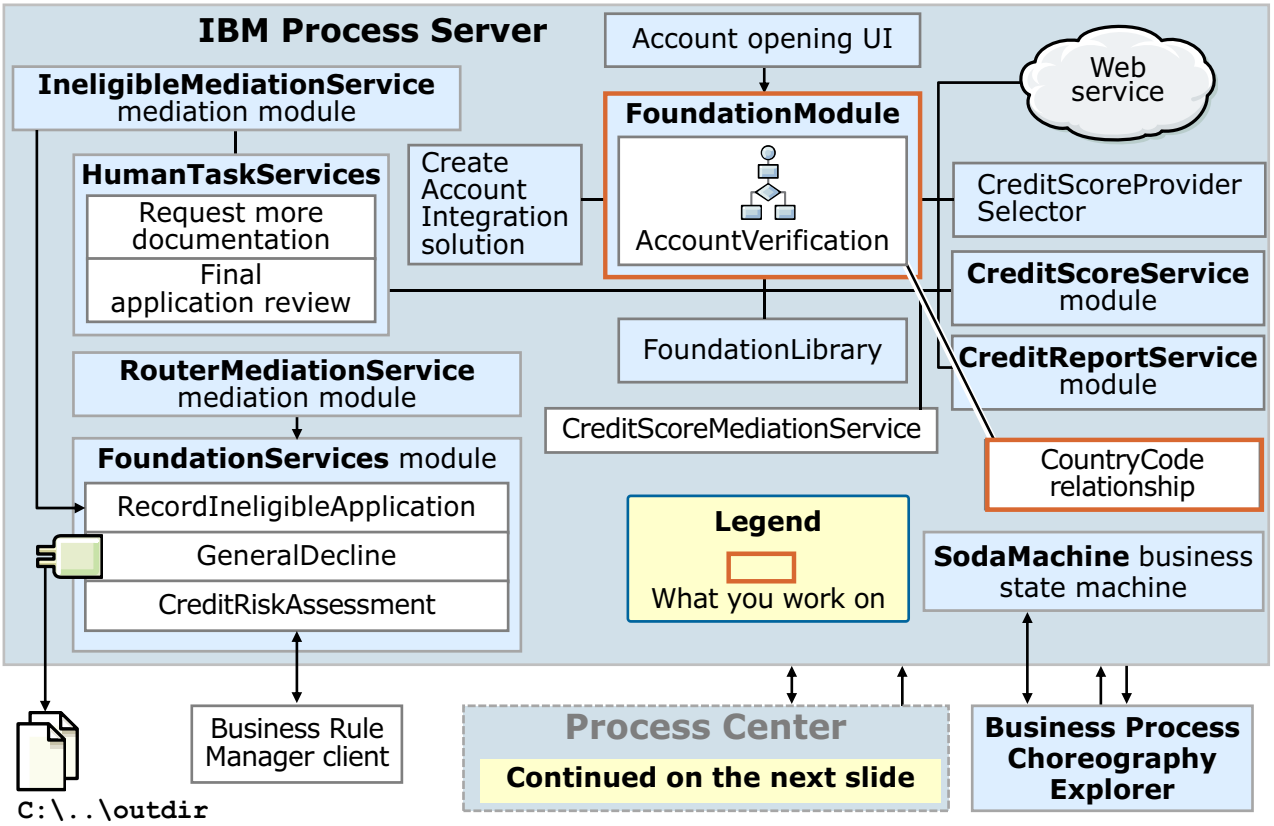
In this exercise, you test the module by using a customer that has a high credit risk. Because of this risk, the application is declined. The process flow takes the “Application is declined” path, which invokes the Generate Decline and Record Declined Application activities. The CountryCodeMap activity is then invoked to translate the country code that the DeclinedApplicationToMQ activity follows. The DeclinedApplicationToMQ activity sends the message to a WebSphere MQ queue.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 9 (1 of 2)



Course business scenario

© Copyright IBM Corporation 2017

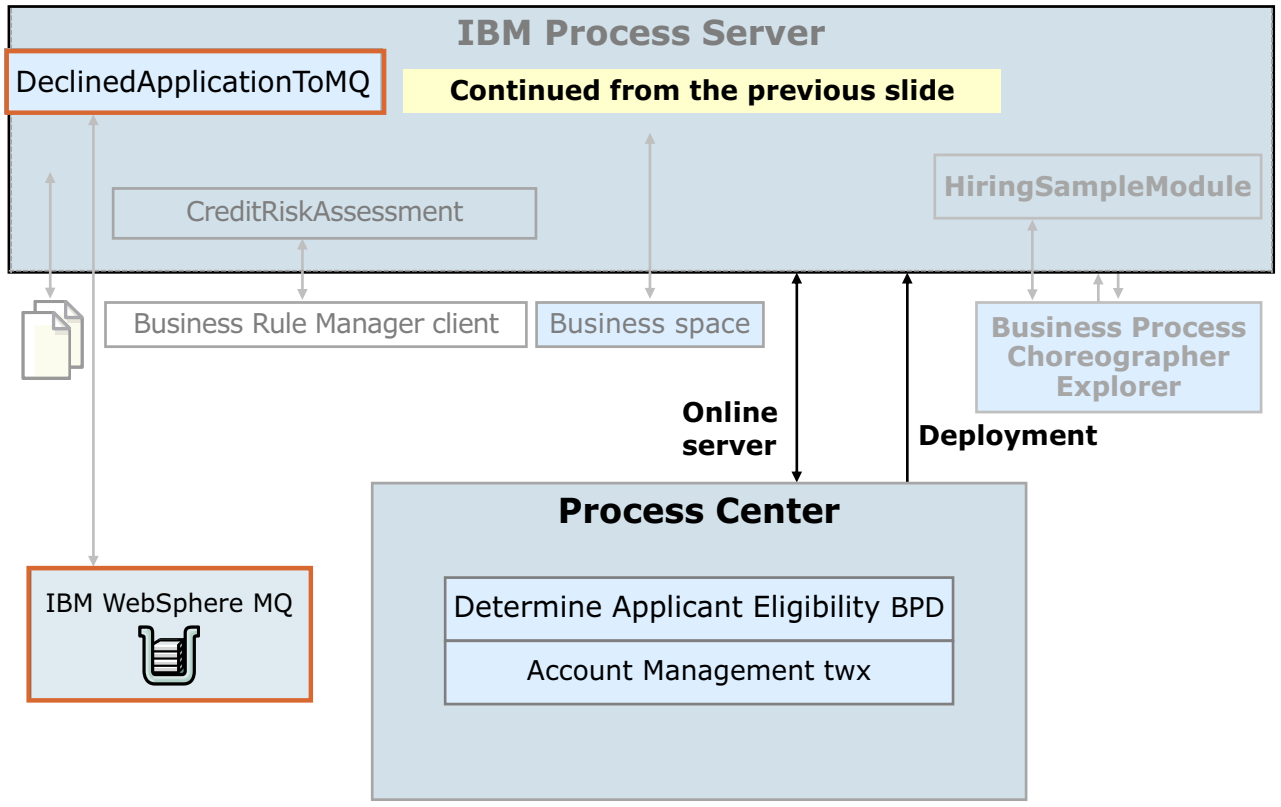
Figure 2-51. Course business scenario: What you work with in Ex 9 (1 of 2)

In this lab, you work with IBM Integration Designer.

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

IBM Training

Course business scenario: What you work with in Ex 9 (2 of 2)



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-52. Course business scenario: What you work with in Ex 9 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM WebSphere MQ. You do not work with Process Center.

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 10

Implementing a mediation flow

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-53. Exercise 10

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

IBM Training

What are some key tasks that you do in Exercise 10?

- Implement a mediation flow with multiple mediation primitives, including fan-out, fan-in, service invoke, XSL transformation, message element setter, and custom mediation

© Copyright IBM Corporation 2017

Figure 2-54. What are some key tasks that you do in Exercise 10?

Any service can replace any other service that does the same processing as the replacing service. Replacing a service should not require any changes to the existing caller. For example, assume that application A starts the service of application B. If application C is functionally equivalent to application B, then application A must be able to start application C instead of application B, without the need to change application A in any manner.

Recall that in the scenario that is used for the exercises in this course, a credit check is run for all eligible credit applications. In this exercise, you replace the existing credit check application with one that queries three different credit report services. The application then returns the lowest of the three obtained credit scores to the calling service. You implement the new credit score inquiry service as a mediation module. This module uses an aggregation flow, which starts the three credit report services, determines the lowest score, and then returns that score to the account verification module.

© Copyright IBM Corp. 2017

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

IBM Training

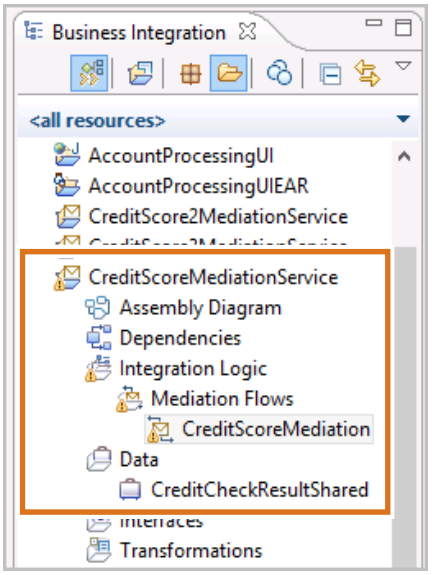
Components that are required for Exercise 10

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreMediationService**
3. **CreditScoreMediation** mediation flow
4. **AccountVerification** BPEL

New components that you create in the lab:

1. **CreditCheckResultShared** business object
2. **CustServiceCountryCode** business object map



© Copyright IBM Corporation 2017

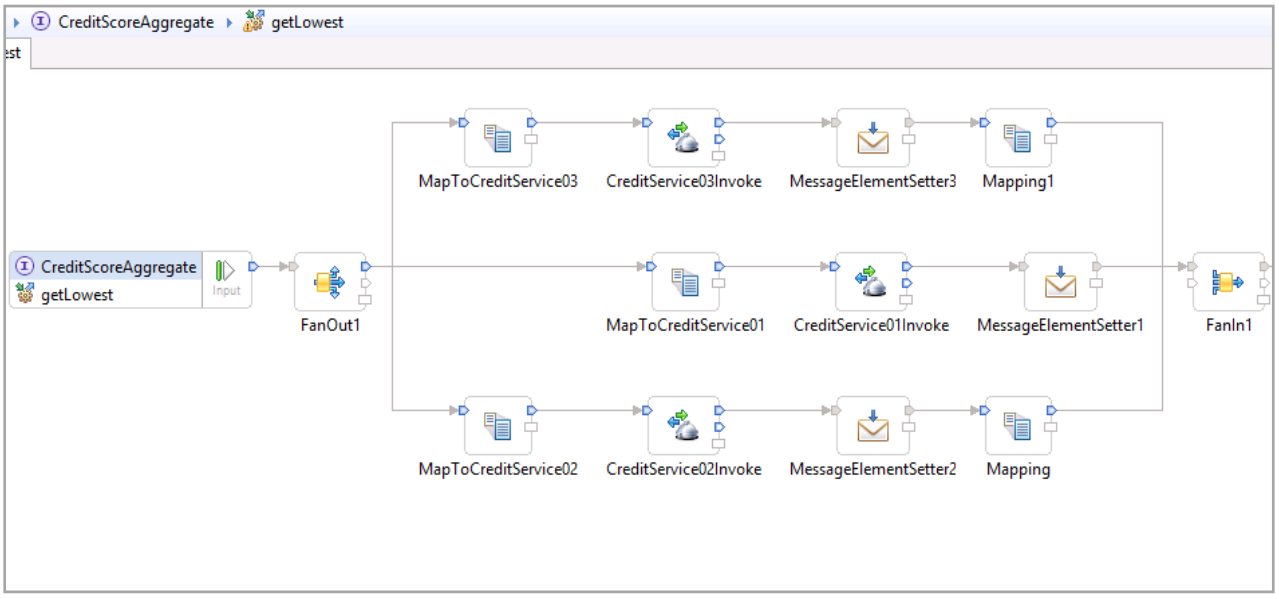
Figure 2-55. Components that are required for Exercise 10

© Copyright IBM Corp. 2017

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

IBM Training

Implement the CreditScoreMediation in Exercise 10



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-56. Implement the CreditScoreMediation in Exercise 10

In this section, you implement the logic in the mediation module to start the three credit score providers. The logic also determines which of the returned credit scores is the lowest, and then returns that score to the invoking service.

To implement the mediation flow, you use the following primitives:

- A fan-out to start the aggregation. From this fan-out, three paths exist, one for each of the credit score service imports.
- A service invoke for each of the credit score services.
- An XSL transformation before each of the credit score invocations to convert the inbound message to the data type that each service invocation requires.
- A message element setter, which follows each of the credit score invocations to copy the response messages from the credit score services into the shared context. The custom mediation primitive queries the shared context.
- Two more XSL transformations, which follow the message element setter primitives for the second and third service invocations (described in further detail in the exercise).
- A fan-in to mark the end of the aggregation block.

© Copyright IBM Corp. 2017

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

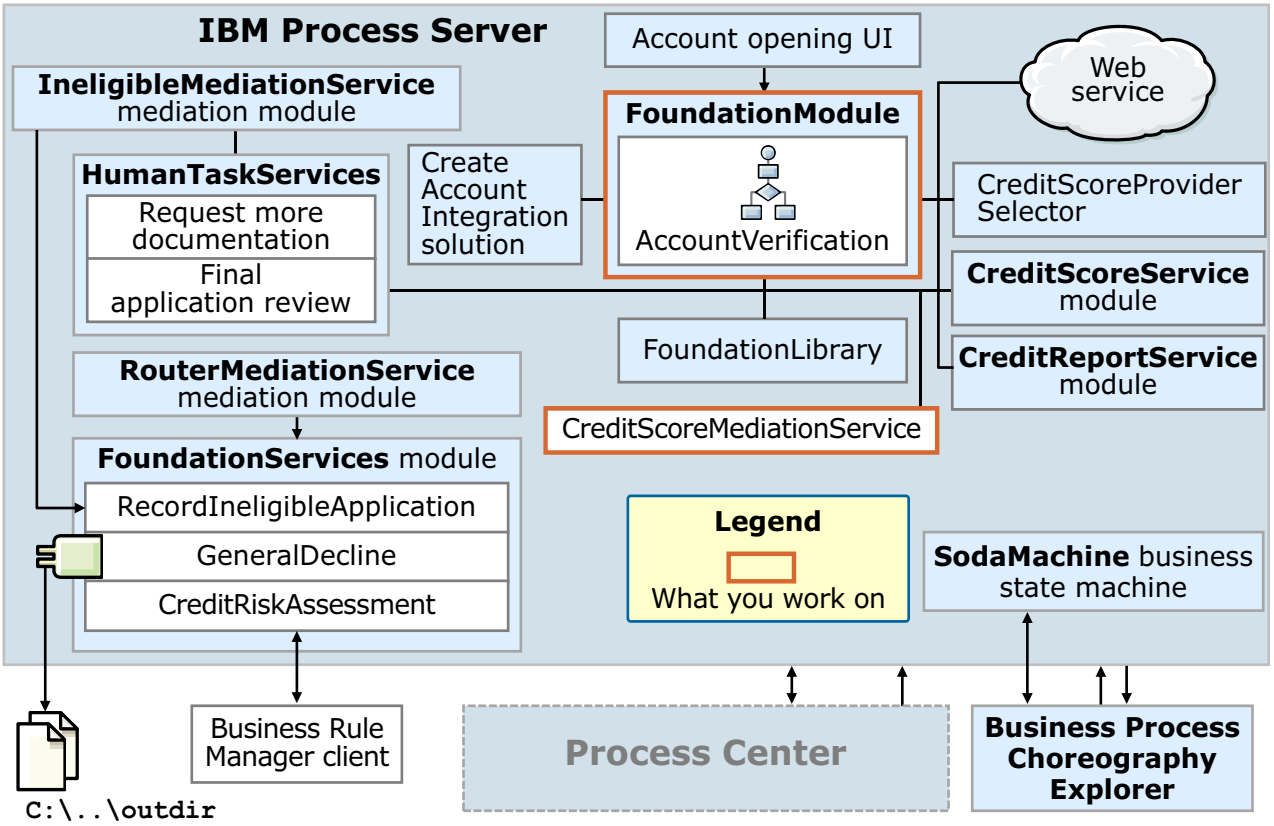
- A custom mediation primitive that determines which of the three credit scores that are returned is the lowest. This primitive constructs the output message that is based on the response message that returned the lowest credit score.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you with in Ex 10



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-57. Course business scenario: What you with in Ex 10

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 11

Writing a generic error handler for IBM Process Server

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-58. Exercise 11

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

IBM Training

What are some key tasks that you do in Exercise 11?

- In this exercise, you write generic error handler logic for IBM Process Server in a mediation subflow
  - The subflow handles runtime failures and logs the messages to the console

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-59. What are some key tasks that you do in Exercise 11?

Often times, you want to have a standard behavior to handle runtime exceptions regardless of failure point. In this exercise, you are going to learn how to define a mediation subflow that defines a standard behavior (or pattern) to handle errors. A mediation subflow is a preconfigured set of mediation primitives that are wired together to realize a common pattern or use case. Mediation subflows are run in the context of a parent flow, and can be reused in mediation flows or in subflows.

© Copyright IBM Corp. 2017

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

IBM Training

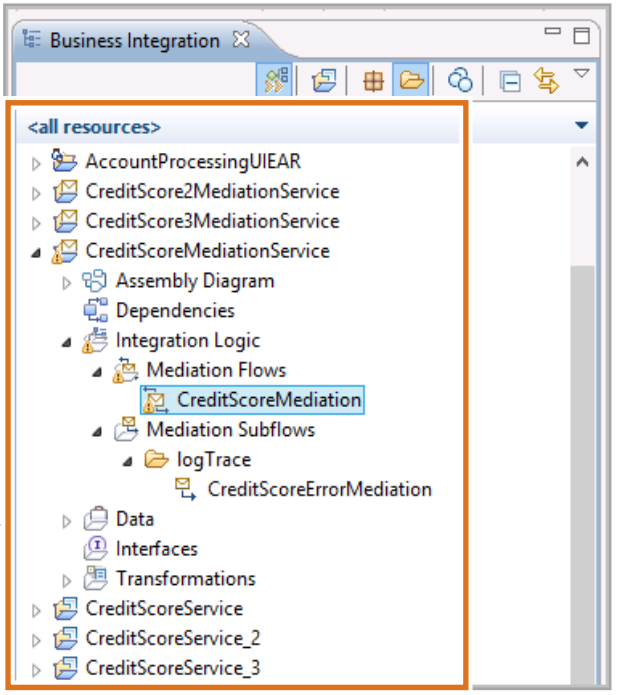
Components that are required for Exercise 11

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreMediationService**
3. **CreditScoreMediation** mediation flow
4. **AccountVerification** BPEL
5. **CreditScoreService**
6. **CreditScoreService_2**
7. **CreditScoreService_3**

New components that you create in the lab:

1. **CreditScoreErrorMediation** mediation subflow
2. **HandleCreditServiceError** mediation subflow



Course business scenario

© Copyright IBM Corporation 2017

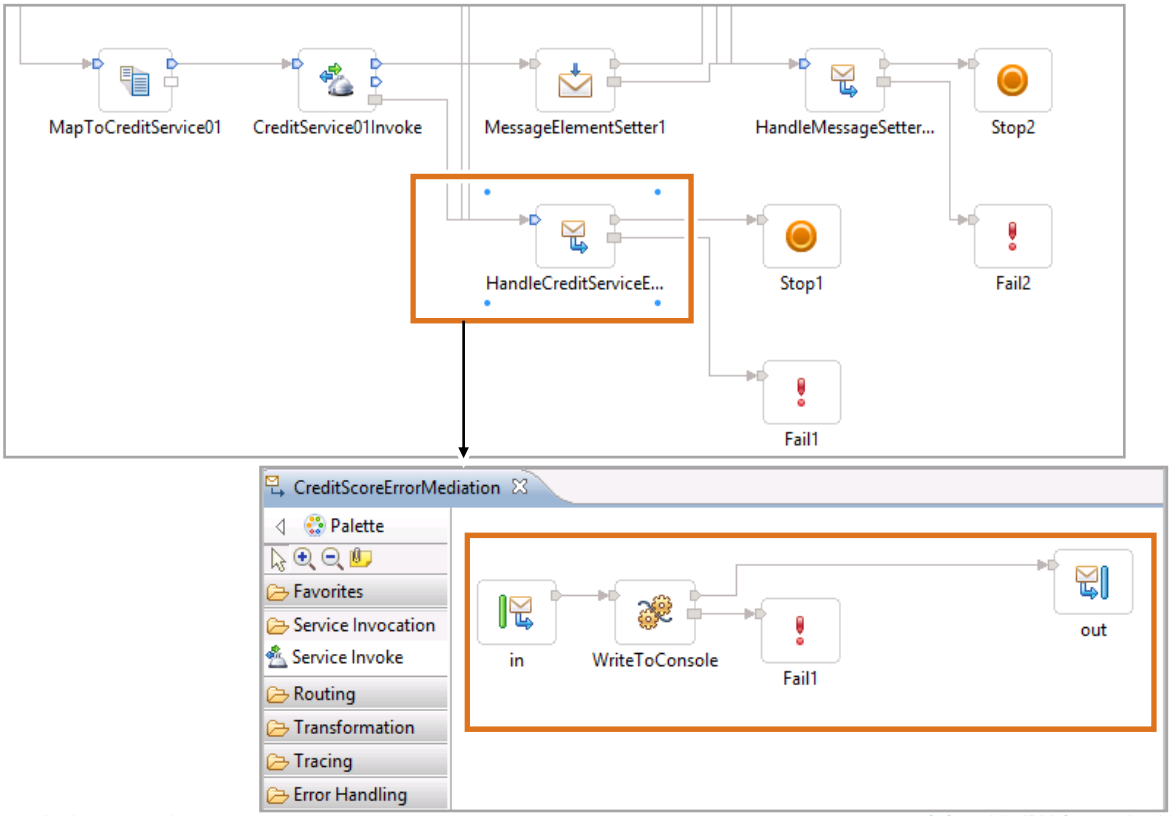
Figure 2-60. Components that are required for Exercise 11

© Copyright IBM Corp. 2017

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

IBM Training

Implement the CreditScoreErrorMediation subflow in Ex 11



© Copyright IBM Corporation 2017

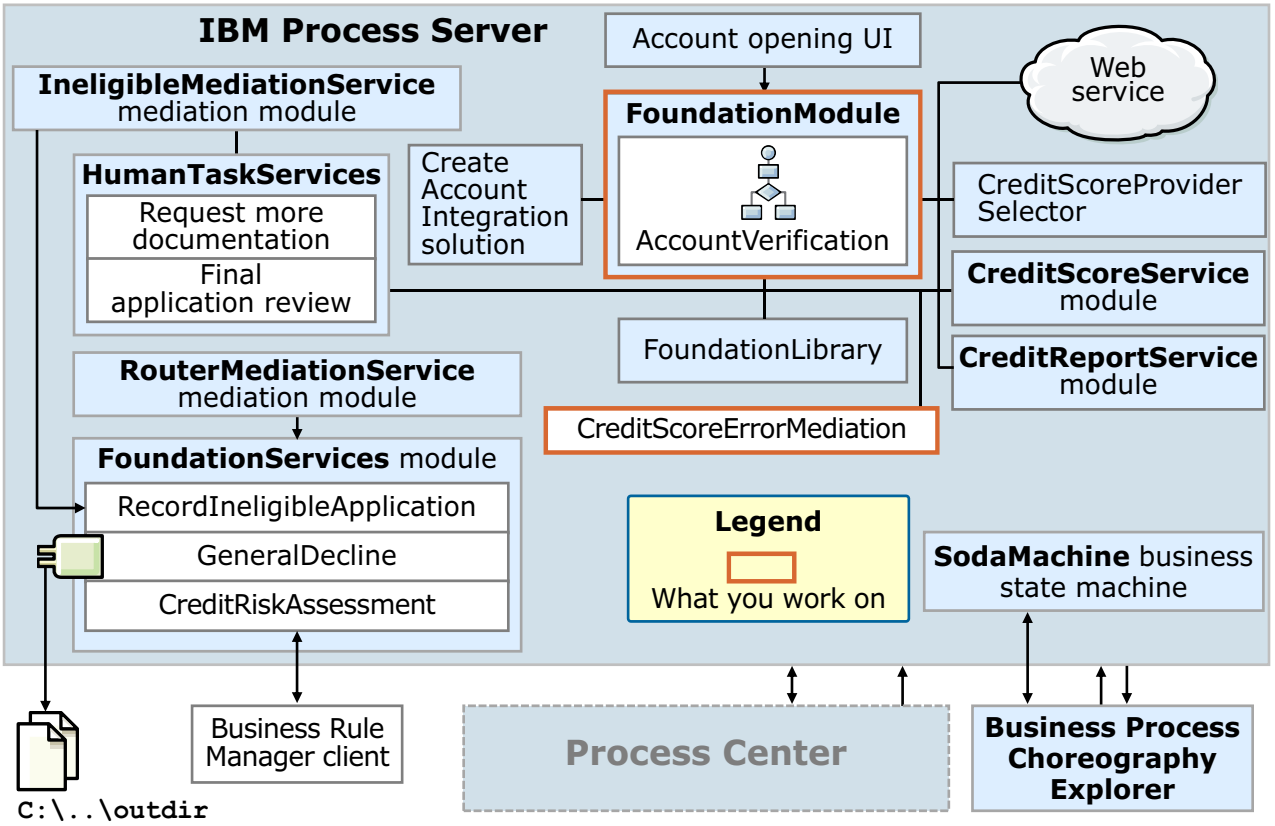
Figure 2-61. Implement the CreditScoreErrorMediation subflow in Ex 11

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 11



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-62. Course business scenario: What you work with in Ex 11

In this lab, you work with IBM Integration Designer.

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

IBM Training

Exercise 12

Implementing security

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-63. Exercise 12

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

IBM Training

What are some key tasks that you do in Exercise 12?

- Set security properties on your application at development time

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-64. What are some key tasks that you do in Exercise 12?

The purpose of this exercise is to introduce you to security settings, not to cover every possible security setting in detail. The settings that you configure here are emblematic of the types of settings you encounter in production applications.

© Copyright IBM Corp. 2017

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

IBM Training

Components that are required for Exercise 12

Prebuilt components that are imported in this lab:

- You do not import any object in this exercise
- You work with the existing infrastructure, and the tools that are available in IBM Business Process Manager
- You work with:
  - **Administrative console**
  - **Process Portal**
  - **Process Admin console**
  - **Process Designer**
  - **Process Center**
  - **IBM Integration Designer**
  - **BPC Explorer**

New components that you create in this lab:

1. Create users and groups

© Copyright IBM Corporation 2017

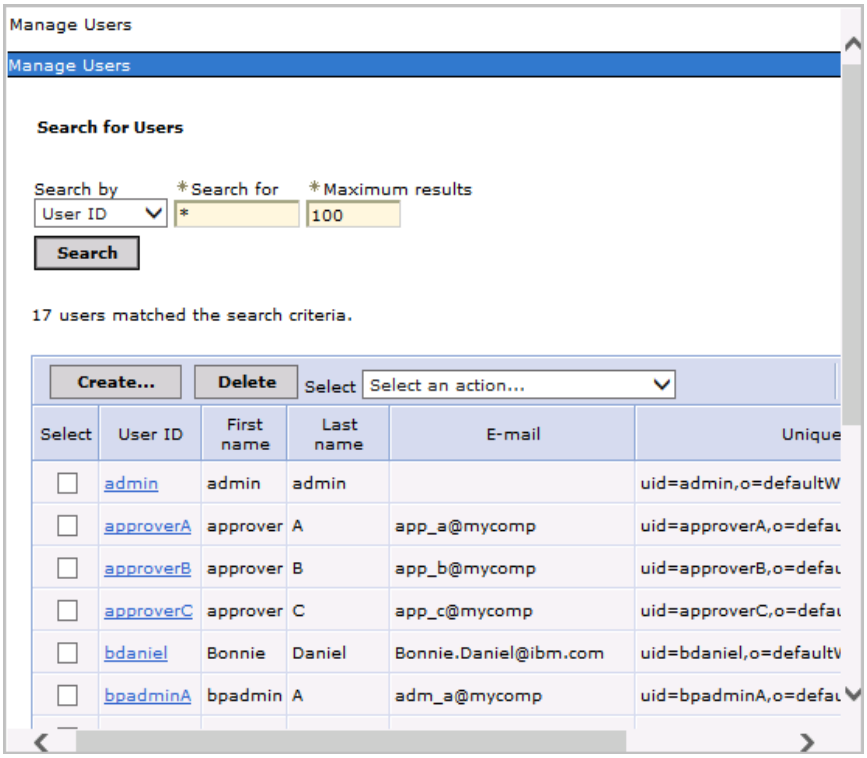
Figure 2-65. Components that are required for Exercise 12

© Copyright IBM Corp. 2017

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

IBM Training

Create users and groups in administrative console in Exercise 12



Course business scenario © Copyright IBM Corporation 2017

Figure 2-66. Create users and groups in administrative console in Exercise 12

In this part of the exercise, you run a customized script that creates several users and groups that you use to test the components in this exercise.

© Copyright IBM Corp. 2017

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

IBM Training

Manage application security by using security qualifier in Exercise 12

	Security	
Component	Security permission	Security identity
AccountVerificationExport		
AccountVerification		
Implementation		
Interfaces		
AccountVerification		
InputCriterion	approvers	
References		
CreditCheckServicePartner		
FinalApplicationReviewPartner		

© Copyright IBM Corporation 2017

Figure 2-67. Manage application security by using security qualifier in Ex 12

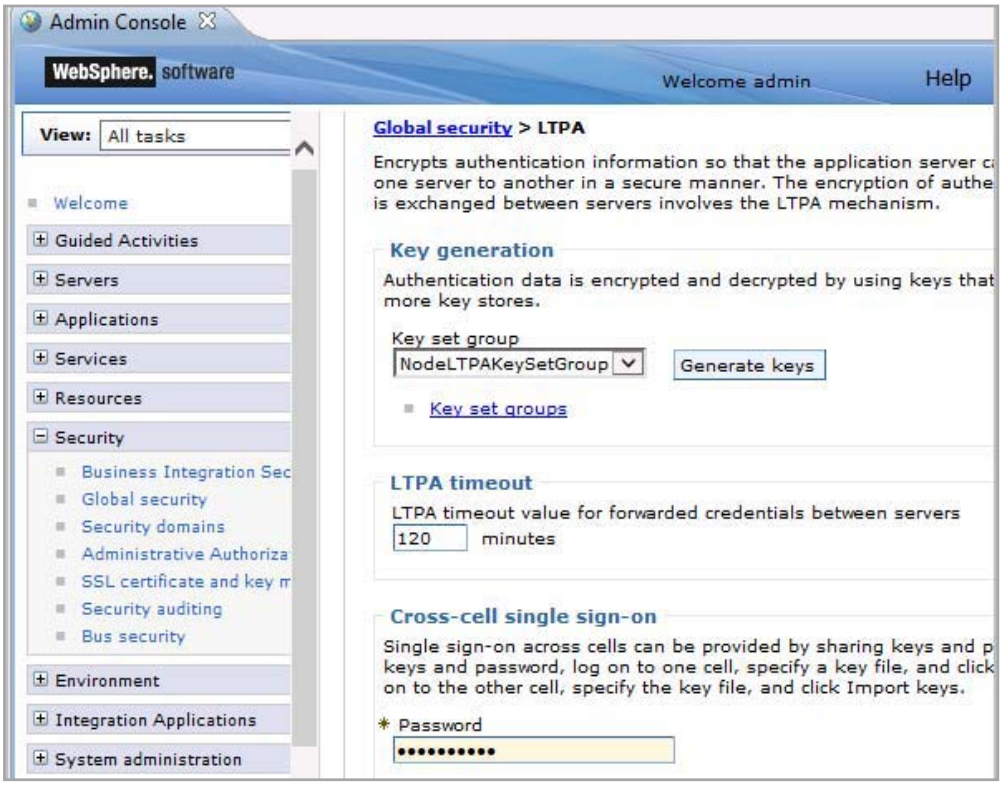
You add the group **approvers** to the **Security permission** qualifier for the **InputCriterion** operation on the **AccountVerification** interface. This addition means that only members of the approvers group have access to start a process instance by starting the InputCriterion operation on the process interface. These steps are about enabling application security.

© Copyright IBM Corp. 2017

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

IBM Training

Create LTPA encryption by using global security in Exercise 12 (1 of 2)



Course business scenario © Copyright IBM Corporation 2017

Figure 2-68. Create LTPA encryption by using global security in Exercise 12 (1 of 2)

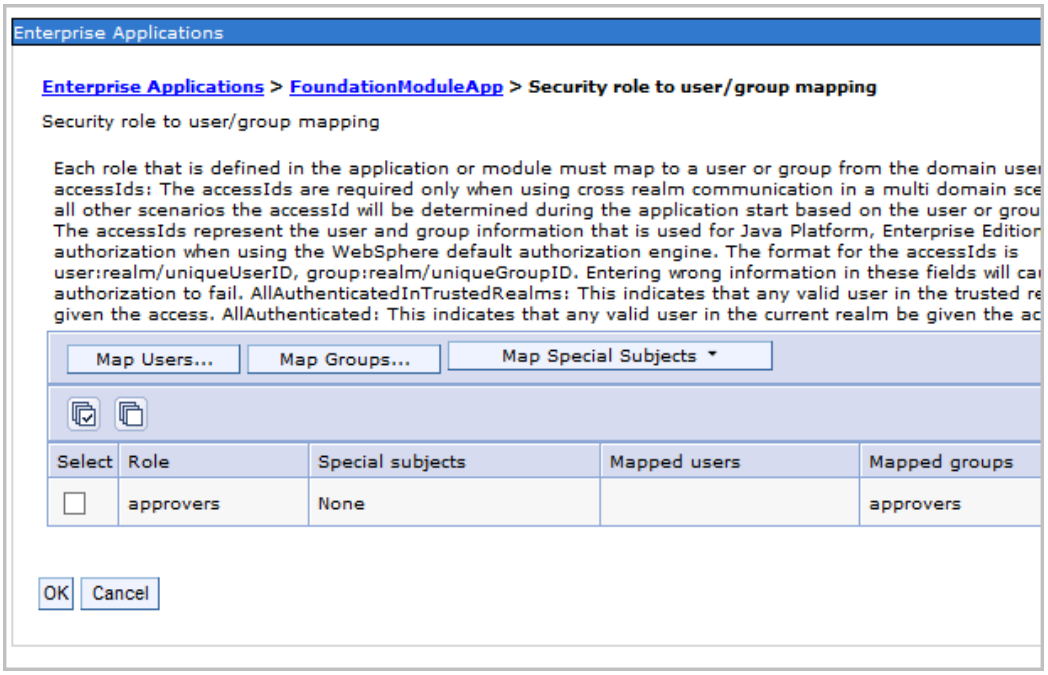
To enforce the permissions for tasks, you must enable security on IBM Process Server. The steps necessary for enabling security on IBM Process Server are identical to the steps for enabling security on WebSphere Application Server. These steps are also identical for turning on security in the WebSphere test environment or a separate installation of IBM Process Server. Here, you create the LTPA encryption key, which is used to establish the authentication token.

© Copyright IBM Corp. 2017

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

IBM Training

Create LTPA encryption by using global security in Exercise 12 (2 of 2)



Course business scenario © Copyright IBM Corporation 2017

Figure 2-69. Create LTPA encryption by using global security in Exercise 12 (2 of 2)

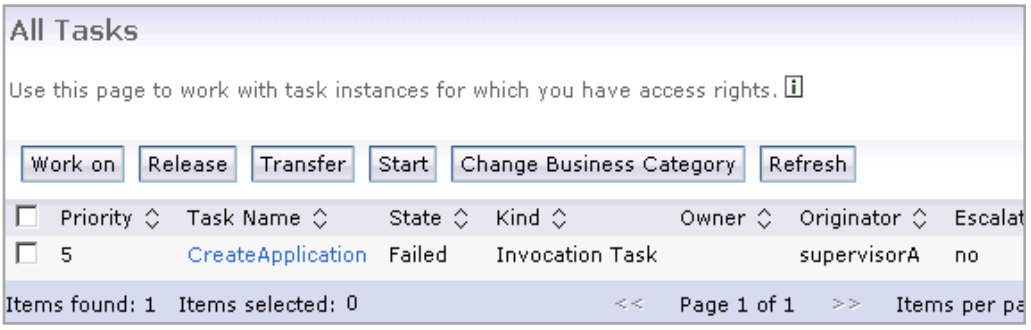
You assign the roles to the **AccountVerification** business process. Those assignments are role names, not the actual user or group names. For security to work, all of the applications that contain roles must map the roles to a user or group. You do that in this portion of the exercise.

© Copyright IBM Corp. 2017

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

IBM Training

Test task security in Business Process Choreographer Explorer in Exercise 12 (1 of 2)



© Copyright IBM Corporation 2017

Figure 2-70. Test task security in Business Process Choreographer Explorer in Exercise 12 (1 of 2)

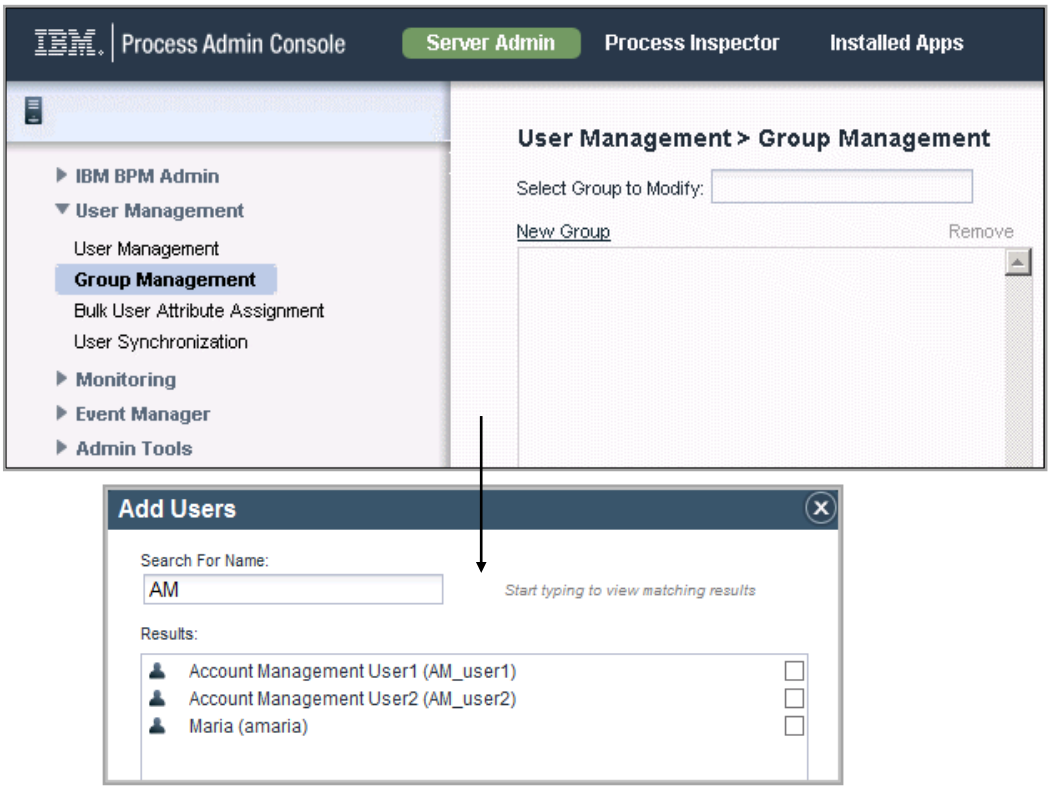
You create an instance of the **AccountVerification** process through the AccountOpeningUI human task user interface. You log in to the Business Process Choreographer Explorer client with the superVisorA user. Since this user is not from an approvers role for which security permission QoS was defined, it does not allow you to create an instance.

© Copyright IBM Corp. 2017

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

IBM Training

Test task security in Business Process Choreographer Explorer in Exercise 12 (2 of 2)



Course business scenario © Copyright IBM Corporation 2017

Figure 2-71. Test task security in Business Process Choreographer Explorer in Exercise 12 (2 of 2)

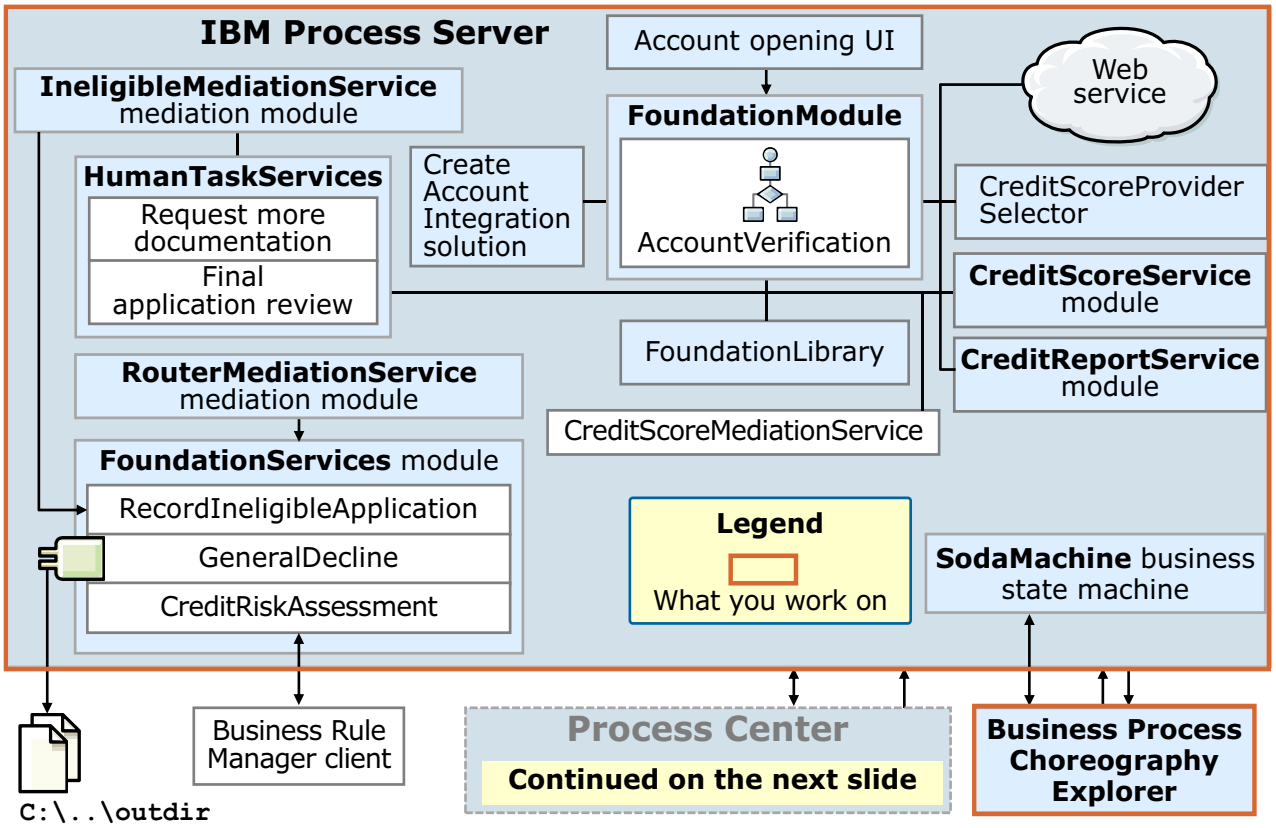
In this portion of the exercise, you experience how to create users and groups, and assign them to a participant group. You further learn how to configure a participant lane to use the created participant group. With this configuration, you enable security to all the activities in the participant lane so that only users of that participant group can participate in the activities of that lane in the business process definition.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 12 (1 of 2)



Course business scenario

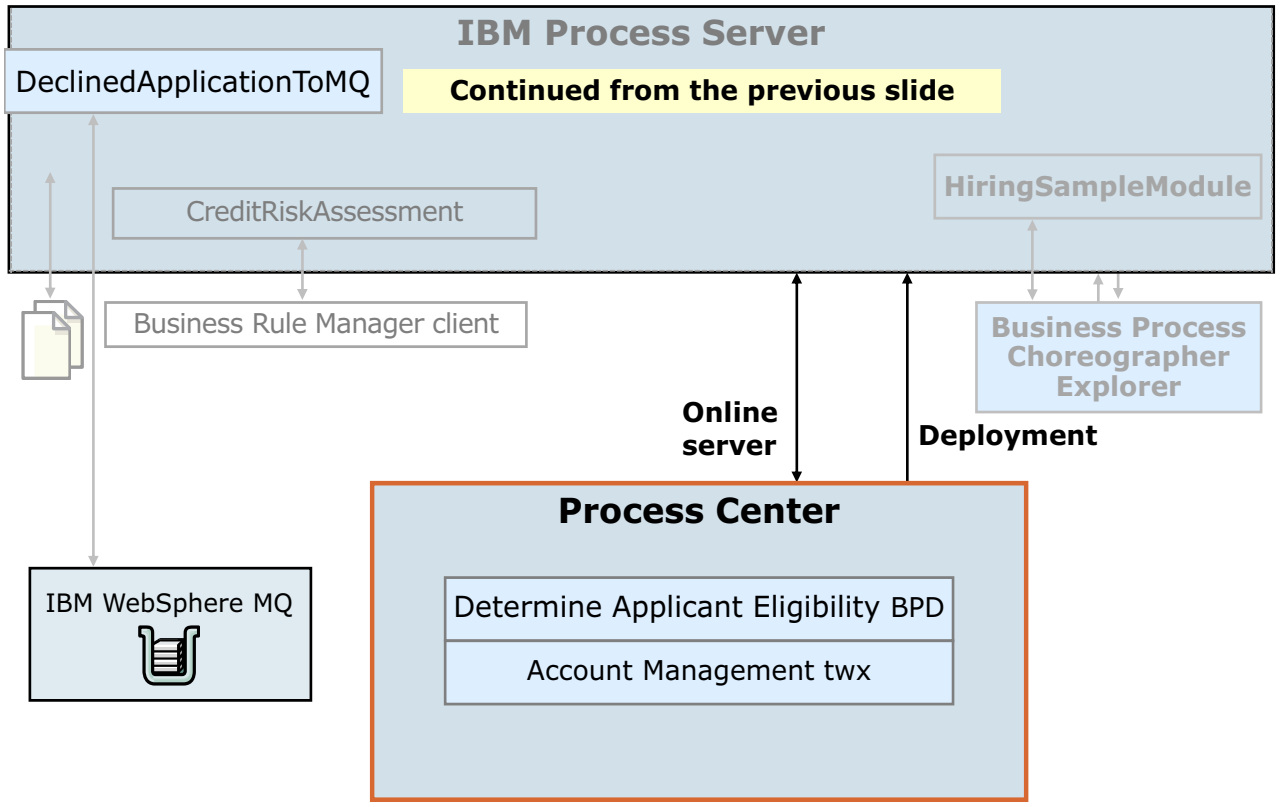
© Copyright IBM Corporation 2017

Figure 2-72. Course business scenario: What you work with in Ex 12 (1 of 2)

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

IBM Training

Course business scenario: What you work with in Ex 12 (2 of 2)



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-73. Course business scenario: What you work with in Ex 12 (2 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 13

Applying governance to process applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-74. Exercise 13

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

IBM Training

What are some key tasks that you do in Exercise 13?

- In this exercise, you explore the sample governance process application, and enable governance to the process application to control its installation on the IBM Process Server
- Finally, you test the installation of a snapshot that goes through the governance process

© Copyright IBM Corporation 2017

Figure 2-75. What are some key tasks that you do in Exercise 13?

© Copyright IBM Corp. 2017

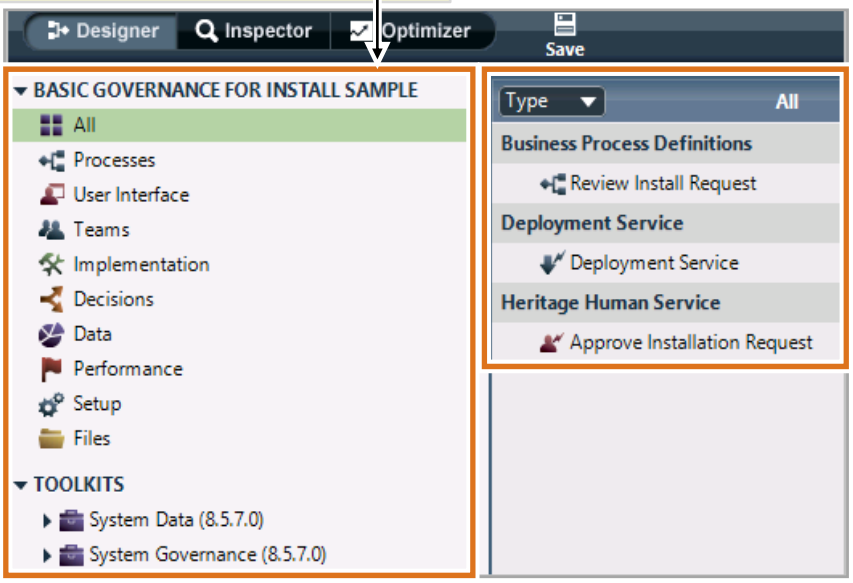
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 13

Prebuilt components that are imported in the lab:

1. **Basic Governance for Install Sample (BGIS)**
2. **System Governance toolkit (TWSYSG)**
3. **Basic_Governance_for_Install_Sample - v1**



New components that you create in the lab:

None

© Copyright IBM Corporation 2017

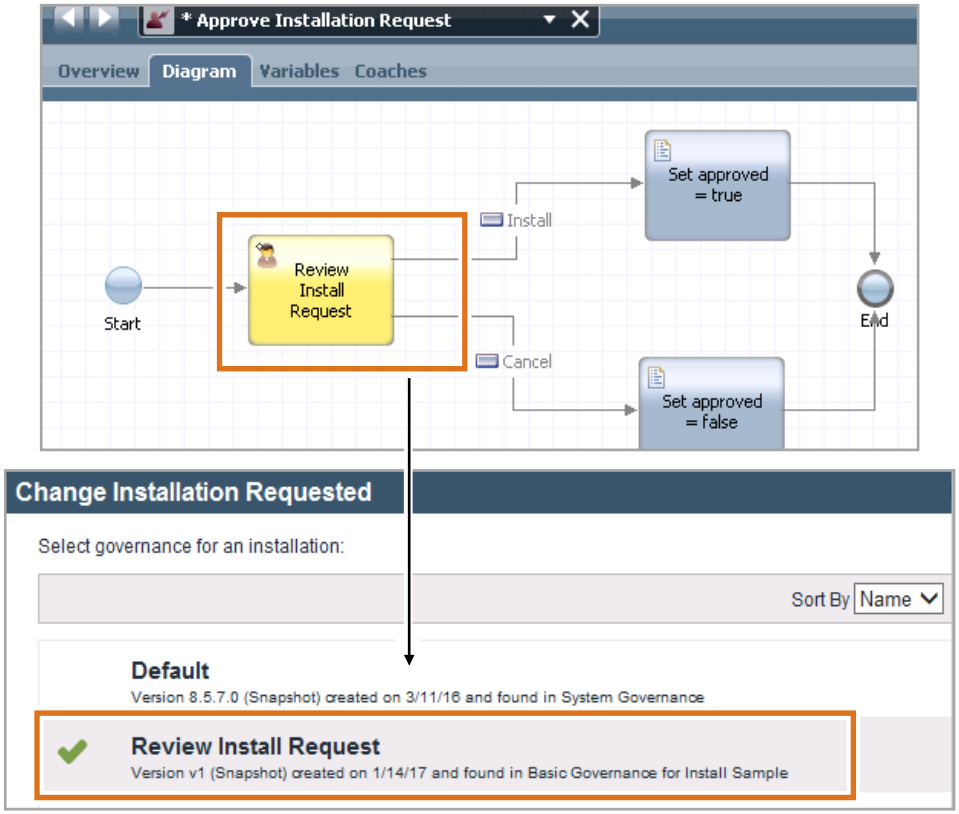
Figure 2-76. Components that are required for Exercise 13

© Copyright IBM Corp. 2017

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

IBM Training

Test governance template in Process Center in Ex 13



© Copyright IBM Corporation 2017

Figure 2-77. Test governance template in Process Center in Ex 13

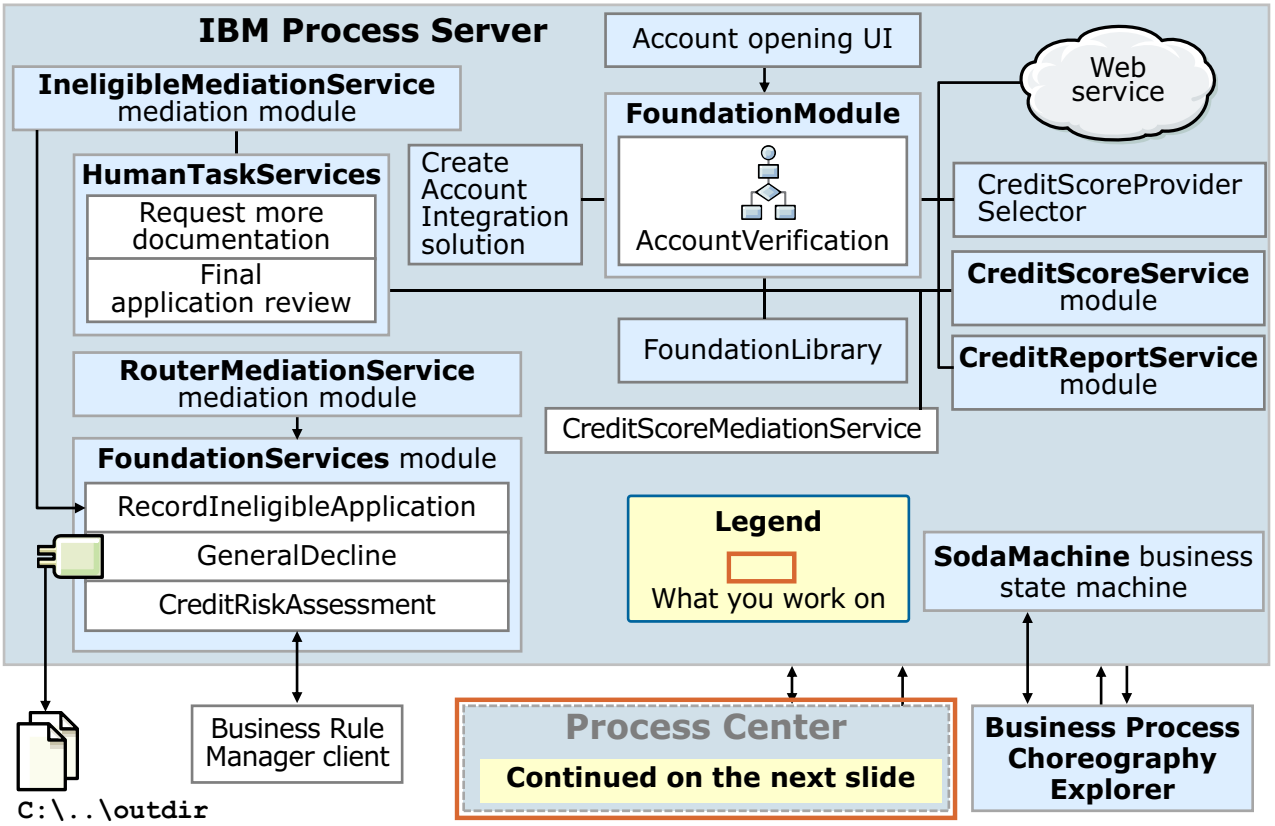
The BPD describes different activities that are involved in the installation approval process. You go through the implementation of each activity in rest of the lab material.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 13 (1 of 2)



Course business scenario

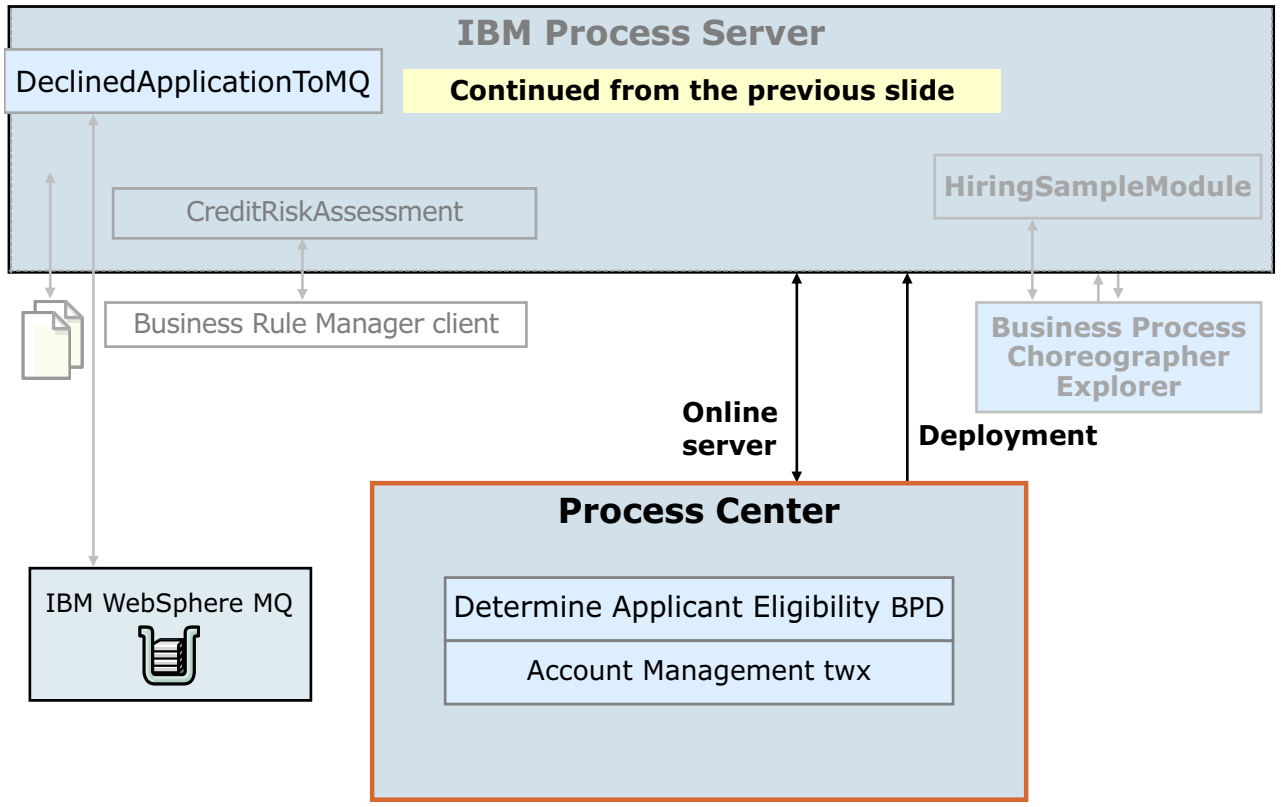
© Copyright IBM Corporation 2017

Figure 2-78. Course business scenario: What you work with in Ex 13 (1 of 2)

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

IBM Training

Course business scenario: What you work with in Ex 13 (2 of 2)



Course business scenario

© Copyright IBM Corporation 2017

Figure 2-79. Course business scenario: What you work with in Ex 13 (2 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Exercise 14

Integrating other applications with IBM Integration Designer

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 2-80. Exercise 14

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

IBM Training

What are some key tasks that you do in Exercise 14?

- Explore the procedures for integrating IBM Business Process Manager assets with other applications, such as IBM Business Monitor

Course business scenario

© Copyright IBM Corporation 2017

Figure 2-81. What are some key tasks that you do in Exercise 14?

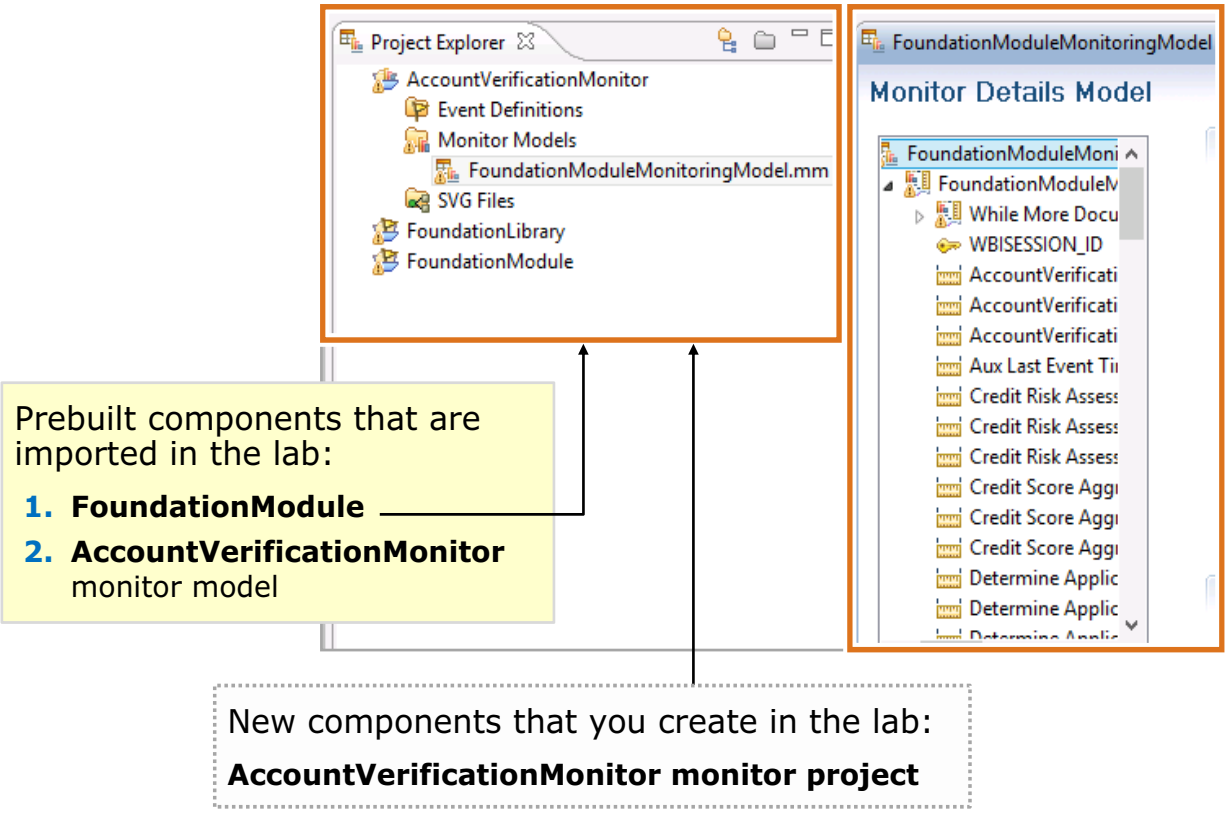
In this exercise, you do the steps that are necessary to integrate IBM Business Process Manager with other IBM products, such as IBM Business Monitor. Because this product is not installed in the lab environment, an effective test is not possible. However, the steps that you do in this exercise demonstrate the steps that are required and the appropriate practices for implementing IBM Business Process Manager in a greater solution.

© Copyright IBM Corp. 2017

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

IBM Training

Components that are required for Exercise 14



Course business scenario © Copyright IBM Corporation 2017

Figure 2-82. Components that are required for Exercise 14

This exercise is not intended to teach you how to build a complete monitor model with key performance indicators. It is intended to demonstrate how tools in IBM Integration Designer can be used to generate monitor models. It is also intended to demonstrate good practices when you integrate IBM Integration Designer with IBM Business Monitor.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe the key components that you build and assemble in this course
- Explain the benefits of each lab exercise in this course

© Copyright IBM Corporation 2017

Figure 2-83. Unit summary

© Copyright IBM Corp. 2017

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

Unit 3. Version control and testing of SCA applications

Estimated time

01:30

Overview

This unit introduces module and library version control in IBM Process Server, and describes strategies for testing SCA applications.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe version control schemes in IBM Process Server
- Identify the IBM Process Server components that provide version control
- Compare version control in IBM Process Server to snapshots in IBM Process Center
- Describe some of the version control considerations for SCA applications
- Describe the IBM Integration Designer tools for debugging applications
- Describe the advanced testing facilities that are available in IBM Integration Designer
- Explain the purpose of the IBM Integration Designer Server Logs view
- Explain the business value of the IBM Support Assistant

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Understanding version control in SCA components
- Considerations during version control
- Cross-component trace capabilities and settings
- Testing and logging with the Component Test Explorer

© Copyright IBM Corporation 2017

Figure 3-2. Topics

© Copyright IBM Corp. 2017

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

3.1. Understanding version control in SCA components

© Copyright IBM Corp. 2017

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

IBM Training

Understanding version control in SCA components

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-3. Understanding version control in SCA components

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

IBM Training

What is version control?

- Version control is a critical component of your application maintenance strategy
- Facilitates production evolution and the management of services that are running in an SCA production environment
- Some common questions that are addressed when implementing version control with SCA applications are:
  - What happens when changes are made to the SCA application?
  - What behavior is expected after the change?
  - Do components communicate with the correct version of the application?
  - How does one handle the change and how is that change made visible to users?

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-4. What is version control?

The importance of version control becomes apparent when one thinks of the future of an SCA application. What happens after you change assets? In the SCA environment, each application is both a consumer and a provider of services. If a person changes the logic to support a business need, what does that mean for those components that depended on it? Do they still expect the old behavior? What about business processes that were “in-flight” when the assets changed? The person who started that process probably did not expect its logic to change halfway through. How do you handle these circumstances?

© Copyright IBM Corp. 2017

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

IBM Training

What can have versions in IBM Integration Designer?

- SCA itself has no concept of version control
- Version control is available for modules and libraries
  - Services and exports inherit their version from the module
- The only IBM Process Server components that understand “version control” are:
  - Process
  - Business state machine
  - Imports

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-5. What can be versioned in IBM Integration Designer?

SCA specification does not support version control. Therefore, the wires between service components in assembly diagrams are static. However, the WS-BPEL specification supports process template version control. Thus, you can use late binding between service components if they are BPEL. An SCA module map to a Java EE enterprise application and an application with running processes cannot be modified. Therefore, support for multiple versions of an artifact at run time is more complex than expected. SCA itself has no concept of version control, and the business process, business state machine, and human task components support the concept of version control.

The IBM Integration Designer concept of a library is a development-time concept only in IBM Process Server. When a module with dependencies on a library is deployed, the artifacts are copied from the library into a utility JAR file in the module EAR file. You must be careful when editing artifacts in a library afterward. You must take care when multiple modules depend on the artifacts, since a change to a library component affects all dependent modules. However, the changes become active for an existing module after the module is deployed. The architectural reason for this limitation is that the SCA artifact loader does not currently support version control.

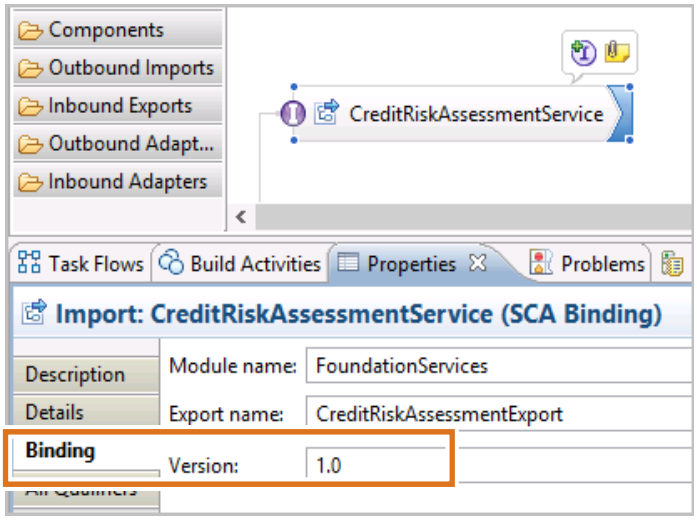
© Copyright IBM Corp. 2017

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

IBM Training

Version control for SCA imports

- If the target version is controlled, SCA import can specify the target module version
- SCA import binding has a matching version value for an SCA export only if it is created by using “drag-and-drop” with an existing SCA export in the workspace



© Copyright IBM Corporation 2017

Figure 3-6. Version control for SCA imports

By using version control, you can create a module and put it into production. You can then decide to create a version of that module and also put that module into production.

SCA bindings inherit their version information from the module they are associated with. When you create an SCA import binding by dragging a version of an SCA export binding, the new SCA import binding is tagged with the version of the export. The static binding resolves to the specific export binding and module version during run time. Although the binding is statically declared, you can modify the SCA import binding at run time. Use the administrative console to reset the version value to use a different deployed version of the export or module. If an export is not known within the workspace, you must be aware of whether the version is assigned to the export and manually assign that version value to the import binding.

The SCA import has a static binding to one particular export, which means an import can establish a connection only to the export for which it was built and configured. If the import has no version declared, it is expected to be bound to a non-versioned export. If multiple numbered versions of the service are deployed to the runtime environment and the import does not have a version declared, the import fails to resolve to any export.

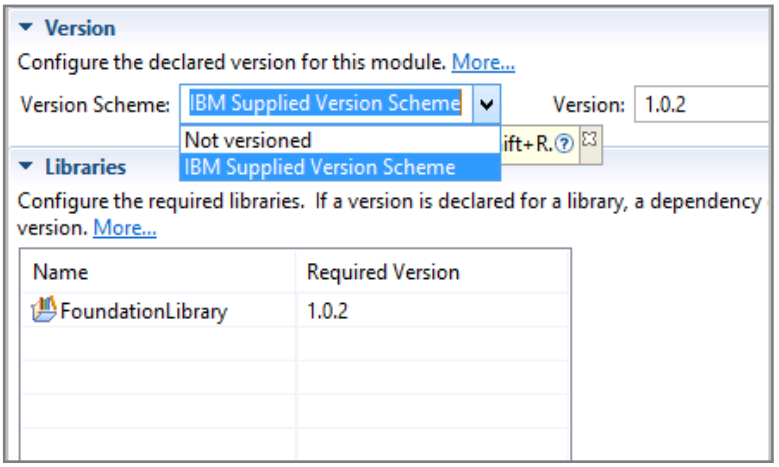
© Copyright IBM Corp. 2017

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

IBM Training

Version control for modules and libraries

- Create versioned modules and libraries for future deployment to a production environment
  - These versions are for use on the server at run time, not for distinguishing levels of completion in a development environment
- Version and version scheme should be set in the *Dependencies* editor
- Services and exports inherit their version from the module
- Export the business integration module by command-line deployment
- **`serviceDeploy`** is used to create a version-aware EAR



© Copyright IBM Corporation 2017

Figure 3-7. Version control for modules and libraries

You can create versioned modules and libraries in IBM Integration Designer for future deployment to a production environment. These versions are for use on the server at run time, not for distinguishing levels of completion in a development environment. Version control of modules and libraries is optional. It can be based on a scheme that IBM provides, or you can choose not to use version control at all. Modules that use a library have a dependency on a specific version of that library, and libraries can also have dependencies on specific versions of other libraries. Services and exports inherit their version from the module.

Version control is beneficial in a production environment for a number of reasons. Version control allows the simultaneous deployment of multiple versions of an SCA module to the same deployment environment. (That is, version control applies to the production environment. It does not apply to the development and testing environment.)

Metadata about module and library version control is stored in the root directory of the project in a file that is called either `sca.module.attributes` or `sca.library.attributes`. The stored data includes the version number (value), the version provider, and the library dependency.

Export a business integration module that specifies command-line service deployment. You can then use the `serviceDeploy` command to generate a version-aware EAR file. Input to the serviceDeploy command is the output archive file from the business integration module export.

© Copyright IBM Corp. 2017

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

You can change the declared version of a module or library and use refactoring to automatically update the target version of all referencing SCA import bindings in all modules and libraries.

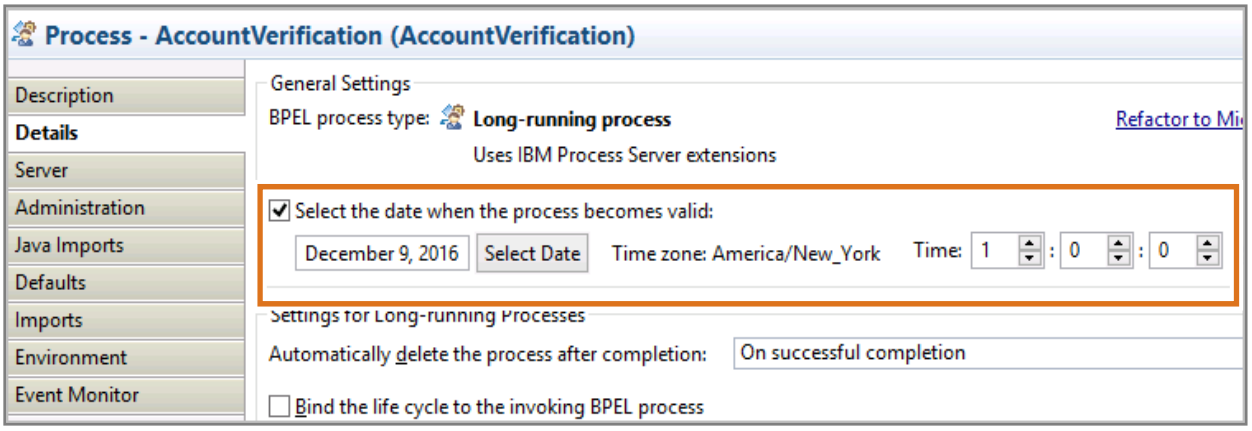
© Copyright IBM Corp. 2017

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

IBM Training

Version control of business process templates (1 of 2)

- In a runtime environment, the following data identifies an instance of a business process and business state machine:
  - Name
  - Namespace
  - Valid-from date
- **Name** and **namespace** can be shared between multiple instances of the same business process
- **Valid-from** dates are used to differentiate multiple instances of the same business process



© Copyright IBM Corporation 2017

Figure 3-8. Version control of business process templates (1 of 2)

The runtime engine can start a new instance of a business process that is set to become valid today, even if an older instance of the same process is still active. The new instance can have information that is not relevant to the previous version so that the previous version is allowed to continue its execution unchanged.

Calendar values are represented in Coordinated Universal Time (UTC).

© Copyright IBM Corp. 2017

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

IBM Training

Version control of business process templates (2 of 2)

- Long-running components can be changed to include new activities and new logic
- Instances started with old versions need to continue on existing templates
- New versions of updated components can be created for new instances by specifying a more recent valid date
- New instances are started with the new template
- Remove the old template only when all instances that were started with the old template are complete

© Copyright IBM Corporation 2017

Figure 3-9. Version control of business process templates (2 of 2)

The key point here is that only the Business Process Choreographer engine understands version control. The rest of the runtime is oblivious. When you install a “new version” of a process, the IBM Process Server runtime sees it as a brand new module with a new BPEL component inside. You install the new version by installing a new application with a process of the same name. The Business Process Choreographer engine is the only thing that realizes this BPEL component is a version of a pre-existing BPEL component.

The valid-from attribute is used in late binding scenarios to determine which version of a process, human task, or business state machine can be used as the template for a new instance. At a specific time, the runtime uses the valid-from date to determine which template to use to create new instances. The runtime uses the version with the valid-from date closest to the current date, provided it is not in the future. For example, assume that multiple versions of a process are deployed and that the date is 1:00 PM on Thursday, December 9, 2016. The version with a valid-from date closest to the date and time and not in the future is used to instantiate any instances of this process.

This qualifier is called *late binding*.

If two artifacts have the same fully qualified name, they are versions of each other.

© Copyright IBM Corp. 2017

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

IBM Training

Running instances with the old version

- Create a version of the process whose binding can be dynamically resolved in the runtime environment
- You do not want the running instances of the process to adopt the changes included in the new version
- Create a module where the process version is contained
- Must specify the date and time (*validFrom*) when the new process version becomes valid
- For clients that use late binding to start this process, the runtime engine recognizes whether the new process is in fact a more recent version
  - The runtime engine selects the new process instead of the older one

© Copyright IBM Corporation 2017

Figure 3-10. Running instances with the old version

Create a version of your process whose binding can be dynamically resolved in the runtime environment. This procedure does not include the generation of a migration specification, which would be necessary to move running instances of the process to the new version.

You must deploy and install a long-running business process to your runtime environment. You decide that you want to create a version of the process, and you do not want the running instances of the process to adopt the changes included in the new version. These steps assume that you already created a business process for which you seek to create a version. If your process contains compensations, or if you want the running instances of the process to continue to use the older version, then follow these instructions.

© Copyright IBM Corp. 2017

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

IBM Training

Process instance migration

- When would you use process instance migration?
  - You want to create a version of the process
  - You would like running instances to adopt the changes included in the new version
- Must define a process migration specification
- Install the new process version and the process migration specification to your runtime environment
- Allows upgrade to a newer version by creating a process migration specification
  - Create a version of the process that can be used as the target of a process instance migration in the runtime environment
  - Upgrade running instances
  - Must deploy and install a long-running business process to your runtime environment

© Copyright IBM Corporation 2017

Figure 3-11. Process instance migration

With process instance migration, you can move running instances to a newer version by creating a process migration specification. This topic explains how to create a version of your process that you can use as the target of a process instance migration in the runtime environment.

You must deploy and install a long-running business process to your runtime environment. You decide that you want to create a version of the process, and you want the running instances of the process to adopt the changes included in the new version.

Note: A running instance that moves to a newer version depends on the state of that instance. New instances of a process can use the newest version.

To create a process version that allows for instance migrations at run time, you must define a process migration specification. You deploy and install the new process version and the process migration specification to your runtime environment. An administrator can then move existing instances of the process in the runtime to the new version of the process.

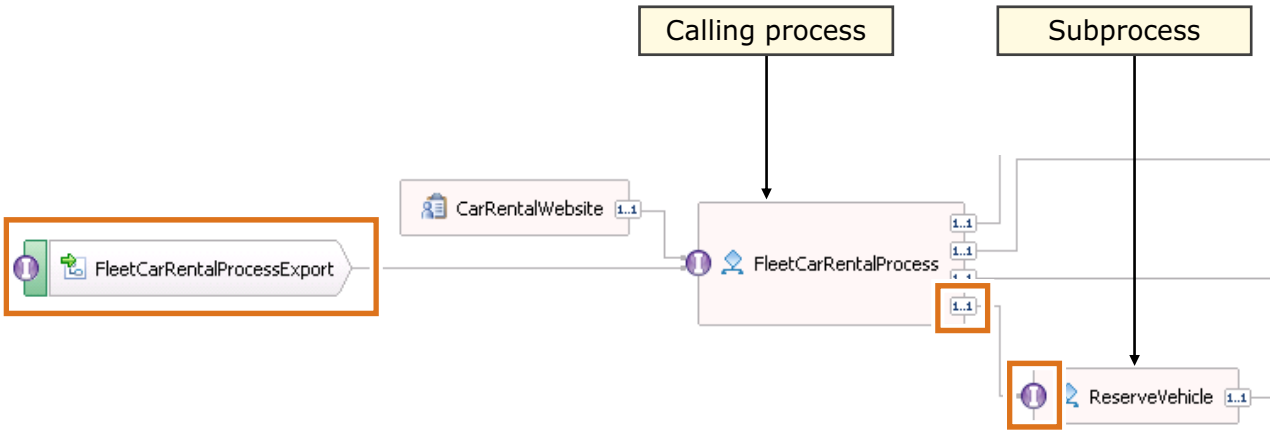
© Copyright IBM Corp. 2017

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

IBM Training

Early binding (static binding)

- Business Process Choreographer API expects **valid-from** date and time
- By default, SCA uses static binding that ignores process version control
- JSP along with a stand-alone reference in a module always starts the process in that module
- Exports from process module versions allow for SCA-level version control



Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-12. Early binding (static binding)

Consider an SCA module that contains a process. If any component is wired to the process, that version of the process is always used to instantiate any necessary process instances. That is, SCA wires are early bound to the version to which they are wired. If a second version of the process is deployed, you have an interesting situation. If you start the process through an SCA wire from either module, you get the associated version.

© Copyright IBM Corp. 2017

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

IBM Training

Long-running process dynamicity: Late binding (1 of 2)

- Consider possible situations where:
  - Process templates that are no longer current might still be valid for existing long-running process instances
  - Current process templates are used for new process instances
  - Process templates become valid in the future according to their valid-from date and time
- An important pattern exists to help with this problem:
  - **Parent-child** processes with **late binding**
  - Allows behavior of long-running process to be changed while its instances are running
  - Parent calls shorter-running child processes to start key steps of a process

© Copyright IBM Corporation 2017

Figure 3-13. Long-running process dynamicity: Late binding (1 of 2)

When you late bind one process to another one, the connection is resolved dynamically in the runtime environment. In that situation, the calling process always picks up the currently valid version of the process that it is starting. New instances always use the newest version of the process, and if you use the migration verification tools, you can also move running instances to the newest version.

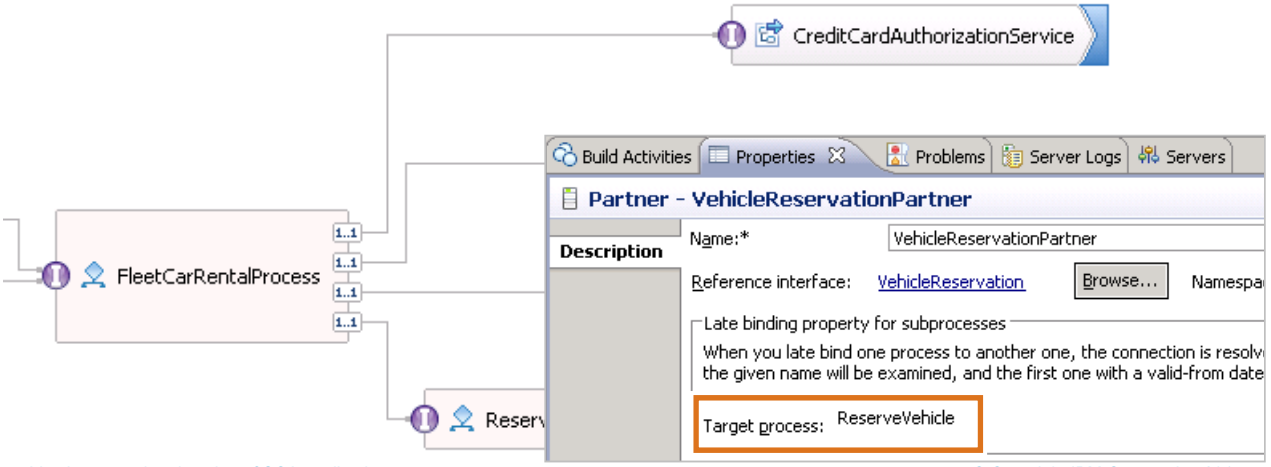
© Copyright IBM Corp. 2017

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

IBM Training

Long-running process dynamicity: Late binding (2 of 2)

- Parent-child late-binding pattern
  - Parent process runs for the length of the whole process
  - Parent contains minimal logic
  - Parent calls shorter-running child processes to start the key steps of a process
  - Late binding of child processes is used
  - Allows behavior of long-running process to be changed while it runs



© Copyright IBM Corporation 2017

Figure 3-14. Long-running process dynamicity: Late binding (2 of 2)

The parent process is so generic that it never needs to be changed. When it calls a child process to do some work, the runtime is configured to use late binding such that the most recent version of the child is picked up. Thus, the apparent logic of an overall process instance can change after it is created.

Wiring a process starts another process. This feature gives you early-binding semantics. Explicitly naming a process template on the partnerLink tells the runtime that processes implement any invocations of this partner and that the binding is late.

If you wire a reference on a process component that is configured for late binding, you get a warning since the SCA wiring target is never started; the late-bound process function runs instead.

In long-running processes, it is observed that a process instance probably spends most of its time “asleep,” waking occasionally upon the receipt of an event or invocation of an API. If the design of the process can change over the period of its life, an option can be to remodel it as a collection of short-lived processes. Each shorter-lived process can either trigger the next process immediately or register with the WebSphere Application Server Scheduling Service. Triggering the process immediately presents an opportunity for late binding to choose the most recent version. Registering the process allows WebSphere Application Server to start the next stage of the process at some suitable time in the future. Although these processes run for a shorter length of time than in the

© Copyright IBM Corp. 2017

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

overarching case, they are still likely to be full-fledged long-running processes since these processes are needed to support human and asynchronous interactions.

© Copyright IBM Corp. 2017

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

3.2. Considerations during version control

© Copyright IBM Corp. 2017

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

IBM Training

Considerations during version control

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-15. Considerations during version control

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Considerations when using version control

- Version control is optional, and by default, versions are not created for modules and libraries
- Dependent projects can have mixed version and nonversion states
- Business processes can be deployed directly to IBM Process Server environment for testing, without requiring deployment by the IT team
- The IBM supplied version scheme uses major, minor, and service numeric components in the format ***<major>.<minor>.<service>*** where
  - ***<major>*** indicates an increment for incompatible changes
  - ***<minor>*** indicates an increment for compatible changes
  - ***<service>*** indicates an increment for bug fixes
  - Example: **`FoundationModule V 1.1.2`**

Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-16. Considerations when using version control

To determine whether newly created modules and libraries have version control by default, go to **Window > Preferences > Business Integration > Module And Library Versions** and select the **IBM Supplied Version Scheme** from the list. Version control is now turned on as the default.

Versioned modules that are exported as EAR files are ***not*** version-aware. However, notice that the integration module exporter prevents you from exporting versioned modules as EAR files.

© Copyright IBM Corp. 2017

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

IBM Training

Considerations for deploying versioned modules

- IBM Process Server supports the following scenarios:
  - Installing a versioned module to a server or cluster in a cell
  - Installing the same version of a module one time for each of one or more servers or clusters in a cell
  - Installing different versions of a module on the same server or cluster
- Deploying a new version of a module does not replace previous versions of the module
- If you update applications without changing versions, those updated applications and artifacts replace existing applications and artifacts except for the security policies
- To preserve version control information, the installation process automatically changes the module name through the **`serviceDeploy`** or **`createVersionedSCAModule`** command:

  `moduleName_versionValue_uniqueCellID`

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-17. Considerations for deploying versioned modules

Previous versions of cell-scoped application artifacts (in this case, business rules) are overwritten. All security policy artifacts are preserved during an application update.

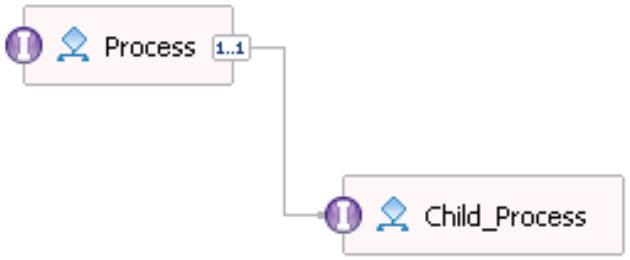
© Copyright IBM Corp. 2017

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

IBM Training

Late binding consideration

- In late binding scenarios, you must avoid changing the new processes that cause compatibility issues when the new version becomes valid:
  - Modifying the correlation sets
  - Changing any interface that the parent process uses to communicate with the subprocess
- "No overarching process" pattern
  - Decompose process into several shorter processes with "hand-off" points
  - The final step of each process step is to instantiate the next "step" of the process



© Copyright IBM Corporation 2017

Figure 3-18. Late binding consideration

© Copyright IBM Corp. 2017

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

IBM Training

Intelligent packing decisions

When making packaging decisions, consider how to avoid future implementation refactoring

- Refactoring can be time-consuming, especially when components (BPEL, Mediation, Java) or adapters are involved
- Package new Advanced Integration services in Process App if it is “a one of” private service
- Package artifacts in a Toolkit to promote component reuse
- Package-related Advanced Integration services in the same Toolkit
  - This approach is similar to having operations on the same interface
- Always use the refactoring tool when moving implementations

© Copyright IBM Corporation 2017

Figure 3-19. Intelligent packing decisions

© Copyright IBM Corp. 2017

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

IBM Training

Using source control management (SCM)

- It is possible to use SCM and IBM Process Center concurrently
- Ensure that you first commit your project to IBM Process Center, and then share it with SCM
  - The `bpm.rep.prefs` file is generated when you commit to IBM Process Center
  - This file must be saved in SCM, or your project loses association with IBM Process Center
- When starting a new workspace, get your source code from SCM, not from IBM Process Center
  - IBM Process Center does not save the SCM metadata files
- Associate IBM Process Center snapshots with SCM versions
  - To revive a snapshot, create a track in IBM Process Center and branches on a version in CVS
  - It is a good idea to use the same name for SCM versions and IBM Process Center snapshots

Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-20. Using source control management (SCM)

It is a good practice to use SCM with IBM Integration Designer and IBM Process Center. You might have to use SCM and IBM Process Center for IBM Integration Designer artifacts. Corporate standards might require that you continue to use a specific SCM, or you might be coming to IBM BPM from heritage IBM Process Server. SCM contains a history that is important for maintaining code integrity. Automated build scripts run from the SCM. Working with two repositories introduces more complexity to the governance system, but careful planning can address those difficulties.

These instructions help you manage that configuration successfully. Although this topic addresses working with Concurrent Versions System (CVS), the principles also apply to other SCM systems. For more information about working with CVS or other SCM systems, see the related links at the end of this topic. You can access actions for source control management system clients that you installed from the IBM Integration Designer Business Integration view by right-clicking a project and opening the Team menu.

Eclipse allows one SCM to be associated with a project. If you use IBM Process Center with another SCM system, treat IBM Process Center as a secondary repository. The SCM system is the primary storage area that developers use as they work with their artifacts. Push projects to the IBM Process Center intermittently, perhaps at project milestones. Do not check projects out from IBM Process Center to change assets. Because you work from CVS, artifacts such as business process definitions, which are solely associated with IBM Process Center, do not exist in the CVS repository.

© Copyright IBM Corp. 2017

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

3.3. Cross-component trace capabilities and settings

© Copyright IBM Corp. 2017

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

IBM Training

Cross-component trace capabilities and settings

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-21. Cross-component trace capabilities and settings

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

IBM Training

Cross-component trace capabilities

- Cross-component trace (XCT) maps **`SystemOut.log`** and **`trace.log`** records back to the SCA programming model
- Supports all SCA call patterns
  - Asynchronous one-way
  - Asynchronous with callback
  - Asynchronous with deferred response
  - Synchronous call
- Works with business objects, or simple data types
- Supports a network deployment (ND) environment
  - Load files from each server
  - Might load only some of the files from all the servers
- Create an IBM Integration Designer Test Client trace from log record data

Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-22. Cross-component trace capabilities

Cross-component trace identifies whether a Service Component Architecture (SCA) operation is successfully completed. It provides information about what modules were used at run time and what service calls were made. Additionally, you can configure cross-component trace with a *data* snapshot to capture data that is sent in and passed between SCA components.

When you enable cross-component trace, invocation records are generated during SCA processing of modules and components. The invocation records help to identify the resulting call chains and to correlate log messages, first-failure data capture (FFDC), and directly logged exceptions to these call chains.

© Copyright IBM Corp. 2017

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

IBM Training

Use cross-component trace for problem determination

- Supports Business Process Execution Language (BPEL) microflows
- Support for long-running business processes
  - Follows flow and correlate log records to long-running processes
  - Supports multiple “pick” and “receive” activities
  - Follows internal thread work
  - Follows callouts to partners
- Supports HTTP, JMS, WebSphere MQ, and IBM MQ JMS bindings
  - Another step toward end-to-end call chains
  - Captures work done as part of export: function selectors and data bindings
- Rolling log support
  - SystemOut and trace files are often split over multiple files
  - Load all the files in for the full call chain

Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-23. Use cross-component trace for problem determination

The separation of these messages along call chains serves two purposes:

- Problem determination with IBM Integration Designer
- A better understanding of the runtime behavior of the system, including an understanding of how the SCA service calls move through the system

The log records that are associated with the IBM Process Server applications hold information about errors or events that occur during processing and can be used for problem determination with IBM Integration Designer.

© Copyright IBM Corp. 2017

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

IBM Training

Settings in cross-component trace

- Disabled
  - No records are generated in the server console and logs
- Enabled
  - Invocation records are generated in the server console, the **`SystemOut.log`** file, and the **`trace.log`** file
  - Record properties do not include any invocation input and output data
- Enabled with data snapshot
  - Invocation records are generated in the server console, the **`SystemOut.log`** file, and the **`trace.log`** file
  - Record properties include invocation input and output data
  - Data is captured in input and output files under the **`logs\xct`** directory

© Copyright IBM Corporation 2017

Figure 3-24. Settings in cross-component trace

Cross-component trace uses the WebSphere Application Server logging service to capture information and to provide context for the other entries in the logs. Cross-component trace captures the information that pertains to SCA component processing (including the sequential flow of the event). This information is written to the `SystemOut.log` or `trace.log` file according to well-defined rules. You can set cross-component trace parameters to write the records to either the `trace.log` or `SystemOut.log`.

`SystemOut.log` and `trace.log` files are located in the server log directory, for example:
`C:\IBM\IID\ps\v8.5\profiles\qbpmaps\logs\server1`

When cross-component trace is enabled with the data snapshot, the input and output files are stored under the `logs\xct` directory, for example:
`C:\IBM\IID\ps\v8.5\profiles\qbpmaps\logs\xct`

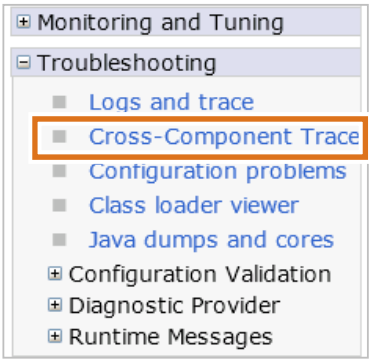
© Copyright IBM Corp. 2017

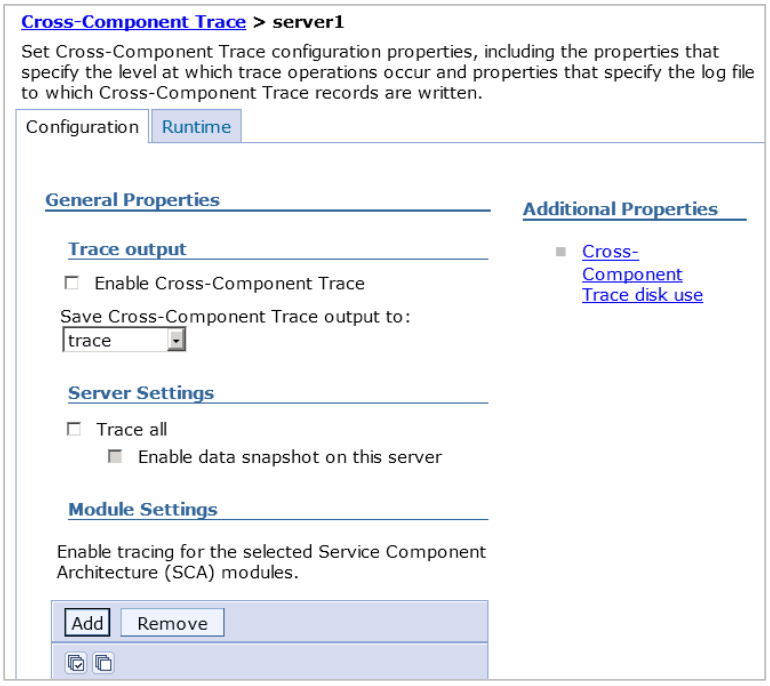
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Activating cross-component trace in production

- Select **Troubleshooting > Cross-Component Trace**
- Select the server to modify; each server is set individually





© Copyright IBM Corporation 2017

Figure 3-25. Activating cross-component trace in production

You can enable and disable cross-component trace from either the administrative console or from the Server Log view in IBM Integration Designer.

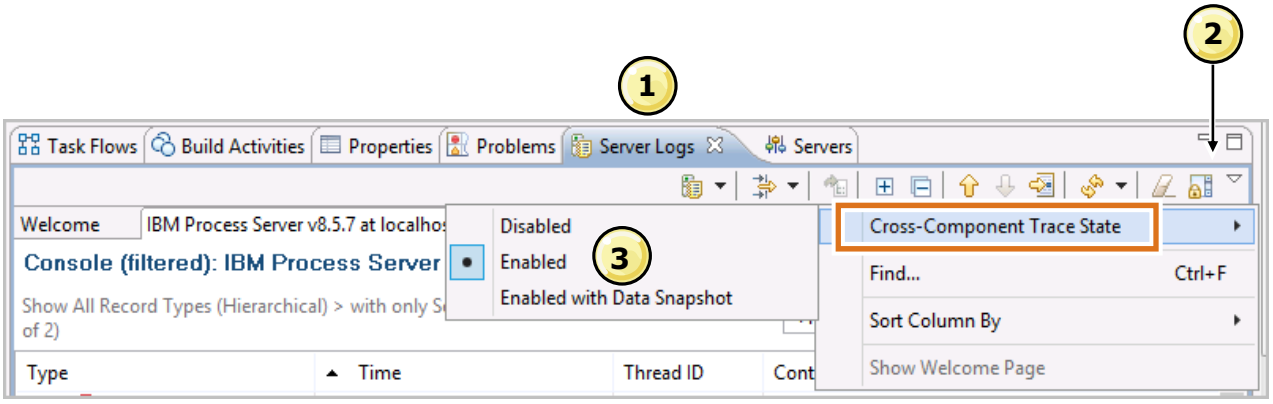
© Copyright IBM Corp. 2017

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

IBM Training

Activating cross-component trace in testing

- When you configure cross-component trace with data snapshot, the generated invocation records contain the input and output data that was passed between the components during processing



© Copyright IBM Corporation 2017

Figure 3-26. Activating cross-component trace in testing

To enable cross-component trace in testing, start the server, and then:

1. Open the Server Logs view.
2. Select the “View” menu.
3. Select **Cross-Component Trace State**.

© Copyright IBM Corp. 2017

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

IBM Training

Server Logs view

- Console view returns a sizable volume of messages
- Server Logs view is the filterable view of server logs in one view
  - **`SystemOut.log`**, **`SystemErr.log`**
- Able to see SCA errors along with server errors
- Highlight events with different colors
- Display only WebSphere server logs
- Potential performance impact
  - Full logs are loaded in memory

© Copyright IBM Corporation 2017

Figure 3-27. Server Logs view

The Server Logs view is used to show server console and log file records. Although the Server Logs view automatically shows console records for each server that is started, you can also manually load and show the server console and log file records for any server. When cross-component tracing is enabled, the Server Logs view also shows invocation records that can contain the invocation data that passed between components. The Server Logs view is a tool for working with server console and log records. It provides several advantages over the traditional console view, such as the ability to filter records, display invocation records in hierarchical format, and load invocation records directly into the integrated test client.

In the Business Integration perspective, you can open the Server Logs view by clicking the Server Logs tab. (If the Server Logs tab is not visible, you can open it by selecting **Window > Show View > Server Logs**.)

The Server Logs view has three main areas:

- **Welcome tab**

  The welcome tab is the home page of the Server Logs view. It provides an overview of the Server Logs view and introduces you to some of the key tasks that you can do in the view, such as:

  - Loading the contents of server consoles or server logs onto the Server Logs view

© Copyright IBM Corp. 2017

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

  - Filtering server console or server log records in the Server Logs view
  - Enabling or disabling cross-component tracing for servers
  - Loading invocation records into the integration test client

- **Server console and log tabs**

  The server console and log tabs show the records of server consoles and server logs that you loaded into the Server Logs view. The following four columns in the server console and log tabs provide detailed information about the records:

  - **Type**: Shows the types of the records, such as log messages, FFDC records, exception records, and invocation records
  - **Time**: Shows the date and time that the records were generated
  - **Thread ID**: Shows the thread IDs of the records
  - **Contents**: Shows the first line of the contents for the records

  You can also right-click any record and select properties to open a properties dialog box that shows the time, thread ID, and contents for the record. You can choose to view the contents in the default translated format (for easier reading and assimilation) or you can view the contents in the raw format in which they were originally generated. Also, if you open the properties window for an FFDC record or an invocation record, the properties dialog box contains another data field. That data field shows the invocation data that was passed between components. Invocation records include “start,” “fail,” or “end” records.

- **Toolbar**

  The toolbar in the Server Logs view comprises several icons that are used to initiate numerous tasks.

  At the bottom of the Server Logs view, you also see a status area, which returns messages that relate to the server console and log records.

  You can click the “Select records to display” icon to open and select menu items that filter the records in the server console or log tabs. By default, any invocation records are shown in hierarchical format and have check boxes that you can use to select them for direct loading into the integrated test client. You can choose to show records in a flattened format rather than a hierarchical format. However, the invocation records do not have check boxes, and you are not able to select them for direct loading into the integrated test client.

© Copyright IBM Corp. 2017

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

IBM Training

Trace records in detail

- Cross-component trace (XCT) table displays four pieces of information about the trace element record
  - Type of trace, time stamp, thread ID, and the trace or log message
- Properties dialog box shows the trace entry in detail

Show All Record Types (Hierarchical) > with only Server State and Error Contents (Page 1 of 1)

<Type text to filter the contents>

Type	Thread ID	Contents
Invocation sequence (FleetCarRentalProcess:cancell	00000122	
End invoke (FleetCarRentalProcess:cancelReserv	00000122	End of the asynchronous invocation of operation FleetCarRentalProce...
End result retrieval (FleetCarRentalProcess:canc	000004d2	End of the result retrieval of operation FleetCarRentalProcess:cancel...
Start invoke (FleetCarRentalProcess:cancelRese	00000122	Start of the asynchronous invocation of operation FleetCarRentalProc...
End component (FleetCarRentalProcess:canc	0000002f	End of the component processing of operation FleetCarRentalProcess...
Start component (FleetCarRentalProcess:car	0000002f	Start of the component processing of operation FleetCarRentalProces...
In BPEL process	0000002f	07e8a3dd-ddab-4640-be64-54345fc78d94 STATE a3da9de9-5311-48...
End BPEL process (FleetCarRentalProces	0000012c	End of processing for BPEL process FleetCarRentalProcess:_PI:90030...
End BPEL process (FleetCarRentalProces	0000012c	End of processing for BPEL process FleetCarRentalProcess:_PI:90030...
Start BPEL process (FleetCarRentalProce	0000012c	Start of processing for BPEL process FleetCarRentalProcess:_PI:9003...
Start BPEL process (FleetCarRentalProce	0000012c	Start of processing for BPEL process FleetCarRentalProcess:_PI:9003...
Start result retrieval (FleetCarRentalProcess:car	000004d2	Start of the result retrieval of operation FleetCarRentalProcess:cance...

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-28. Trace records in detail

You can use the Server Log view in IBM Integration Designer to show invocation records that can contain the invocation data that is passed between components. Using the Server Log view, you can filter records, display invocation records in hierarchical format, and load invocation records directly into the integration test client. For more information about the Server Log view and cross-component trace from the development environment perspective, see the IBM Integration Designer IBM Knowledge Center.

© Copyright IBM Corp. 2017

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

IBM Training

Troubleshoot SCA call chains by using the Cross-Component Trace view (1 of 2)

Show All Record Types (Hierarchical) > with only Server State and Error Contents (Page 1 of 1)

<Type text to filter t

Type	Thread ID	Contents
Invocation sequence (FleetCarRentalProcess:cancell	00000122	
End invoke (FleetCarRentalProcess:cancelReserv	00000122	End of the asynchronous invocation of operation FleetCarRenta
End result retrieval (FleetCarRentalProcess:canc	000004d2	End of the result retrieval of operation FleetCarRentalProcess:c
Start invoke (FleetCarRentalProcess:cancelRese	00000122	Start of the asynchronous invocation of operation FleetCarRen
End component (FleetCarRentalProcess:canc	0000002f	End of the component processing of operation FleetCarRentalP
Start component (FleetCarRentalProcess:car	0000002f	Start of the component processing of operation FleetCarRental
In BPEL process	0000002f	07e8a3dd-ddab-4640-be64-54345fc78d94 STATE a3da9de9-53
End BPEL process (FleetCarRentalProces	0000012c	End of processing for BPEL process FleetCarRentalProcess:_PI:
End BPEL process (FleetCarRentalProces	0000012c	End of processing for BPEL process FleetCarRentalProcess:_PI:
Start BPEL process (FleetCarRentalProce	0000012c	Start of processing for BPEL process FleetCarRentalProcess:_P
Start BPEL process (FleetCarRentalProce	0000012c	Start of processing for BPEL process FleetCarRentalProcess:_P
Start result retrieval (FleetCarRentalProcess:car	000004d2	Start of the result retrieval of operation FleetCarRentalProcess

© Copyright IBM Corporation 2017

Figure 3-29. Troubleshoot SCA call chains by using the Cross-Component Trace view (1 of 2)

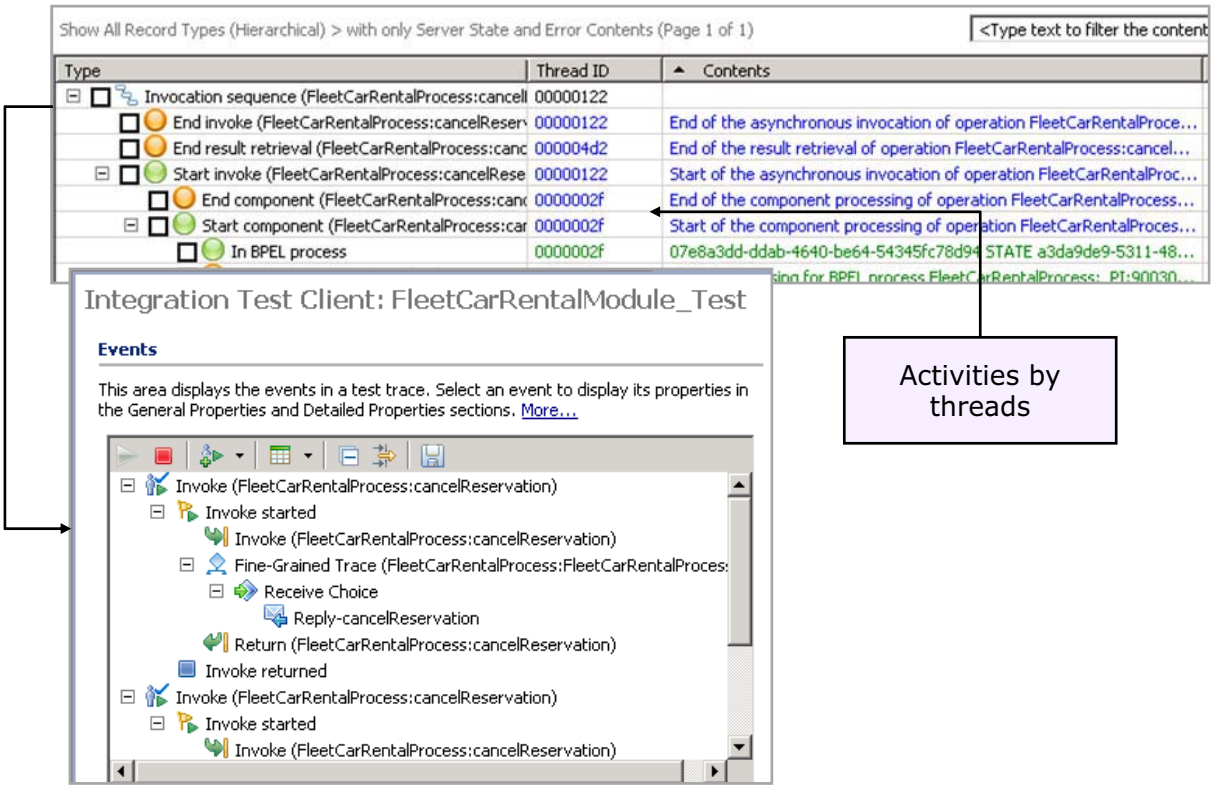
Cross-component trace identifies whether a Service Component Architecture (SCA) operation successfully completed. With it, you identify `SystemOut.log` or `trace.log` data that is associated with IBM Process Server modules and components.

© Copyright IBM Corp. 2017

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

IBM Training

Troubleshoot SCA call chains by using the Cross-Component Trace view (2 of 2)



© Copyright IBM Corporation 2017

Figure 3-30. Troubleshoot SCA call chains by using the Cross-Component Trace view (2 of 2)

Events that can be captured include:

- Errors that occur during processing because of corrupted data
- Errors when resources are not available, or are failing
- Interpretation of code paths

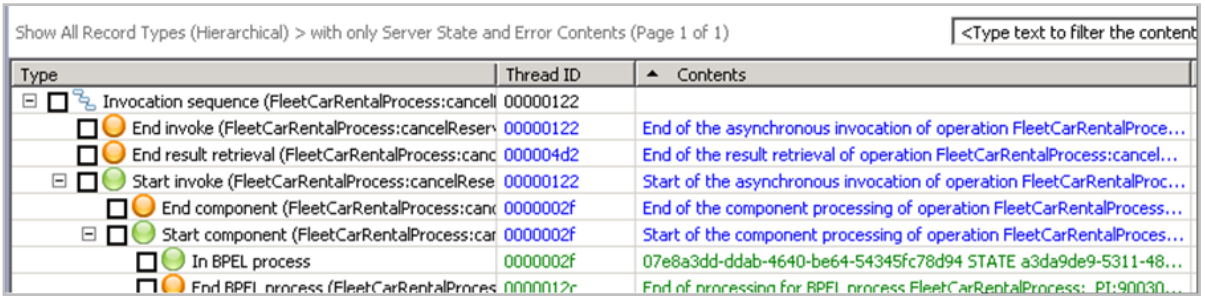
© Copyright IBM Corp. 2017

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

IBM Training

Call and log sequence

- Use the log message view to find a record of interest by time, thread, and text
- Shows execution path of all SCA components even if the source code is not in the current workspace
  - Invocation, exceptions, and input or output data
  - Can be imported into a test client for a more detailed view



© Copyright IBM Corporation 2017

Figure 3-31. Call and log sequence

You can also right-click any record and select Properties to open a Properties dialog box that shows the time, thread ID, and contents for the record. You can choose to view the contents in the default translated format (for easier reading and assimilation) or you can view the contents in the raw format in which they were originally generated. Also, if you open the Properties window for an FFDC record or an invocation record, the Properties dialog box contains another data field. That data field shows the invocation data that was passed between components. Invocation records include Start, Fail, or End invocation records.

© Copyright IBM Corp. 2017

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

3.4. Testing and logging with the Component Test Explorer

© Copyright IBM Corp. 2017

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

IBM Training

Testing and logging with the Component Test Explorer

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-32. Testing and logging with the Component Test Explorer

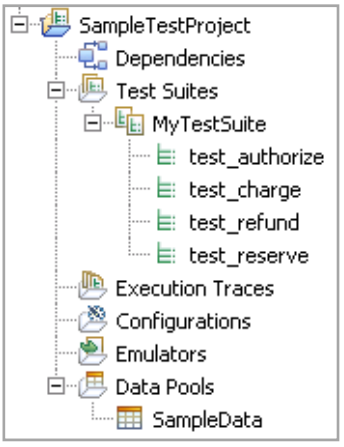
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Component test projects (1 of 2)

Component test projects provide a way to automate running test cases

- Component tests automate and test operations in the integrated test client
- Test projects provide testing for components in integration (SCA) modules
- Test projects are packaged as SCA modules (can add Java libraries by using dependencies)
- Component test projects can be deployed and run on the server
- Results are displayed in the integration test client
- Component test projects include:
  - **Test suites**: Containers for test cases
  - **Test cases**: Containers for operations
  - **Test configurations**: Used to control tests



Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-33. Component test projects (1 of 2)

In component testing, you use the test suite editor and associated wizards to create and define test cases. Those cases comprise multiple operations, which you can use to automate and simultaneously test the operations in the integrated test client. The component test project that you create is a module that is deployed to the IBM Process Server test environment along with the module that is being tested. Component test projects can be started from an Ant script.

Component test projects are containers for test suites. Test suites are containers for test cases. Test cases are containers for operations, and test configurations are used to control your tests.

Component test projects

Generally, creating a component test project to contain your test suites is the first step in preparing to do component testing.

Test suites

After you create a component test project, you generally create a test suite to contain your test cases. When you create a test suite, you can choose from one of the following test patterns:

- Operation-level testing
- Scenario-based testing

© Copyright IBM Corp. 2017

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

In operation-level testing, a separate test case is created for each operation that you select for component testing. In scenario-based testing, a single test case is created for all of the operations that you select for component testing.

Test cases

You can think of a test case as a container for multiple operations that you select for testing. You can use test cases to automate and simultaneously test the operations in the integrated test client. Test data for test cases is defined in the test data table, which contains a set of named variables that can be used for either input or output in the test case invocations.

Test configurations

A test configuration specifies one or more modules to test. Each configuration might include zero or more emulators for components or references in the module. Each configuration might also include zero or more monitors for the wires in the module. When you open the test suite editor, a default test configuration is automatically created that you can use for testing. The default test configuration is often all that you need for testing your test cases. However, you can choose to add modules to your test configuration, or add emulators and monitors to your test configuration modules.

Test suite editor

The test suite editor is the designated tool for editing test suites. It features a rich user interface that you can use to easily manage your test suites and test cases. The test suite editor was designed to closely resemble the test client, which helps you transition from using one tool to the other.

Top-down testing of test cases

In component testing, you use the test suite editor and associated wizards to create and define test cases. Those test cases comprise multiple operations, which you can use to automate and simultaneously test the operations in the integration test client. In top-down testing, you define test cases by selecting components and operations for testing. You then run the test cases in the integration test client (in bottom-up testing, you use integration test client invocations to define test cases).

Component test projects are deployable SCA modules. This architecture simplifies build automation where a component test project can be built like any other SCA module. In addition, you can add dependent Java libraries easily since component test projects have a dependency editor. By adding Java libraries, you can add your own custom utility functions.

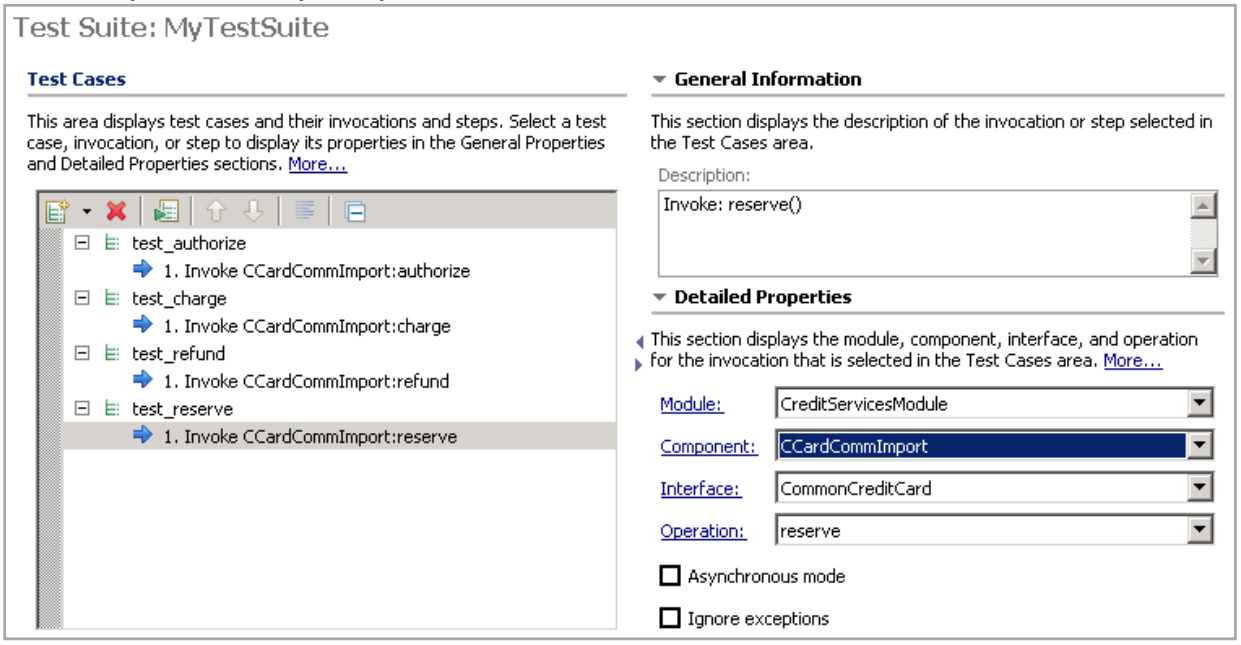
© Copyright IBM Corp. 2017

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

IBM Training

Component test projects (2 of 2)

- Component test projects are created in the test suite editor
- Associated wizards are used to create and define test cases that comprise multiple operations



Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-34. Component test projects (2 of 2)

The test suite editor interface has three main pages and a table view:

- **Overview page**: In the overview page, you can quickly view the essential information for the test suite that is open in the test suite editor.
- **Test cases page**: In the test cases page, you can do numerous activities to prepare your test suite and test cases for testing. These activities include selecting an operation invocation for which you want to specify values in the test data table.
- **Configurations page**: In the configurations page, you can edit the default test configuration. You can use it to add modules to your test configuration, or add emulators and monitors to your test configuration modules, to more precisely control your tests.
- **Test data table view**: At the bottom of the test suite editor, a test data table view is provided which you can use to import Java extension classes or packages and specify, view, and edit values for operations.

© Copyright IBM Corp. 2017

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

IBM Training

Working with test cases (1 of 2)

- All component test projects that are deployed to the server can be viewed
- All test cases can be run or scheduled for execution
- Run history is saved and stored in a database on the server
- Component test cases
  - View and execute
  - Examine results
  - Scheduler

© Copyright IBM Corporation 2017

Figure 3-35. Working with test cases (1 of 2)

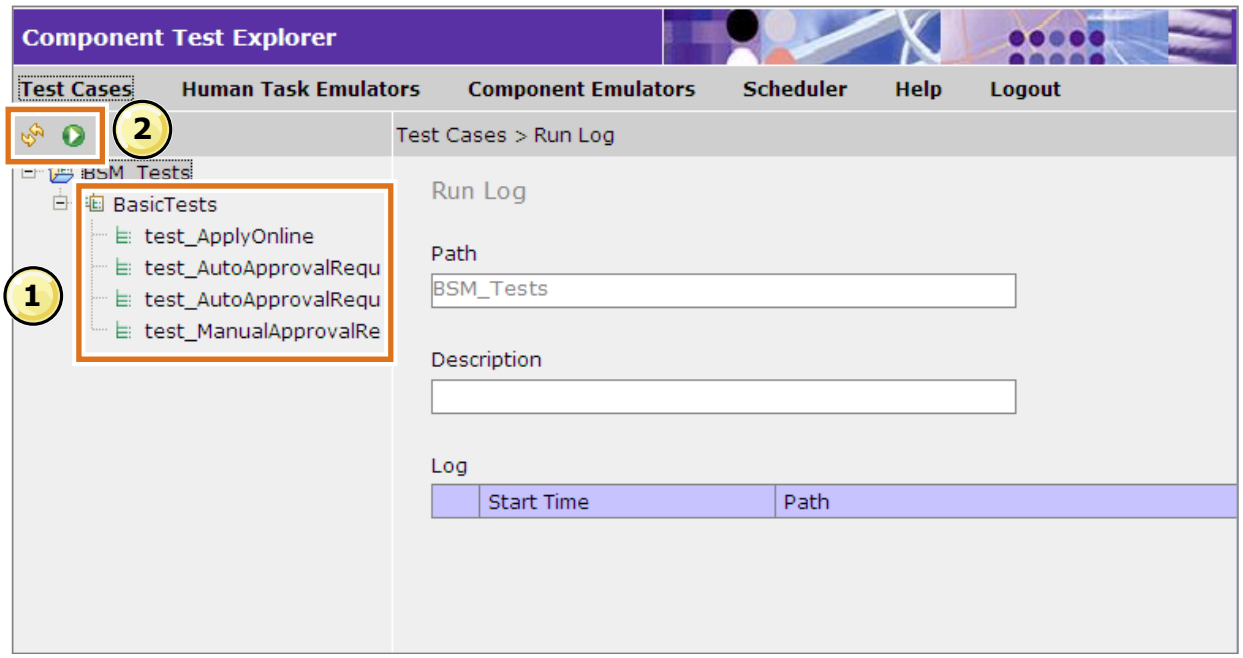
To work with the Component Test Explorer, you must first create a component test project in IBM Integration Designer. This test project might be composed of test suites or cases, execution traces, configurations, data pools, and emulators. After the component test project is constructed, it must be deployed to the server in the same way that an SCA application is deployed (for example, **Add** and Remove Projects).

© Copyright IBM Corp. 2017

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

IBM Training

Working with test cases (2 of 2)



© Copyright IBM Corporation 2017

Figure 3-36. Working with test cases (2 of 2)

When working with test cases in the Component Test Explorer:

1. Select the test that you want to run from the set of loaded tests.
2. Click **Execute** to begin the test.
3. When the test is complete, its results are logged in the explorer. Select a test to view its results or to remove it from the log.

© Copyright IBM Corp. 2017

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

IBM Training

Using emulation in test cases (1 of 2)

- Why use an emulator?
  - The component might not be available at test time
  - Execution of the component might produce errors
  - The component might rely upon other live or production systems that are not conducive to testing
- Global emulators can be defined for any component or human task that is defined on the server

© Copyright IBM Corporation 2017

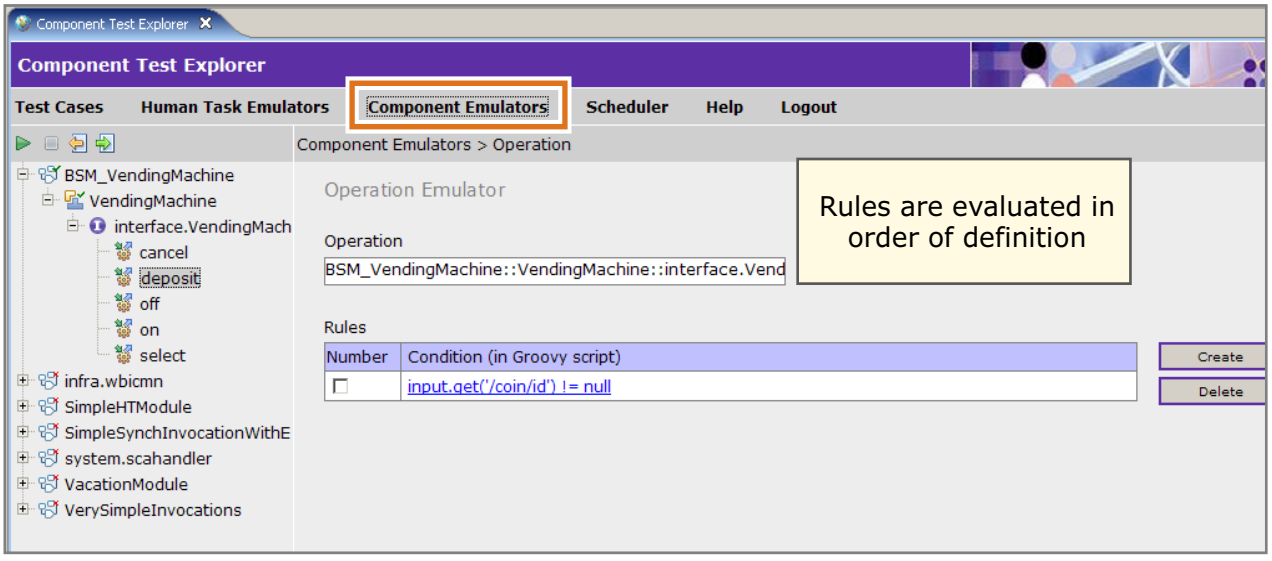
Figure 3-37. Using emulation in test cases (1 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Using emulation in test cases (2 of 2)



© Copyright IBM Corporation 2017

Figure 3-38. Using emulation in test cases (2 of 2)

An emulator supplies a start button and a stop button. The start button activates all enabled emulators. If the user selected “yes” to emulate a component, an emulator is enabled. A green check mark next to the component in the left pane navigator determines the components that are enabled. The stop button deactivates the emulators.

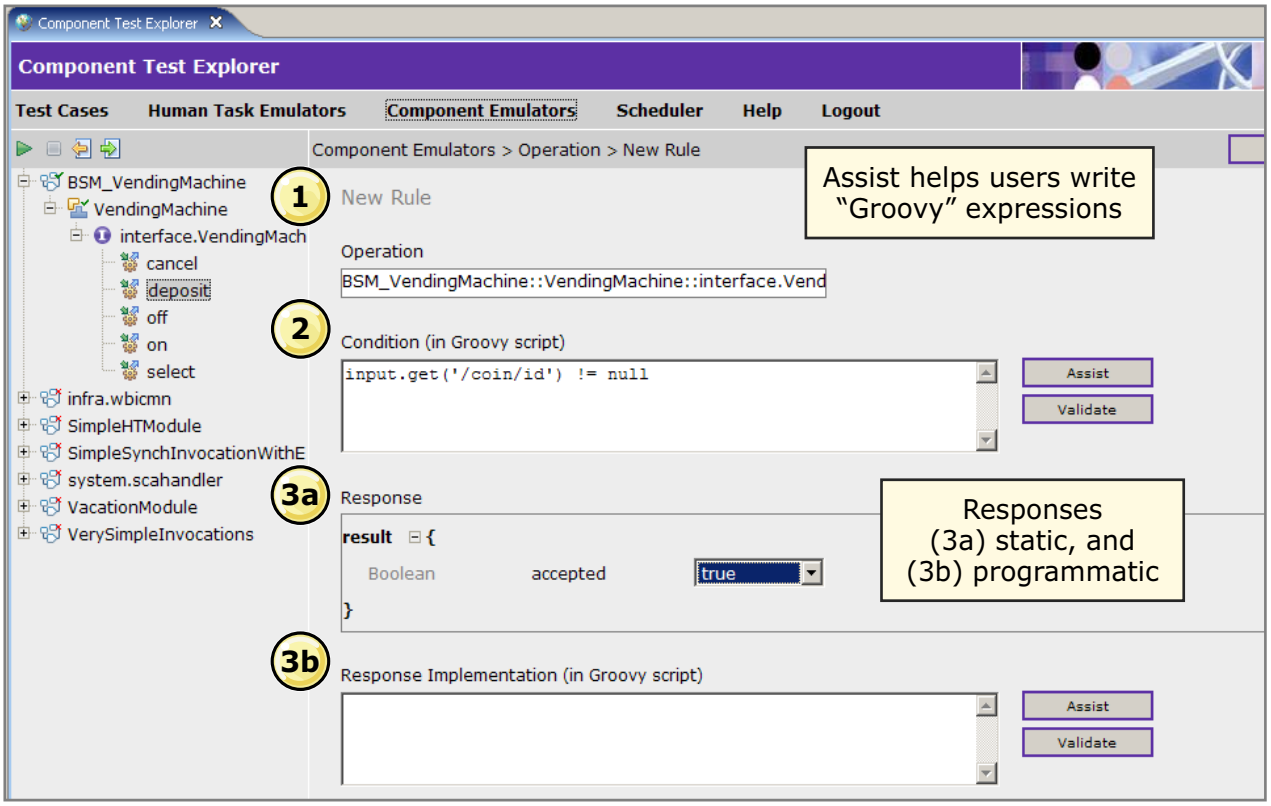
Import and export buttons are also provided to import or export XML to or from the emulator.

© Copyright IBM Corp. 2017

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

IBM Training

Building a new emulation rule



 © Copyright IBM Corporation 2017

Figure 3-39. Building a new emulation rule

Choose the component that you want to emulate from the set of applications in the left menu. A menu asks whether you want to emulate the component. If you answer yes, emulation is turned on for the component in the left menu (a green check mark is the indicator), and the list of emulation rules are shown. To create an emulation rule, click **Create** and then:

1. Give the Operation rule a name.
2. Write the conditions for the rule.
3. Set the response to either (3a) statically, or (3b) programmatically. Use the Groovy assist editor to help write the response implementation.

   Use Groovy assistance to emulate interface information for either request or response parameters.

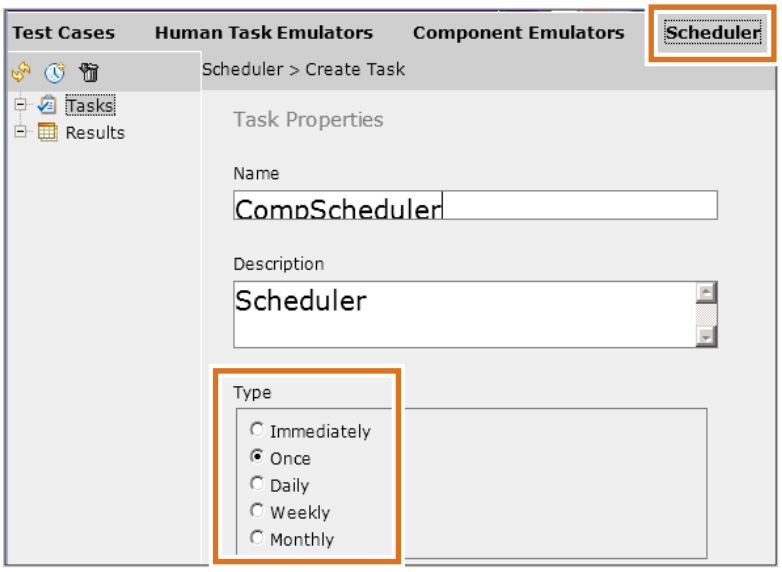
© Copyright IBM Corp. 2017

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

IBM Training

Using the scheduler

- Scheduler is divided into tasks and results
- **Tasks** are scheduled jobs
  - Tasks have a unique name
  - Assign a frequency to each task (run one time or on an interval)
  - Intervals are daily, weekly, or monthly
  - Choose the tests to start
  - Tests must be loaded on the server to schedule



- **Results** record the output from scheduled jobs
  - Results are logged to a database
  - Click the results to show the start and finish times of each job

Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-40. Using the scheduler

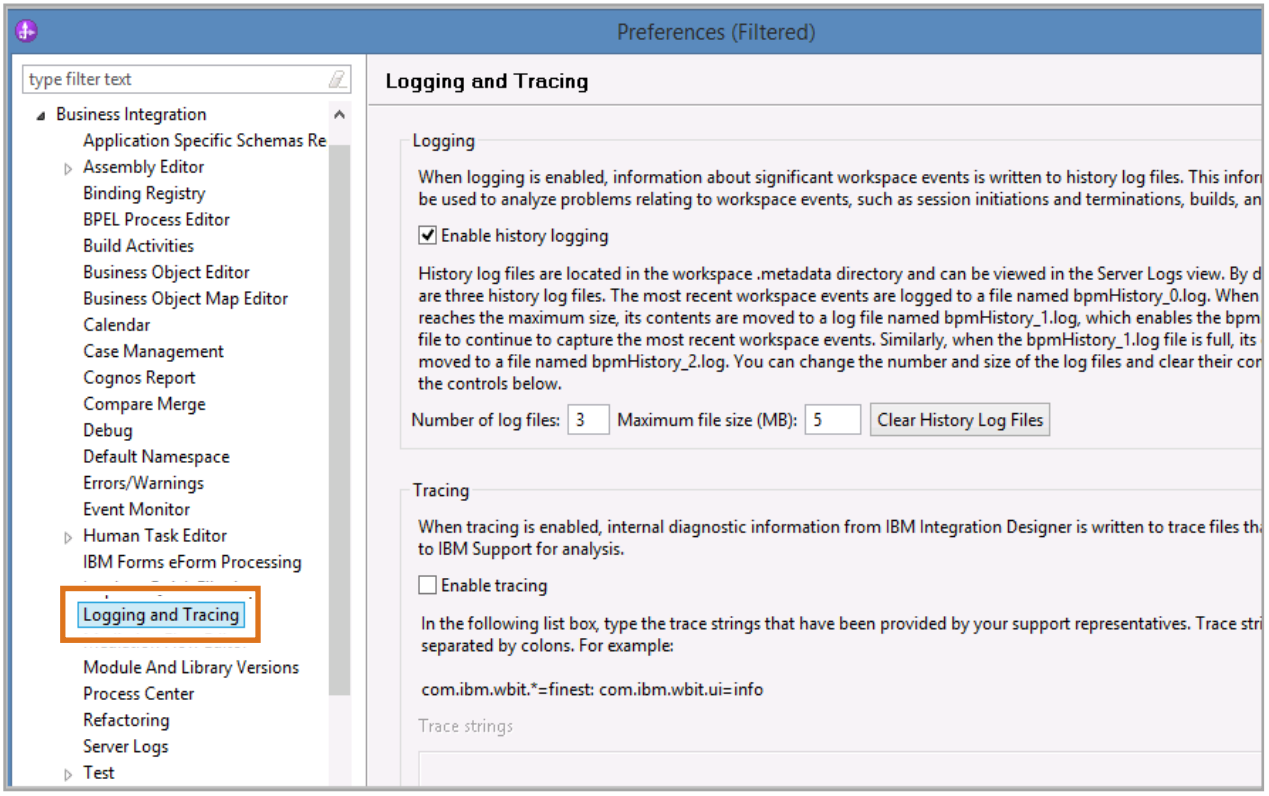
You can also right-click any record and select Properties to open a Properties dialog box that shows the time, thread ID, and contents for the record. You can choose to view the contents in the default translated format (for easier reading and assimilation) or you can view the contents in the raw format in which they were originally generated. Also, if you open the Properties window for an FFDC record or an invocation record, the Properties dialog box contains another data field. That field shows the invocation data that was passed between components.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Integration Designer problem determination



Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-41. IBM Integration Designer problem determination

The following characteristics of logging and tracing are important:

- Logging:
  - On by default
  - To see workspace event history during IBM Integration Designer error
  - In the `.metadata` folder of the workspace `widHistoryxx.log` file
  - Can be viewed in the Server Logs view
  - Load from file
- Tracing:
  - Internal diagnostic information
  - Logging levels can be changed in real time without needing to restart

Logging and tracing in IBM Integration Designer help with problem determination of the tool environment. Logging is turned on by default and keeps three log files at the size of 5 megabytes. You can use the log to see workspace event history during IBM Integration Designer errors. For example, if you receive a null pointer error when opening an editor, that is captured in the log.

© Copyright IBM Corp. 2017

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

The `widHistoryxx.log` is in the `.metadata` folder of the workspace that had the error. The "`xx`" in the log file name represents the number of log files. The easiest way to view these files is in the Server Logs view where you choose to "load from file". You can select all of the `widHistoryxx.log` files. Tracing is not on by default and is used for internal diagnostic information. You use tracing to help report issues to IBM Support. Saving any change for logging or tracing in the preferences has an immediate effect. You do not need to restart the software or workspace.

© Copyright IBM Corp. 2017

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

IBM Training

The IBM Support Assistant

- Can assist you with every step of problem determination
- For new users: IBM Support Assistant explains in detail what steps to take
  - Provides supporting information to help you understand the problem and the possible solutions
- For experienced users: IBM Support Assistant provides quick reminders of what steps to take
  - Shows only the information that is necessary for you to quickly move through the steps
- Additional information is only a click away if you want to brush up on a specific section
- For more information, see: http://www.ibm.com/software/support/isa/

© Copyright IBM Corporation 2017

Figure 3-42. The IBM Support Assistant

The IBM Support Assistant assists you with every aspect of debugging, from basic problem-solving to gathering the diagnostic information that is needed to open a problem management report with IBM support.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe version control schemes in IBM Process Server
- Identify the IBM Process Server components that provide version control
- Compare version control in IBM Process Server to snapshots in IBM Process Center
- Describe some of the version control considerations for SCA applications
- Describe the IBM Integration Designer tools for debugging applications
- Describe the advanced testing facilities that are available in IBM Integration Designer
- Explain the purpose of the IBM Integration Designer Server Logs view
- Explain the business value of the IBM Support Assistant

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-43. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: The SCA specification does not implement version control, but the WS-BPEL specification does.
2. True or False: When multiple running process templates have the same name, the template with the most recent valid-from date is always used to start the instance.
3. True or False: The only tool that you can use to create a version-aware EAR file is the serviceDeploy tool.

© Copyright IBM Corporation 2017

Figure 3-44. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: The SCA specification does not implement version control, but the WS-BPEL specification does.
   The answer is True.
2. True or False: When multiple running process templates have the same name, the template with the most recent valid-from date is always used to start the instance.
   The answer is True.
3. True or False: The only tool that you can use to create a version-aware EAR file is the serviceDeploy tool.
   The answer is True.

© Copyright IBM Corporation 2017

Figure 3-45. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Version control for SCA applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 3-46. Exercise 2

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

IBM Training

Exercise objectives

- Create multiple versions of modules and libraries
- Deploy modules by using the serviceDeploy tool
- Install and start the generated EAR file
- Test the versioned application with cross-component tracing
- Troubleshoot applications in the administration console and Failed Event Manager
- Test the versioned application

Version control and testing of SCA applications

© Copyright IBM Corporation 2016

Figure 3-47. Exercise objectives

© Copyright IBM Corp. 2017

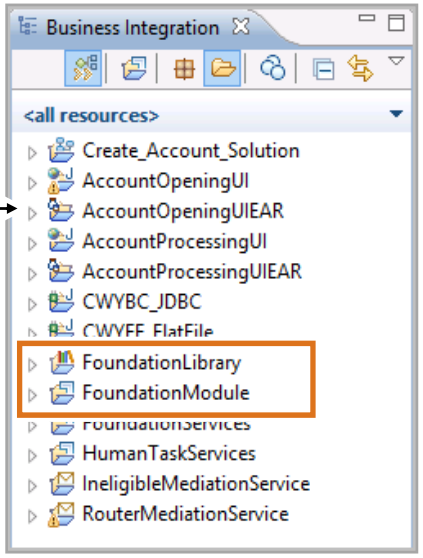
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **FoundationLibrary**



New components that you create in the lab:

1. FoundationModule_v1.0 ear
2. FoundationModule_v2.0 ear

Version control and testing of SCA applications

© Copyright IBM Corporation 2017

Figure 3-48. Components that are required for Exercise 2

Version control is the ability for the runtime environment to identify snapshots in the lifecycle of a solution or service and to be able to concurrently run multiple snapshots at the same time. Version control is beneficial in a production environment for a number of reasons. Version control allows the simultaneous deployment of multiple versions of an SCA module to the same deployment environment.

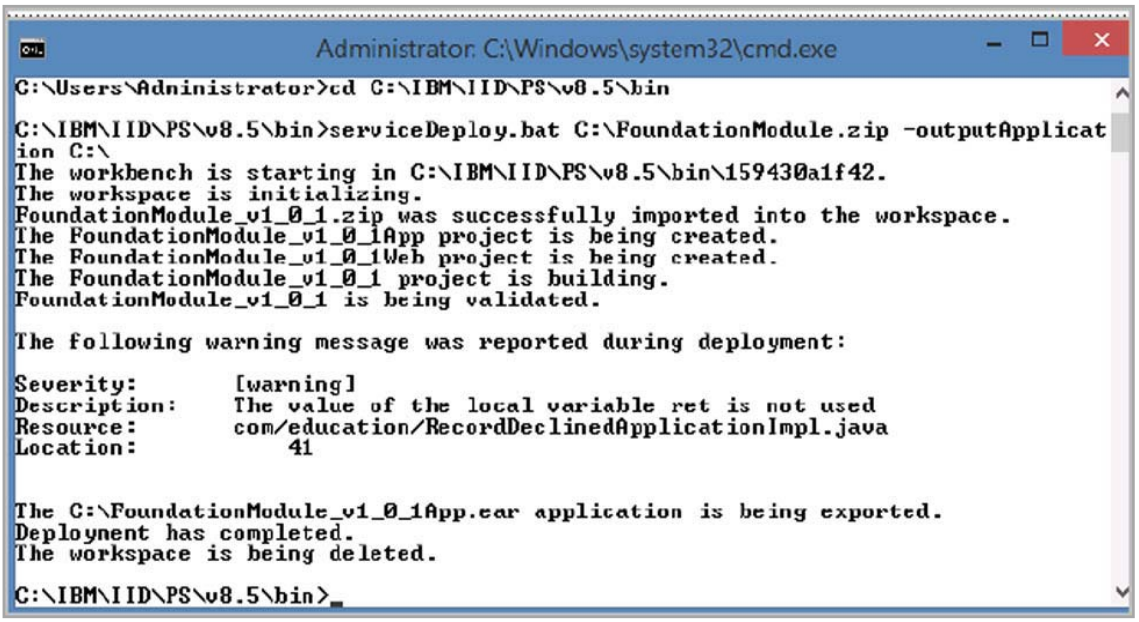
In this exercise, you create versioned modules and libraries in IBM Integration Designer for future deployment to a production environment. These versions are for use on the server at run time, not for distinguishing levels of completion in a development environment. The stored data consists of the version number (value), the version provider, and the library dependency.

© Copyright IBM Corp. 2017

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

IBM Training

Run the serviceDeploy command to create a version-aware EAR file in Exercise 2



Version control and testing of SCA applications © Copyright IBM Corporation 2017

Figure 3-49. Run the serviceDeploy command to create a version-aware EAR file in Exercise 2

In this section, you deploy the module to the server and verify that the version information is included. Versioned modules can be exported for command-line service deployment and deployed with the serviceDeploy command to create a version-aware EAR file. Versioned modules that are exported as EAR files are not version-aware.

© Copyright IBM Corp. 2017

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

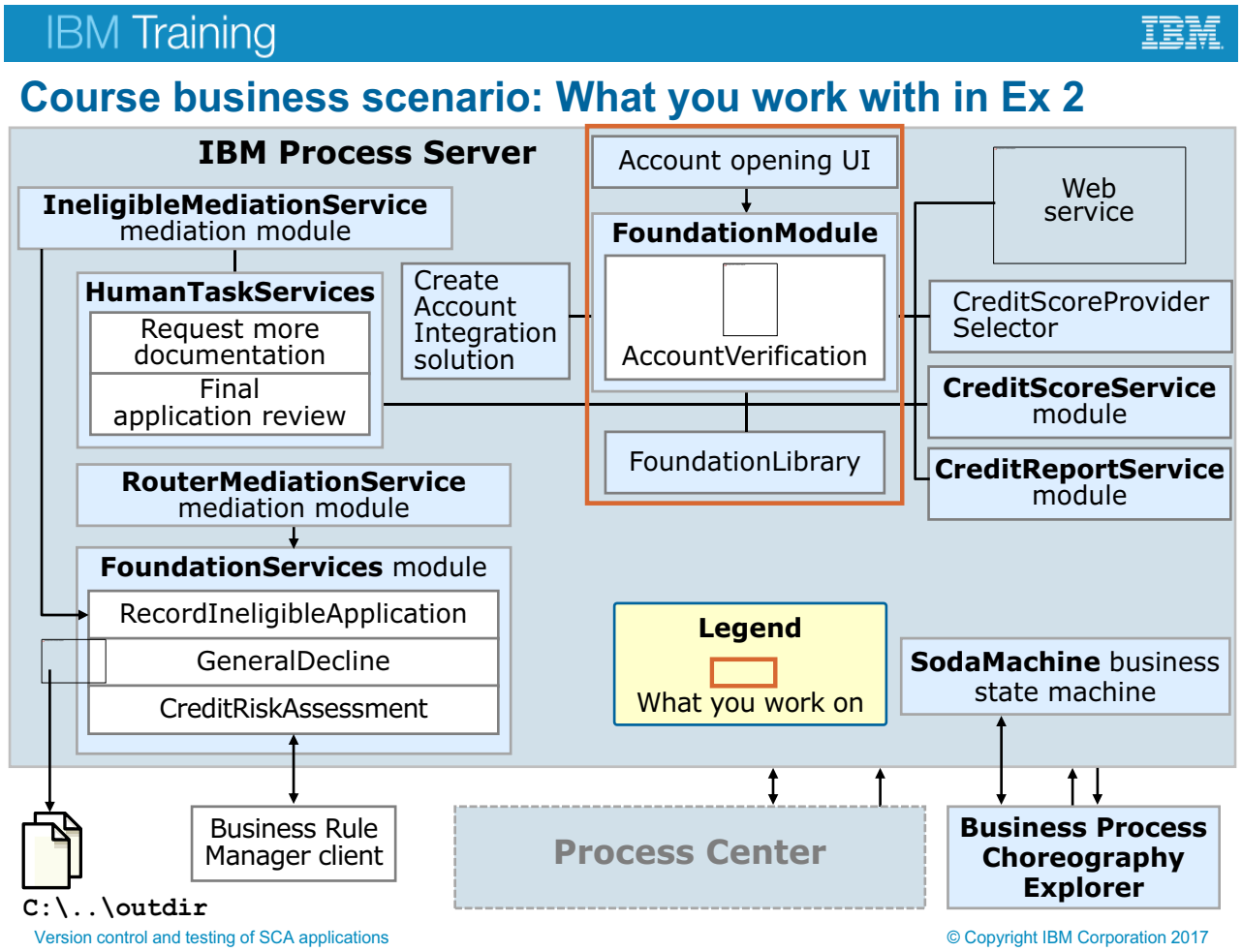


Figure 3-50. Course business scenario: What you work with in Ex 2

In this lab, you work with the IBM Process Server test environment.

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

Unit 4. SCA bindings, the SCA runtime, and SCA quality of service qualifiers

Estimated time

01:30

Overview

This unit describes how the SCA runtime environment handles synchronous and asynchronous communication, SCA bindings, and SCA quality of service qualifiers.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain how the SCA runtime handles synchronous and asynchronous communication
- Describe the function of the Failed Event Manager
- Describe the SCA import and export bindings
- Identify the components of an SCA import or export
- Describe the role of SCA quality of service qualifiers in transmission priority and level of route reliability
- Explain the purpose and value of interface qualifiers, implementation qualifiers, and reference qualifiers

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- SCA runtime
- Failed Event Manager
- SCA bindings
- SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-2. Topics

© Copyright IBM Corp. 2017

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

4.1.SCA runtime

© Copyright IBM Corp. 2017

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

IBM Training

SCA runtime

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 4-3. SCA runtime

© Copyright IBM Corp. 2017

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

IBM Training

SCA concepts: Interactions

- Synchronous calls
  - Exchanging data by reference within a service module
  - Allows implementation of services that use a shared memory form of interaction with their clients
- SCA provides a simple model for asynchronous invocations
- Several types of invocations are by value
  - Asynchronous invocation
  - Exchanging data with an exported service
  - Exchanging data through imported external service
- Shared memory optimization in same JVM
- Shared memory interaction cannot be exported from a service module or used asynchronously

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-4. SCA concepts: Interactions

In synchronous calls, data gets passed by reference in memory within a module. The reason is because a module essentially deploys as an EAR file and runs in its own JVM. For asynchronous invocations, the data gets passed by value.

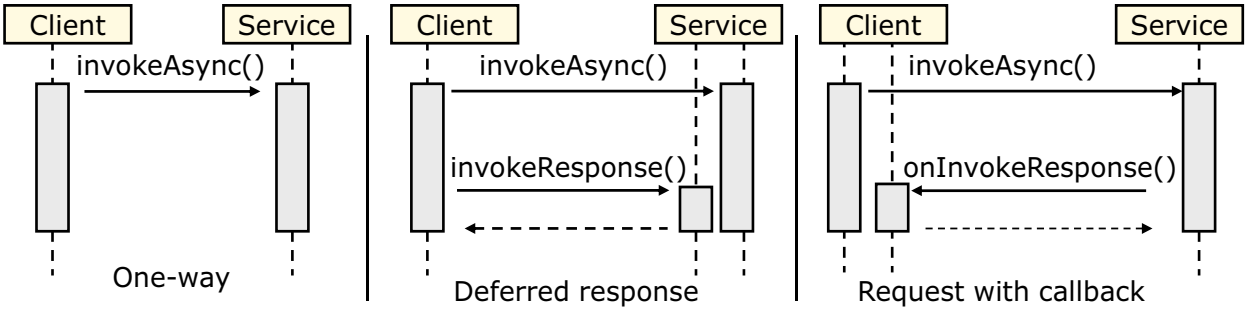
© Copyright IBM Corp. 2017

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

IBM Training

SCA invocation styles

- SCA components can call services synchronously or asynchronously, depending upon the preferred interaction style that is defined in the interface
  - The **synchronous** method is called `as invoke`
  - The **asynchronous** method is called `as invokeAsynch`
- A synchronous call waits for the return value before proceeding
- Three kinds of asynchronous invocations exist:
  - **One-way**: No response is expected ("fire and forget")
  - **Deferred response**: Caller fetches the response later by using a ticket
  - **Request with callback**: Callee sends the response back to the caller when the result becomes available



SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-5. SCA invocation styles

The IBM Process Server runtime engine uses the service integration bus for asynchronous invocations or RMI/IIOP (Remote Method Invocation over Internet InterORB Protocol) for synchronous invocations. In asynchronous invocations, message consumers and producers do not communicate directly; they use a bus to send and receive messages. Three types of asynchronous interaction styles are available. With all three types, the client receives control back immediately from the SCA runtime upon an `invokeAsync()` call; however, the client can capture the response later in three different ways:

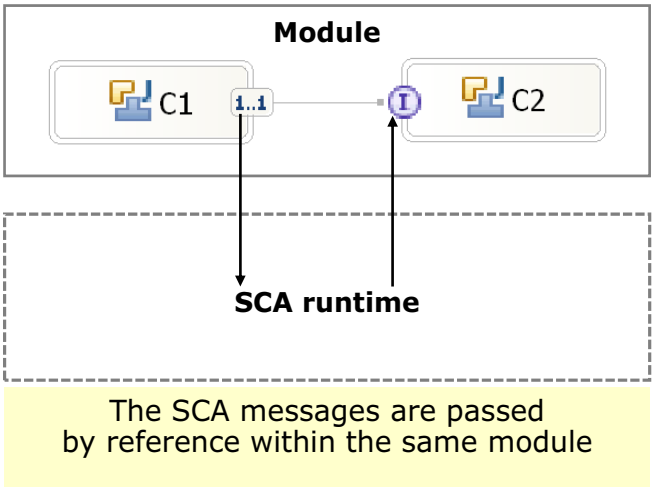
- First, the client might choose to discard the response entirely (or if it is a call to a void method). In this case, the asynchronous invocation is known as "one-way".
- Another option is for the client to call `invokeAsync()` and then continue processing until some later time when the client makes a request to capture the response. This scenario is termed "deferred response."
- Finally, the client also has the option of doing an asynchronous "request with callback." To implement this option, the client must first implement the `ServiceCallback` interface. Then, after calling `invokeAsync()`, the SCA runtime provides a callback to the `ServiceCallback` handler to provide the response to the client.

© Copyright IBM Corp. 2017

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

IBM Training

Reference: SCA synchronous intramodule



© Copyright IBM Corporation 2017

Figure 4-6. Reference: SCA synchronous intramodule

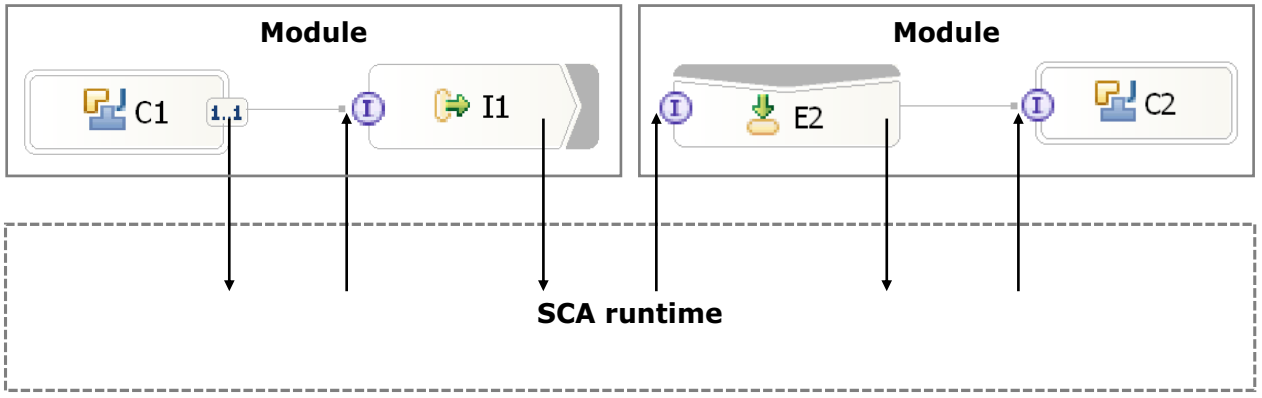
The first communication pattern occurs between SCA components within the same module interacting synchronously. The SCA messages between the reference of component C1 and the interface of component C2 are passed by reference in the SCA runtime, which means that it happens in memory. This communication pattern is fast.

© Copyright IBM Corp. 2017

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

IBM Training

Reference: SCA synchronous intermodule (one server)



The SCA messages are passed by value across modules

© Copyright IBM Corporation 2017

Figure 4-7. Reference: SCA synchronous intermodule (one server)

The second communication pattern is also of the type synchronous. When spanning the SCA communication over module boundaries on the same server, the SCA messages are passed in the SCA runtime environment as well (no queues are used).

The difference here is that the messages are passed by value. The speed is up to 10 times slower than the speed of SCA synchronous intramodule communication. Copying parameters into the scope of module M2 results in speed slowdown.

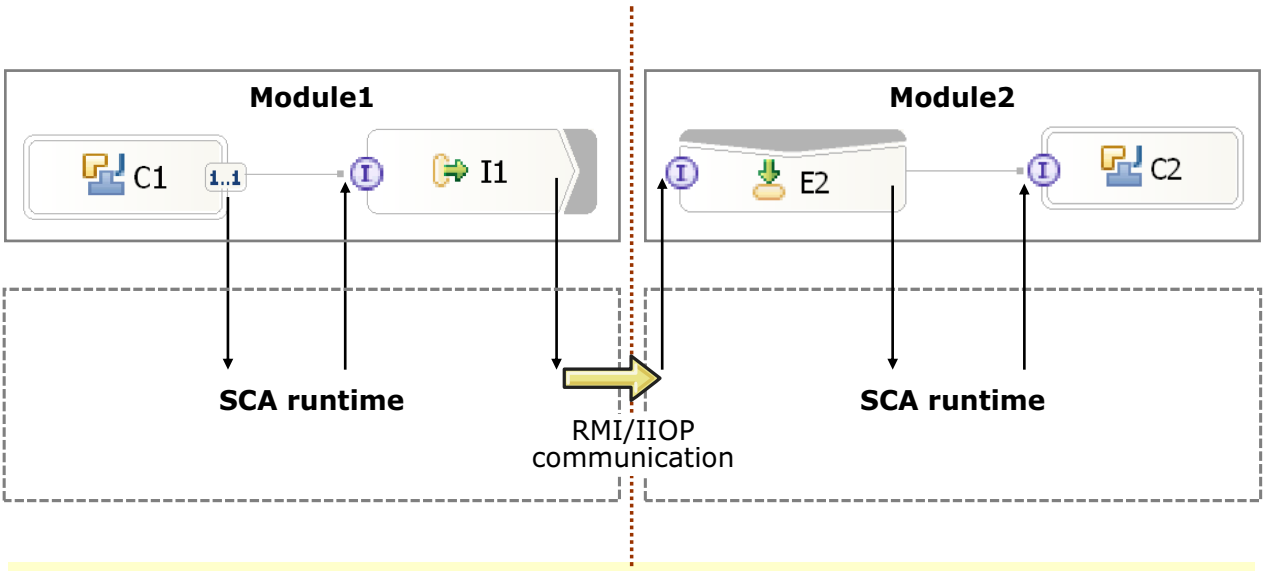
Intermodule SCA parameters, whether synchronous or asynchronous, are passed by value (that is, the input is copied from M1 to M2, and the results are copied backwards from M2 to M1). Optimization of the runtime environment allows messages to be passed by value in this case without serialization.

© Copyright IBM Corp. 2017

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

IBM Training

Reference: SCA synchronous intermodule (two servers)



The SCA messages are passed by value across different servers by using RMI/IIOP

© Copyright IBM Corporation 2017

Figure 4-8. Reference: SCA synchronous intermodule (two servers)

The third communication pattern is synchronous, spanning SCA modules that exist on different servers. The SCA messages are transferred from one SCA at run time to the SCA runtime on the second server by using the RMI/IIOP protocol.

The speed here is up to 100 times slower than SCA synchronous intramodule communication and up to 10 times slower than SCA synchronous intermodule communication. The slowdown is primarily because of the serialization and deserialization of data for RMI/IIOP communication between servers.

© Copyright IBM Corp. 2017

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

IBM Training

SCA-to-runtime artifact mapping: Asynchronous

	Module	Export	Component	Import
Component	M			
Artifact	sca.module	E.export	C.component	I.import
System bus message resources [SCA. SYSTEM. <*cellname*>. Bus]	• sca/M	• sca/M/export/E • sca/M/ exportlink/E	• sca/M/ component/C	• sca/M/import/I • sca/M/ importlink/I

© Copyright IBM Corporation 2017

Figure 4-9. SCA-to-runtime artifact mapping: Asynchronous

In the development environment (IBM Integration Designer), the following artifacts (Java class files) are created for you:

- For the SCA export named "`E`": The artifact file `E.export` is generated.
- For the SCA component named "`C`": The artifact file `C.component` is generated.
- For the SCA import named "`I`": The artifact file `I.import` is generated.

When deploying the module into the runtime environment, several JMS destinations are created on the system bus for the components that are deployed in the EAR file. The bottom row of the slide shows their IBM Process Server identifiers. Later in this unit, you see how these destinations are used.

The `sca/M/exportlink/E` destination is created only for exports with SCA binding; other binding types do not result in the creation of this destination. The same is true for the `sca/M/importlink/I` destination. It is created only for imports with SCA binding. Other bindings do not result in the creation of this destination.

© Copyright IBM Corp. 2017

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

IBM Training

Asynchronous messaging resources at deployment

- SCA modules use JMS queues on the SIBus primarily for asynchronous communication
- At run time, the SCA components read from or write to these queues
- At deployment time, different resources (related to JMS) need to be created
- Tasks that are performed include:
  - Creating SI bus destinations
  - Setting the forward routing path
  - Creating activation specifications
  - Creating connection factories
  - Instantiating message-driven beans
- These resources are created automatically for you when you deploy a new module on IBM Process Server

© Copyright IBM Corporation 2017

Figure 4-10. Asynchronous messaging resources at deployment

At deployment time JMS resources are configured based on the content of the EAR file (enhanced EAR). These JMS resources include: queues, activation specs, MDBs, and routing paths.

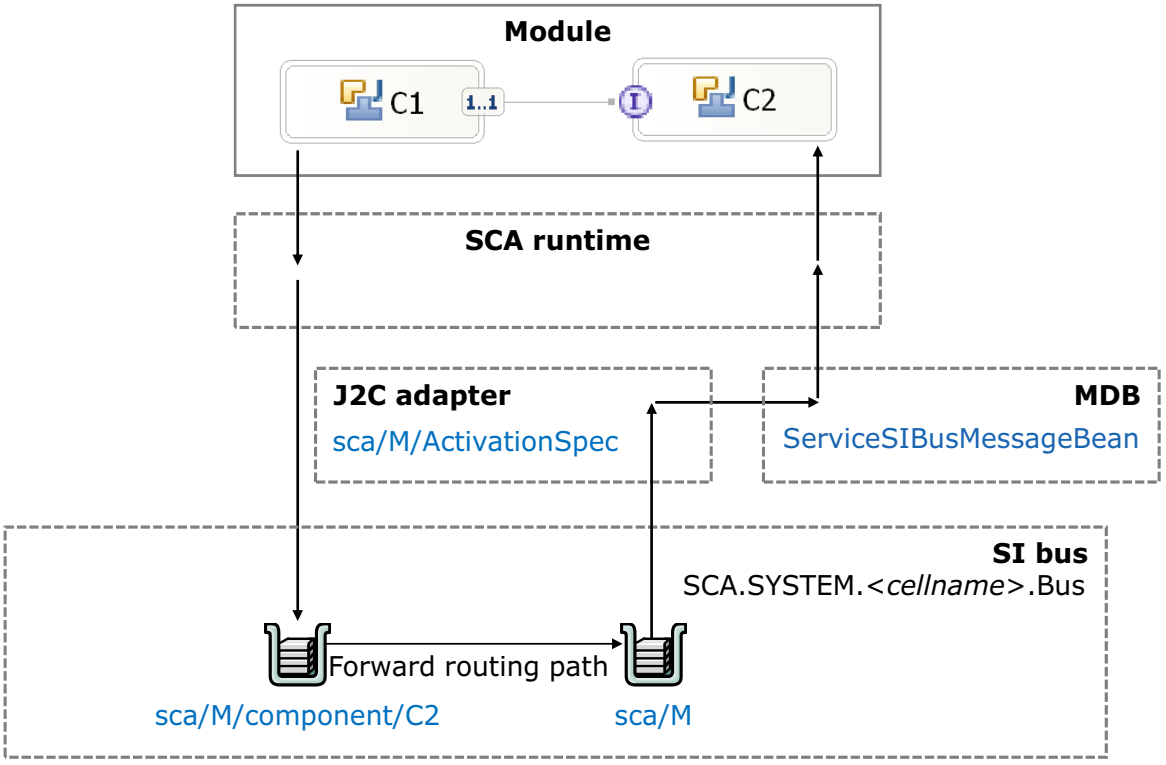
Asynchronous SCA communication uses these automatically created resources for message exchange.

© Copyright IBM Corp. 2017

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

IBM Training

SCA asynchronous intramodule



SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-11. SCA asynchronous intramodule

The first asynchronous pattern address the communication between the SCA components within the same module. The main differences between this and synchronous invocation styles are the additional SIBus components that are used in message transport.

As JMS communication is used here, standard JMS mechanisms like activation specifications and message-driven beans are used to process messages along with queue destinations. The J2C adapter in this diagram is the SIBus resource adapter. The message-driven bean in the diagram is an SIBus MDB, not a JMS MDB. The forward routing path identifies a sequence of bus destinations that a message must pass through after it was delivered to the bus destination to which the producer is attached. This routing path is created on the destination automatically upon module deployment.

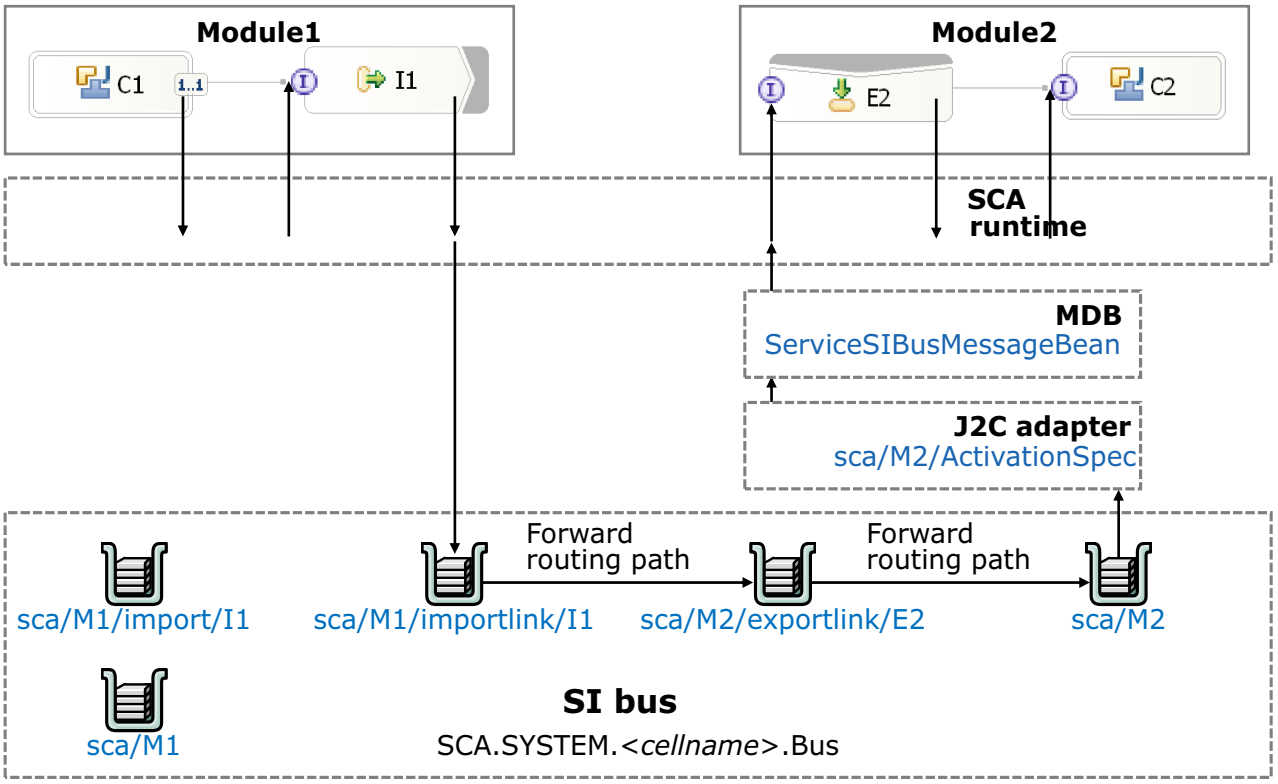
Speed is at least 100 times slower than SCA synchronous intramodule communication. It is also slower than SCA synchronous intermodule communication, but the SCA runtime is optimized so that no serialization or deserialization occurs within the same module.

© Copyright IBM Corp. 2017

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

IBM Training

SCA asynchronous intermodule (one server)



© Copyright IBM Corporation 2017

Figure 4-12. SCA asynchronous intermodule (one server)

When SCA asynchronous communication is used between modules on the same server, the messages are transferred by using importlink and exportlink queues (`sca/M1/importlink/I1`, sca/exportlink/E2) to the `sca/M2` queue. The activation specification and the message-driven bean process the message from the “central” on queue `sca/M2` into the module scope for Module2.

The J2C adapter in this diagram is the SIBus platform resource adapter. The message-driven bean in the diagram is an SIBus MDB.

Asynchronous communication between modules in this scenario involves passing parameters by value. However, SCA is optimized so that if the modules are in the same JVM, no serialization, or deserialization occurs.

At run time, the code checks whether the I1 import is asynchronous for either SCA or JMS. If the communication is asynchronous, then communication between C1 and I1 is converted to synchronous communication.

© Copyright IBM Corp. 2017

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

4.2.Failed Event Manager

© Copyright IBM Corp. 2017

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

IBM Training

Failed Event Manager

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 4-13. Failed Event Manager

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

IBM Training

Failed Event Manager

- Use the failed event manager to find and manage failed events on all servers in a deployment environment
- By using the interface, you can:
  - View and edit the data for a failed event
  - Resubmit a failed event
  - Delete a failed event

© Copyright IBM Corporation 2017

Figure 4-14. Failed Event Manager

Use the failed event manager to find and manage failed events on all servers in a deployment environment. With this interface, you can view (and in some cases, edit) the data for a failed event, resubmit a failed event, or delete a failed event.

© Copyright IBM Corp. 2017

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

IBM Training

Failed Event Manager event types

- Service Component Architecture (SCA) failed events
  - Failed asynchronous request or response
- Java Message Service (JMS) failed events
  - Failed data binding or function selector operation
- Business Process Choreographer stopped, terminated, and failed events
- WebSphere MQ failed events
- Business Flow Manager hold queue events
  - A navigation message might be stored in the hold queue if an infrastructure, such as a database, is unavailable or if the message is damaged

SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-15. Failed Event Manager event types

In the context of SCA, an event is a request or response that a service application receives. It can come from an external source (such as an inbound application adapter) or an external invocation to a web service. The event comprises a reference to the business logic it wants to operate and its data, which is stored in a Service Data Object (a business object). When an event is received, the appropriate application business logic processes it.

In the context of JMS, a failed event is generated when:

- A JMS data binding or function selector operation fails
- An exception occurs in a downstream component and is thrown back to the JMS export or import

The recovery service handles failed events from JMS data bindings and IBM MQ JMS data bindings.

In the context of Business Process Choreographer, exceptions can occur that, if the process logic does not handle it, cause an activity to stop or the process instance to fail. A failed event is generated when a long-running Business Process Execution Language (BPEL) process fails and one of the following happens:

- The process instance enters the Failed or Terminated state.

© Copyright IBM Corp. 2017

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

- An activity enters the Stopped state.

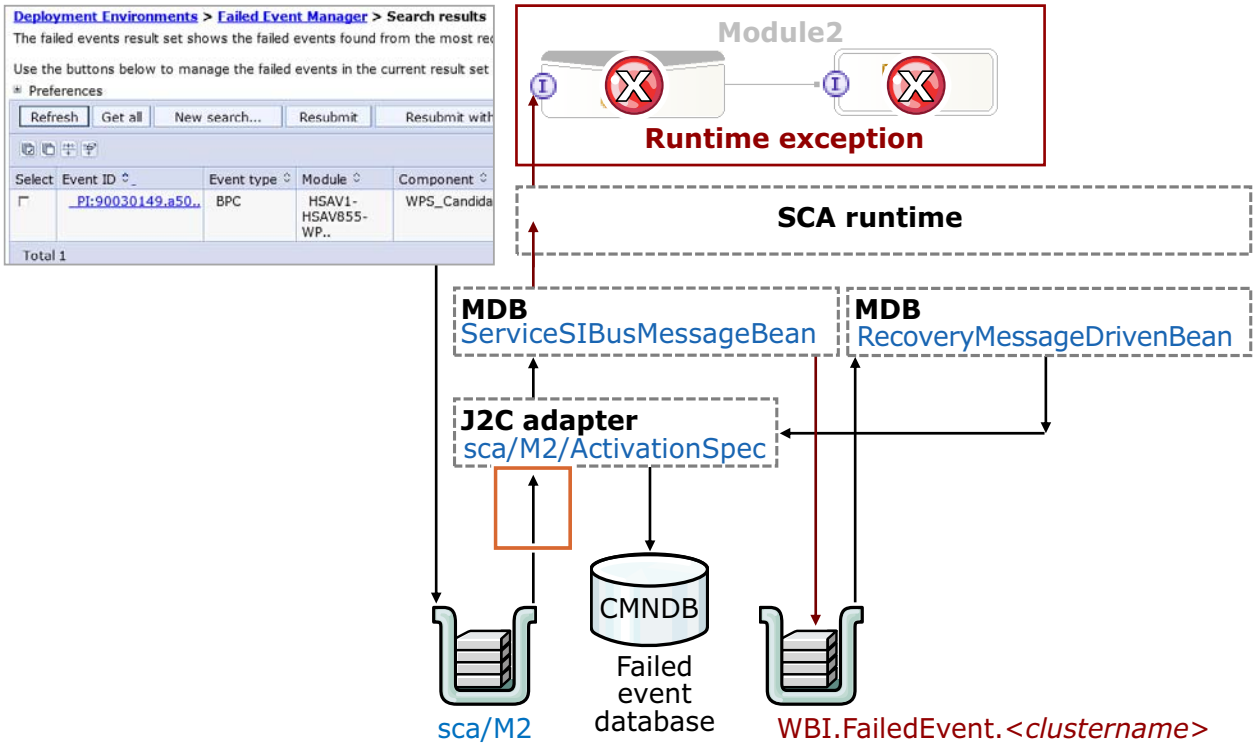
In the context of WebSphere MQ, a failed event is generated when a problem occurs (such as a data-handling exception) in the WebSphere MQ or WebSphere MQ JMS binding export or import that an SCA module uses.

© Copyright IBM Corp. 2017

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

IBM Training

Managing failed events



© Copyright IBM Corporation 2017

Figure 4-16. Managing failed events

In the example, an unavailable module can result the execution in a runtime exception. The recovery service processes the failed event message and stores in the failed event database (CMNDB). The Failed Event Manager application provides capabilities for managing these failed events.

© Copyright IBM Corp. 2017

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

IBM Training

Events, notifications, and exceptions (1 of 3)

Exception type	Interaction type	Error location	SCA runtime action	Administrator action
Business exception	Synchronous	Target component	The exception is reported to the caller as a **ServiceBusinessException**	None
Business exception	Asynchronous request/response	Target component	The exception is reported as a **ServiceBusinessException** in a subsequent response method or callback method invocation	None
System exception	Synchronous	Anywhere between the caller and the target component	The exception is reported to the caller as a **ServiceRuntimeException**	None

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-17. Events, notifications, and exceptions (1 of 3)

The following pages list exceptions that can occur during SCA module communication. Two types of exceptions can occur:

- Business exceptions
- System exceptions

When a service component is called synchronously, or does a synchronous invocation, both the service requester and the service provider run in the same thread of execution. (A thread of execution can be defined as the smallest unit of processing that an operating system can schedule.) The calling component is blocked until a response is received from the provider. As soon as the service provider is called, all processing is suspended until the thread returns a response. The target can return a response message, an exception, or nothing (in a one-way operation) to the client. If the result is an exception, it can be either a business exception or a system exception.

Business exceptions address errors within the logic of the deployed business process. Administrators cannot resolve these exceptions. They must be handled in the application by the application developer.

The administrator cannot handle system exceptions with synchronous invocation (taking place in memory), but the application can handle them.

© Copyright IBM Corp. 2017

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

IBM Training

Events, notifications, and exceptions (2 of 3)

Exception type	Interaction type	Error location	SCA runtime action	Administrator action
System exception	Asynchronous	**On the requester side** (before reaching the target component) while sending a request or one-way message	• The SIBus message that is used to carry the asynchronous interaction is put back on the **SIBus destination of the requester** • The message is represented and replayed several times • If it still cannot be processed, then it is put on the **SIBus exception destination** that is configured for the SIBus destination of the recipient	Inspect the SI bus exception queue and use a failed event administration application (business-application-specific) to receive and process the SIBus message
System exception	Asynchronous	**At the target component**, while receiving a request or one-way message	• The SIBus message that is used to carry the asynchronous interaction is put back on the **SIBus destination of the requester** • The message is represented and replayed several times • If it still cannot be processed, then it is put on the **SIBus exception destination** that is configured for the SIBus destination of the recipient	Inspect the SIBus exception queue and use a failed event administration application to receive and process the SIBus message

SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-18. Events, notifications, and exceptions (2 of 3)

When a service component is called asynchronously, the service requester and the service provider run in different threads. For an asynchronous invocation, the calling component is not blocked while the provider processes the request. The SCA system returns control to the requester immediately, allowing it to continue processing. The service requester might receive a system exception during the invocation, or the service provider might experience a business or system exception while processing the request. The service requester does not receive system exceptions that occur in the service provider thread. According to the SCA asynchronous programming model, runtime exceptions that occur at the target component are not returned to the source component.

In asynchronous invocation, the administrator can handle the system exceptions because messages that cannot be delivered are persisted in queues (data store). The administrator can inspect the queues and take adequate action to solve the exception (for example, resending the message after an unavailable component is restarted). This procedure is independent of the error location (requester or target component).

© Copyright IBM Corp. 2017

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

IBM Training

Events, notifications, and exceptions (3 of 3)

Exception type	Interaction type	Error location	SCA runtime action	Administrator action
System exception	Asynchronous	**On the response** from the target component to the requester	• The SIBus message that is used to carry the asynchronous interaction is put back on the **SIBus destination of the requester** • The message is represented and replayed several times • If it still cannot be processed, then it is put on the **SIBus exception destination** that is configured for the SIBus destination of the recipient	Inspect the SIBus exception queue and use a failed event administration application (business-application-specific) to receive and process the SIBus message

© Copyright IBM Corporation 2017

Figure 4-19. Events, notifications, and exceptions (3 of 3)

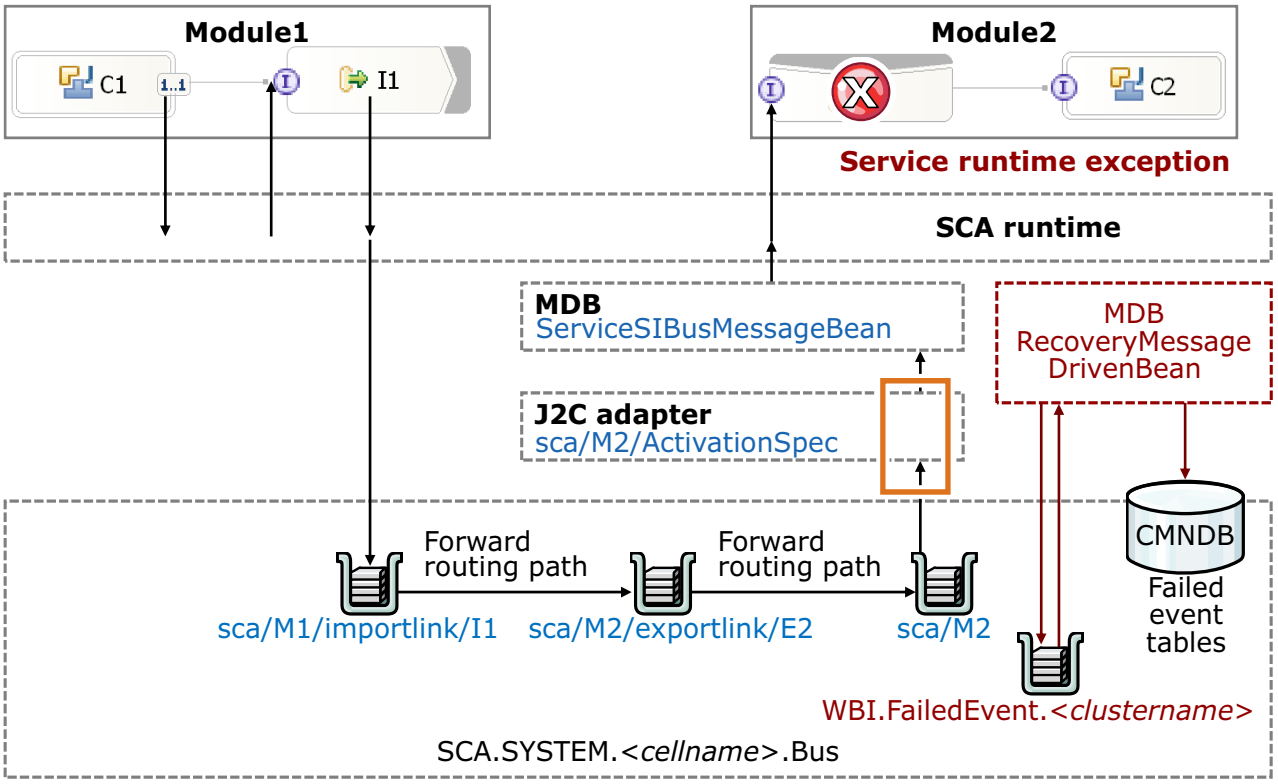
The administrator can handle the exception for error location “on the response”.

© Copyright IBM Corp. 2017

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

IBM Training

SCA asynchronous intermodule failures



© Copyright IBM Corporation 2017

Figure 4-20. SCA asynchronous intermodule failures

When a runtime exception occurs, the failed message is sent several more times (the number can be configured at the destination level; the default is 5). If it still cannot be processed, then it is put on the SIBus exception destination that was configured for the SIBus destination of the recipient.

The recovery service message-driven bean reads the message from the failed event queue and persists it to the failed event database tables in CMNDB.

© Copyright IBM Corp. 2017

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

IBM Training

Recovery service runtime (1 of 3)

- IBM Process Server includes the recovery service, which is a built-in service that monitors for failed operations between SCA components
- The SCA failed asynchronous interactions are redirected to failed event destinations
  - Destinations are created on the SCA system bus: **SCA.SYSTEM.<*cellname*>.Bus**
  - One failed event destination per deployment target
    (for a single server): **WBI.FailedEvent.<*nodename*>.<*servername*>**
    (for a cluster): **WBI.FailedEvent.<*clustername*>**
- The recovery message-driven bean reads failed events from the failed queue
- Failed events are saved in the failed event database

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-21. Recovery service runtime (1 of 3)

The bulleted list in the slide lists the sequence of actions that are taken to handle a failed event:

1. The message is redirected to a failed destination. For IBM Process Server, the destination has a dedicated name, including server or cluster scope.
2. The MDB of the recovery service reads the failed events from this failed event destination and persists them in the failed event schema in the common database (CMNDB).
3. Recovery or failed event processing takes place only in this manner for IBM Process Server by using WebSphere Platform Messaging. Events are routed to the system exception queue on the SCA system bus named:
   `_SYSTEM.Exception.Destination.node.<serverName>-SCA.SYSTEM.<cellName>.Bus`

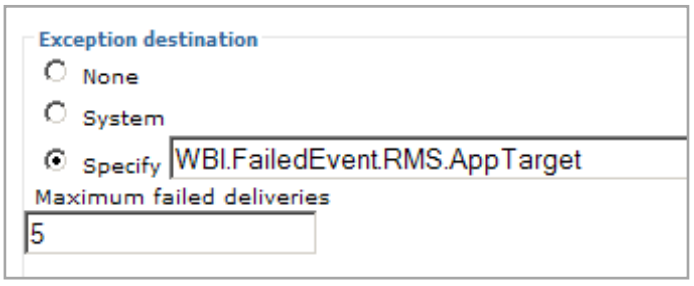
© Copyright IBM Corp. 2017

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

IBM Training

Recovery service runtime (2 of 3)

- Destinations that are created on the SCA system bus are configured to route failed messages to the recovery exception destination
- When a system failure occurs, the failed message is captured in the exception destination
- IBM Process Server recovery feature generates a failed event that represents the system error and stores it in the failed event database
- Three database tables:
  - FailedEvents
  - FailedEventBOTypes
  - FailedEventMessages



© Copyright IBM Corporation 2017

Figure 4-22. Recovery service runtime (2 of 3)

© Copyright IBM Corp. 2017

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

IBM Training

Recovery service runtime (3 of 3)

- Works only with asynchronous interactions
  - Asynchronous reliability qualifier needs to be “assured”
  - Enabled by default in IBM Integration Designer

```
"<references>
    <scdl:referenceQualifier
      Xsi:type="scdl:Reliability"
     value="assured"/>
    ...
    </references>"
```

- Resubmission of failed events
  - Submit the asynchronous interaction messages back to their original destination
  - Requests are resubmitted to the target component
  - Responses are resubmitted to the component that made the request

SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-23. Recovery service runtime (3 of 3)

Recovery of failed events by the recovery service runtime takes place only if communication is **asynchronous** and the quality of service (QoS) qualifier is **assured**. For most asynchronous interactions in IBM Integration Designer, this setting is default. The example on this page shows the qualifier for an SCDL reference.

One option to handle failed events is to resubmit them to the target component.

A service component is defined in a Service Component Definition Language (SCDL) file. A component file that is created with SCDL is roughly analogous to an EJB deployment descriptor in that it defines the interface, implementation, and several quality of service requirements of an SCA component.

© Copyright IBM Corp. 2017

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

4.3.SCA bindings

© Copyright IBM Corp. 2017

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

IBM Training

SCA bindings

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 4-24. SCA bindings

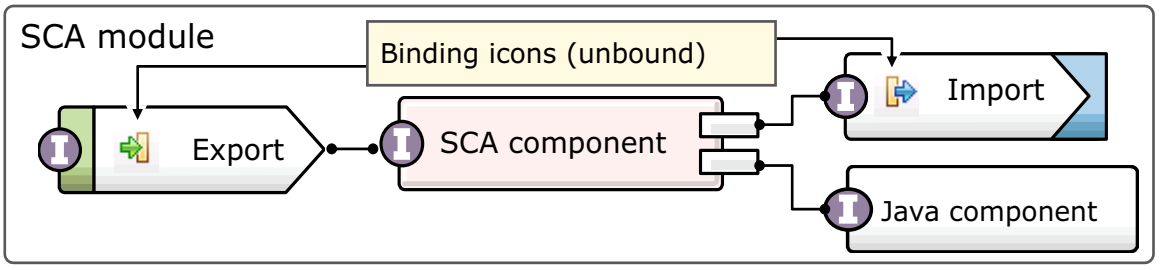
© Copyright IBM Corp. 2017

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

IBM Training

Incoming and outgoing interactions with external components

- Exports process **incoming** requests from outside SCA modules
  - The reference of the export is associated with a specific interface type
  - The export is connected to an SCA component through a wire
- Imports process **outgoing** requests to components outside SCA modules
  - The import contains a specific interface type
  - The interface of the import is the target of a reference through a wire
- Bindings determine how imports and exports interact with components outside a module
  - Bindings specify the means of transporting the data (protocol)



SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-25. Incoming and outgoing interactions with external components

An export component is responsible for exposing a mechanism that the client use to invoke the services of that module. An import component is responsible for describing how to call some service that exists externally. Both the import and export components have something in common: they shield the heart of the module from the need to know details of how to interact with external applications.

Every import has an associated interface type. As a result, the import can be wired to any other component that has a matching interface type. The caller of the import has no knowledge that it is calling an import, only that it is calling “just another” SCA component that has a specific interface type.

An export component has a reference that is associated with a specific interface type. As a result, the export can be the source of a wire or connection to some other SCA component. The called SCA component has no knowledge that an export component called it.

Imports and exports require binding information, which specifies the means of transporting data. Bindings for imports and exports have different purposes. An **import binding** describes the specific way that an external service is bound to an import component. An **export binding** describes how that export (or service) is published or made available to clients outside the module. Binding types are associated with imports and exports by using the assembly editor.

© Copyright IBM Corp. 2017

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

The binding type determines what type of client is supported; for example, a web service binding makes the service available to any web-based client, while an SCA binding makes it available to other SCA modules. Import bindings tell the SCA runtime processes how to access an external service. For example, if you publish a service with an SCA export binding, an import with a JMS binding is not able to successfully call it. The SCA binding is the default binding; if no binding is specified for an export, the runtime process assumes an SCA binding when the module is deployed. You can use an import with an SCA binding to access a service in another SCA module. You can use a web service import binding to bind an external web service to an import. The import binding does not have to be defined at the development time; it can be defined at deployment time.

You can create more than one export for a component or import, so you can effectively export the component or import with more than one binding type. You can even choose to export one set of interfaces with one binding type and another set of interfaces with another binding type. Because the SCA bindings are easy to use (for example, the SCA export binding requires no binding information), you might choose to use them for prototyping work, and replace or augment them as the prototype evolves.

© Copyright IBM Corp. 2017

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

IBM Training

Export binding types

- When an export is created in a mediation module or integration module, it must be bound to a transport type
- Transport types include:
  - Service Component Architecture (default)
  - Web service
  - Hypertext Transfer Protocol (HTTP)
  - Enterprise JavaBeans (EJB)
  - Java Message Service (JMS)
  - Generic JMS (for independent vendor JMS providers)
  - WebSphere MQ
  - WebSphere MQ JMS
  - Enterprise information system (EIS)
- Several export components with different binding types can be exposed, and many callers can use them

SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-26. Export binding types

When an export component is added to the SCA assembly diagram, it must be bound to a particular transport binding. In the following pages, you examine each of the export binding types and options in detail. You should note that the component does not have more than one export with the same binding type. This approach is a good practice.

© Copyright IBM Corp. 2017

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

IBM Training

Import binding types

- When an import is created in a mediation module or integration module, it must be bound to a particular transport
- Transport types include:
  - Service Component Architecture (default)
  - Web service
  - Hypertext Transfer Protocol (HTTP)
  - Enterprise JavaBeans (EJB)
  - Java Message Service (JMS)
  - Generic JMS (for independent vendor JMS providers)
  - WebSphere MQ
  - WebSphere MQ JMS
  - Enterprise information system (EIS)
- Modules can also contain several import components with different binding types so they can call various services

© Copyright IBM Corporation 2017

Figure 4-27. Import binding types

An import component must also be bound to a particular transport binding. The import hides the mechanics of how the service call is implemented. All the caller sees is an exposed service interface. The actual network or invocation protocol is hidden. Import bindings are covered later in this unit. It is a good practice that the component must not call more than one import with the same binding type.

© Copyright IBM Corp. 2017

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

IBM Training

JMS bindings (1 of 2)

- JMS is the Java Message Service specification, which describes how a Java application can send and receive messages
- Business value of JMS bindings:
  - Uses applications that are asynchronous or need assured delivery
  - Supports non-IBM JMS implementations
  - Easy integration with WebSphere MQ and WebSphere Platform Messaging
  - Supports configurable message correlation (inbound and outbound)
  - Supports event sequencing for exports
  - Supports configurable reply connection for imports
  - Supports both publish/subscribe and queue-based messaging
- JMS has three types of bindings:
  - JMS/MQ
  - JMS (WebSphere Platform Messaging and WebSphere Enterprise Service Bus)
  - Generic JMS (for third-party JMS providers)

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-28. JMS bindings (1 of 2)

Java Message Service is a specification that describes how a Java application can send and receive messages. It supplies an API and semantics that are vendor independent. IBM implemented JMS in the service integration bus (WebSphere Platform Messaging), WebSphere Enterprise Service Bus, and WebSphere MQ. If a Java application exists that is using JMS, IBM Process Server can send messages to it and receive messages from it. JMS binding supports the JMS message class plus all five JMS message subclasses: `TextMessage`, `BytesMessage`, `ObjectMessage`, `StreamMessage`, and `MapMessage`.

The JMS binding closely aligns with JMS/MQ binding capabilities. JMS bindings support:

- Configurable correlation schemes for both imports and exports
- Event sequencing for exports
- Configurable reply connection for imports

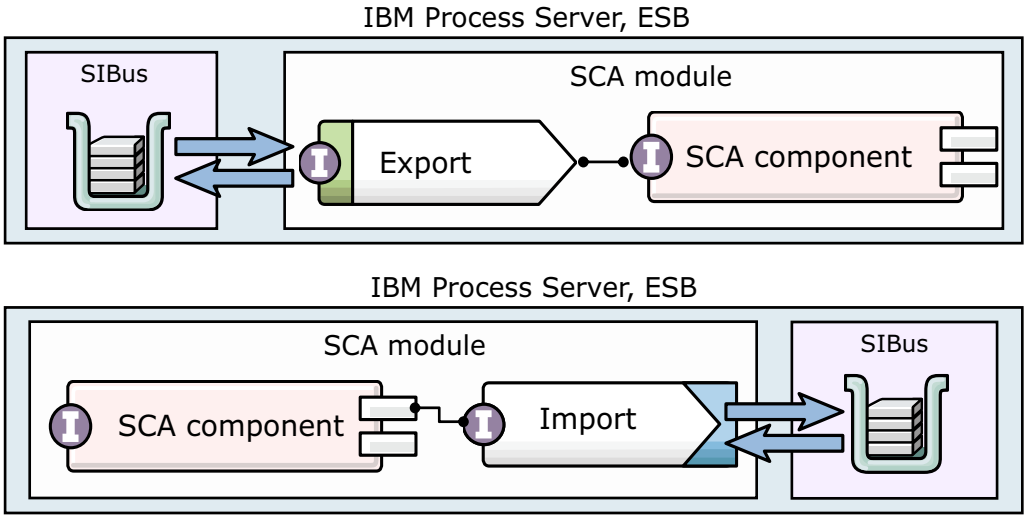
© Copyright IBM Corp. 2017

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

IBM Training

JMS bindings (2 of 2)

- Providing an export component with JMS binding allows an IBM Process Server or ESB module to watch an associated queue or topic and, when a message arrives, to receive the message and process it
- Providing an import component with JMS binding results in the production of a message when the service interface is called



© Copyright IBM Corporation 2017

Figure 4-29. JMS bindings (2 of 2)

JMS supports both point-to-point queuing and publish/subscribe messaging through topics. An SCA export with a JMS binding monitors an associated queue or topic, and when a message arrives, the module receives the message and processes it. An SCA import can also have a JMS binding, which results in producing a message when the service interface is called.

© Copyright IBM Corp. 2017

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

IBM Training

Generic JMS bindings

- Generic JMS binding supports independent vendor JMS providers
  - Other web application servers
- Behavior is comparable to JMS and JMS/MQ bindings
  - Supports point-to-point and publish/subscribe styles
  - Correlation schemes and event sequencing supported
  - Security by using authentication aliases
  - Obeys the SCA programming model
- Support is generic
  - No provider-specific connectivity options
  - Provider must be preconfigured
  - Limited capability to generate resources at deployment

© Copyright IBM Corporation 2017

Figure 4-30. Generic JMS bindings

Generic JMS is a standard API for sending and receiving messages in a vendor-neutral manner. Generic JMS allows components to create, send, receive, and read messages regardless of the messaging system that is used, including non-IBM messaging providers. Generic JMS resources for WebSphere Application Server are automatically configured. Manual setup is required for independent vendor JMS resources.

WebSphere MQ can be used in IBM Process Server in multiple ways. They include WebSphere MQ bindings, MQ/JMS bindings, WebSphere MQ Link, and generic JMS bindings. It is preferable to use the WebSphere MQ bindings or the MQ/JMS bindings because they achieve the best synergy and performance.

© Copyright IBM Corp. 2017

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

IBM Training

Import and export resources

Most imports and exports use the following components:

- **Data binding** is a map between a native data format and a business object
  - The `DataBinding` Java class takes a stream of data and builds a business object or takes a business object and builds a stream of data
- **Function selectors** assign incoming messages or requests to the correct service operation
- **Data handlers** use data bindings or function selectors to transform data from one type to another

The generated resources are normally adequate

- For the remaining cases, import and export binding resources can be customized, depending on the binding type

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-31. Import and export resources

IBM Process Server provides a technology that is called `DataBinding`. `DataBinding` is a Java interface that is called at run time in IBM Process Server. When IBM Process Server receives a message through an export component, the data of the message must be transformed into a business object. An instance of a `DataBinding` implementation is called and has the physical data that is passed to it. The `DataBinding` is then responsible for building and populating the business object that is returned.

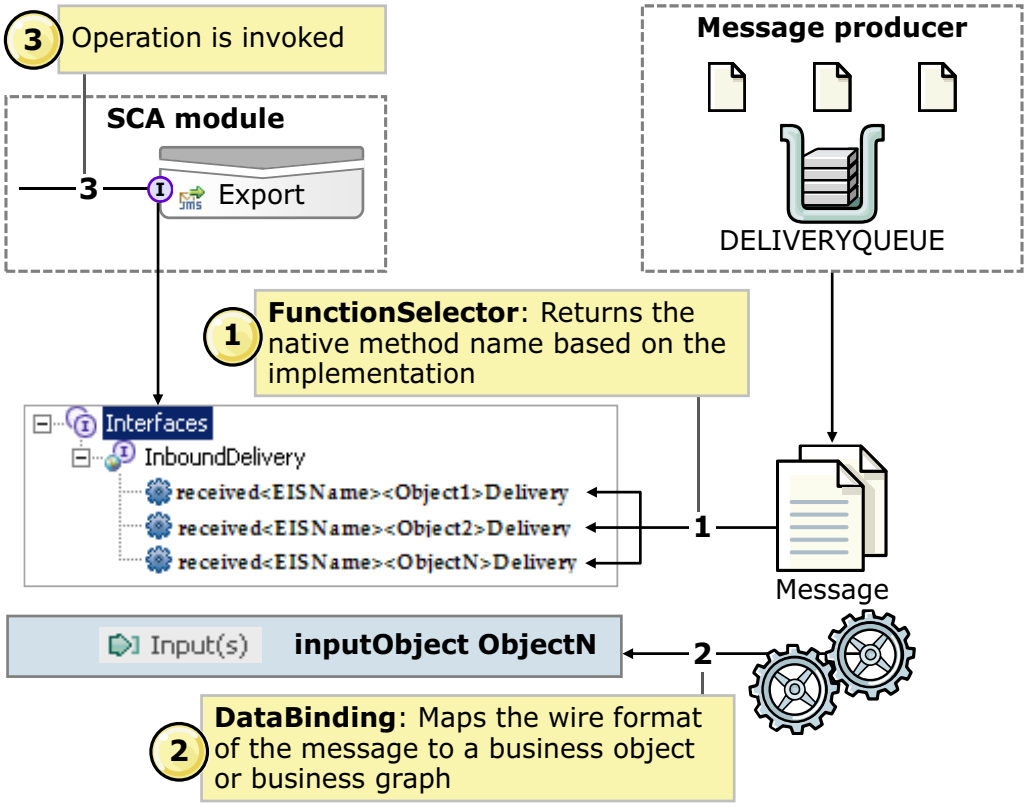
When an import component is reached, the `DataBinding` again has a role. The `DataBinding` is called and supplied with an instance of a business object. The logic of the `DataBinding` must then serialize the content of the business object into a physical representation that is used as the body of the message. Although a custom `DataBinding` can easily be written, IBM supplies starter or generic `DataBinding` implementations that are sufficient in most cases.

© Copyright IBM Corp. 2017

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

IBM Training

Binding resource example: JMS binding



SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-32. Binding resource example: JMS binding

A JMS message comes into IBM Process Server through a specific operation invocation, and it gets written onto a respective queue (interface binding). The MDB picks up the message, and based on the Target Function Name property value in the header, the server knows which operation receives this message (method binding). Before the IBM Process Server can work with this data, this data must be transformed into SDO (data binding).

© Copyright IBM Corp. 2017

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

4.4.SCA quality of service qualifiers

© Copyright IBM Corp. 2017

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

IBM Training

SCA quality of service qualifiers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 4-33. SCA quality of service qualifiers

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

IBM Training

Quality of service qualifiers

- Qualifiers are specifications that define a set of communication characteristics that an application requires for transmission priority, level of route reliability, transaction management, and security level
- You can specify quality of service qualifiers in the assembly editor
- These qualifiers define how much management must be provided for a component at run time
- Think about quality of service qualifiers when you are designing your error handling strategy
- You can set qualifiers on:
  - Interfaces
  - References
  - Implementations

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-34. Quality of service qualifiers

Qualifiers are specifications that define a set of communication characteristics that an application requires for transmission priority, level of route reliability, transaction management, and security level. You can specify quality of service qualifiers in the assembly editor. These qualifiers define how much management must be provided for a component at run time.

Usually, you should think about quality of service qualifiers when you are designing your error handling strategy. You can set qualifiers on:

- Interfaces
- References
- Implementations

The system generates qualifiers that promote the propagation of transactions and reliable asynchronous message delivery where possible. You can change these preset qualifiers in the assembly editor. In some cases, the implementation requires that some qualifiers are set a certain way. In such cases, the system either blocks you from making the change or generates an error message.

Some implementations determine how qualifiers must be set. In these situations, the system generates the qualifiers for you. If you change the values of those generated qualifiers, you risk

© Copyright IBM Corp. 2017

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

creating validation errors. If you are working with Java components, you need to set the qualifiers yourself.

© Copyright IBM Corp. 2017

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

IBM Training

Qualifiers that are supported in IBM Integration Designer (1 of 3)

Location	Qualifier	Description
Interface	Data validation qualifier	Confirms that the data that is passed in to an operation (usually a business object instance) matches the XSD type inputs of the operation
	Event sequencing	Places a control on the order in which the runtime environment processes events
	Join activity session	Specifies whether the target service can join a client-propagated activity session
	Join transaction	Specifies whether the target service can join a client-propagated transaction
	Security permission	Allows access only to users who are granted for the particular security role

© Copyright IBM Corporation 2017

Figure 4-35. Qualifiers that are supported in IBM Integration Designer (1 of 3)

© Copyright IBM Corp. 2017

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

IBM Training

Qualifiers that are supported in IBM Integration Designer (2 of 3)

Location	Qualifier	Description
Reference	Asynchronous invocation	Determines whether asynchronous invocations can occur as part of a client transaction
	Asynchronous reliability	Three qualifiers (reliability, request expiration, and response expiration) can be used to determine the reliability of asynchronous message delivery
	Suspend activity session	Determines whether the client propagates the activity session to the referenced component
	Suspend transaction	When the invocation is synchronous, the qualifier determines whether the client propagates the transaction to the referenced component

© Copyright IBM Corporation 2017

Figure 4-36. Qualifiers that are supported in IBM Integration Designer (2 of 3)

© Copyright IBM Corp. 2017

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

IBM Training

Qualifiers that are supported in IBM Integration Designer (3 of 3)

Location	Qualifier	Description
Implementation	Activity session	Determines whether the processing of the component is executed under an activity session, or another unit-of-work scope to a global transaction
	Security identity	Provides a logical name for the identity under which the implementation executes at run time
	Transaction	Determines the logical unit of work that the processing of the component executes

© Copyright IBM Corporation 2017

Figure 4-37. Qualifiers that are supported in IBM Integration Designer (3 of 3)

© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing

- Inbound events can be sequenced
  - Ensures that events are processed in the correct order; for example, the withdraw money can happen only after the deposit
  - For components that are invoked by using the SCA asynchronous invocation style
- User can choose keys to sequence the events
- Supports event sequencing generically for all component kinds
- An event sequencing qualifier annotation is on the interface of the component

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-38. Event sequencing

Sometimes you need to ensure that the data is processed in order. For example, you want to be sure that an employee record is created before any update event for that employee record is processed. Clients determine the sequence in which events are sent to a component, but controlling the sequence in which clients contribute events to a queue is not always enough. The event that would add an employee record might process slowly while an update event might go through quickly. You can establish a qualifier for such events to ensure that they are processed in the order they are received.

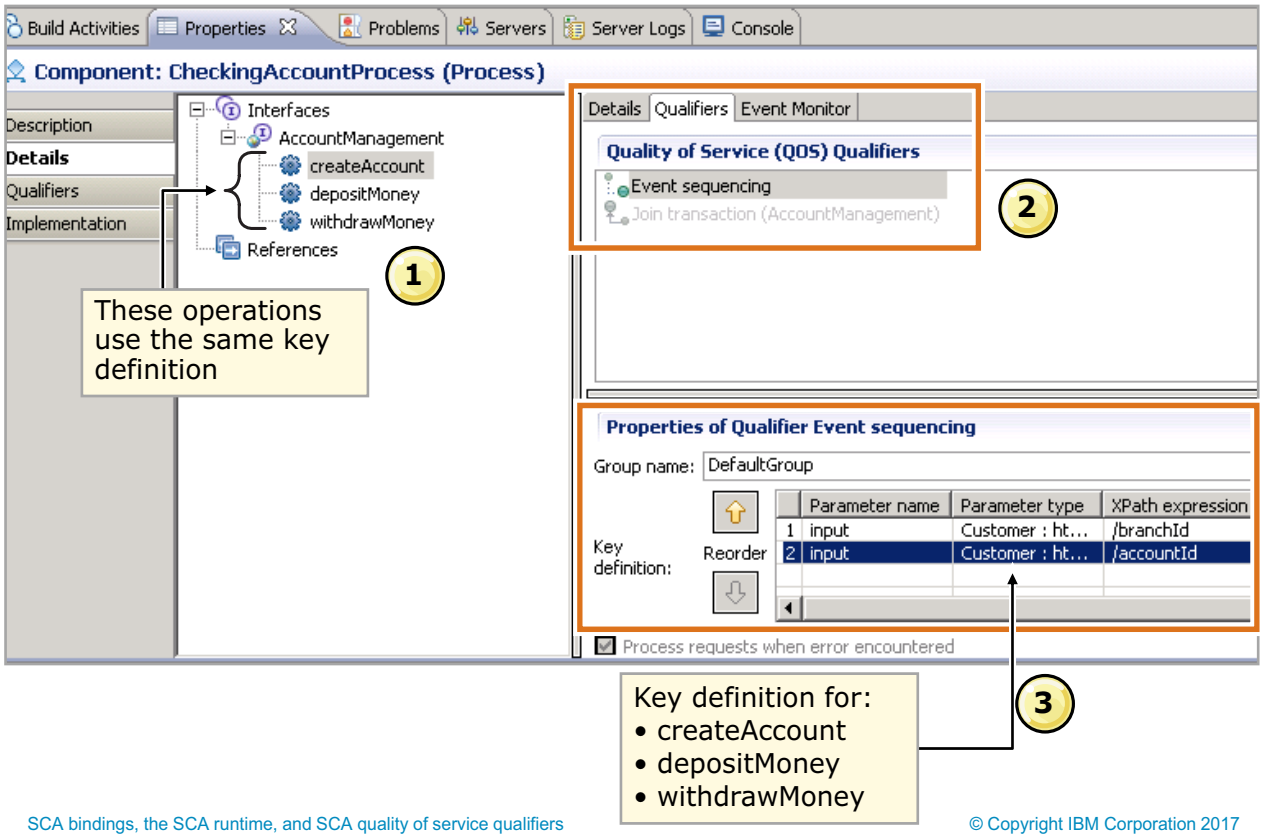
The event sequencing qualifier ensures that events are processed in order and that one event completes its processing before the next one is processed. The event sequencing qualifier places a control on the order in which the runtime environment processes events. You specify a group of one or more operations and a key. When two or more events are received that invoke any of those operations that have the same key, the event sequencing qualifier ensures that they are processed in the order they are received. Without a keyed qualifier, the multithread runtime environment does not necessarily process events in the order that they are received.

© Copyright IBM Corp. 2017

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

IBM Training

Setting the event sequencing qualifier



SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-39. Setting the event sequencing qualifier

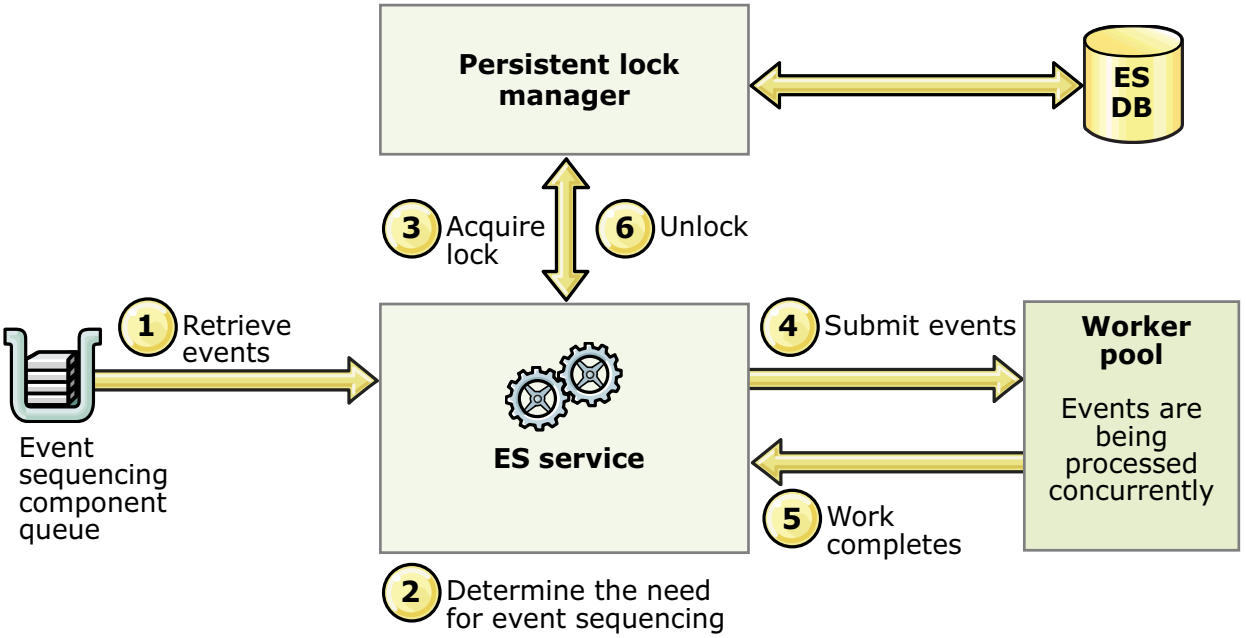
1. In the Properties view, click the Details tab, which shows the component's interfaces and references.
2. Click Qualifiers, click Add, and select Event Sequencing from the list provided. The event sequencing qualifier is added to the operation. When you select that qualifier, you can see its properties in a subordinate view.
3. Create or select a name for the group to which this operation belongs. Events for all the operations in the same group are considered together for sequencing. They are kept in a different sequence from events for operations in other groups. Events in the same group that have the same key value are delivered in the order that they arrive. In the Properties of Qualifier Event sequencing view, click in the table cell for Parameter name and select the relevant parameter from the list provided. The parameters are business objects or simple types.

© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing (ES) runtime



SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-40. Event sequencing (ES) runtime

The event sequencing runtime triggers a pending operation to acquire a lock before it is executed. When the execution of the business logic completes, the event releases the lock.

1. When a service receives events, those events join a queue.
2. Determine the need to ensure that events are processed in order and that one event completes its processing before the next one is processed.
3. Those events that must be sequenced first acquire a lock. Events are required to acquire a lock first. If an event fails to acquire a lock, the event is unqueued.
4. Target component processes the event.
5. Transaction completes
6. After the target component processes the event successfully, the lock is released. Detect that the target component finished processing the event and grant the lock to the next blocking event. Events with same key are sequenced. Work completion detection is required for ensuring component isolation, and when the lock is released for current event, it is granted to next event in the queue.

© Copyright IBM Corp. 2017

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

IBM Training

Which components need event sequencing?

- When multiple components are wired together, the qualifier might be needed on multiple components:
  - For **one-way** operations, the qualifier is required on all components in the chain
  - For **request/response** operations, the qualifier is required only on the first component in the chain (**`compA`**)
  - The same rules apply even if components are chained across modules

© Copyright IBM Corporation 2017

Figure 4-41. Which components need event sequencing?

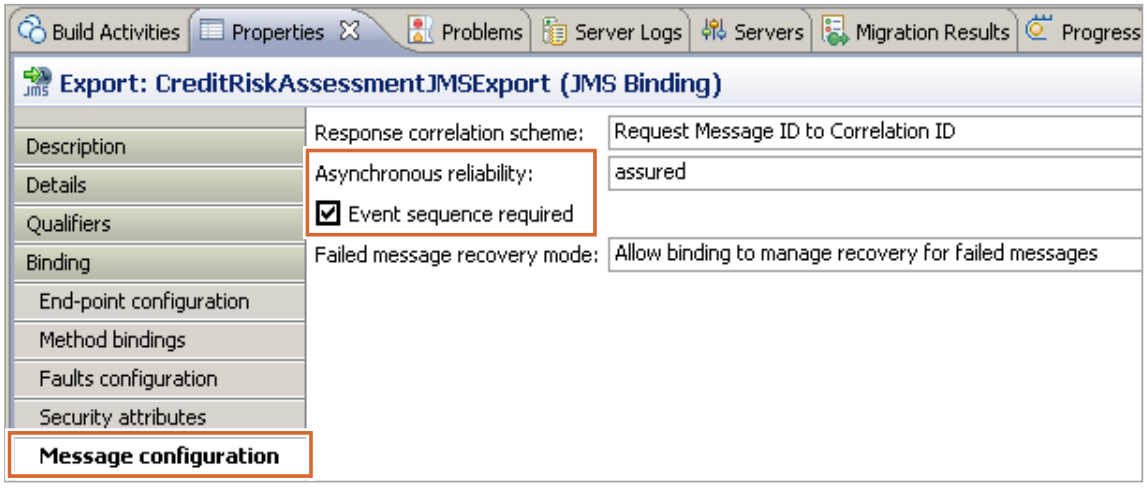
© Copyright IBM Corp. 2017

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

IBM Training

Enable event sequencing: JMS exports

- Event sequencing for JMS export bindings is supported in a clustered environment only when the destinations are not partitioned
  - For event sequencing to function in a network deployment environment with clusters, only one active messaging engine per cluster is possible
  - For event sequencing in a network deployment environment with servers that are not in a cluster, each server can have an active messaging engine



© Copyright IBM Corporation 2017

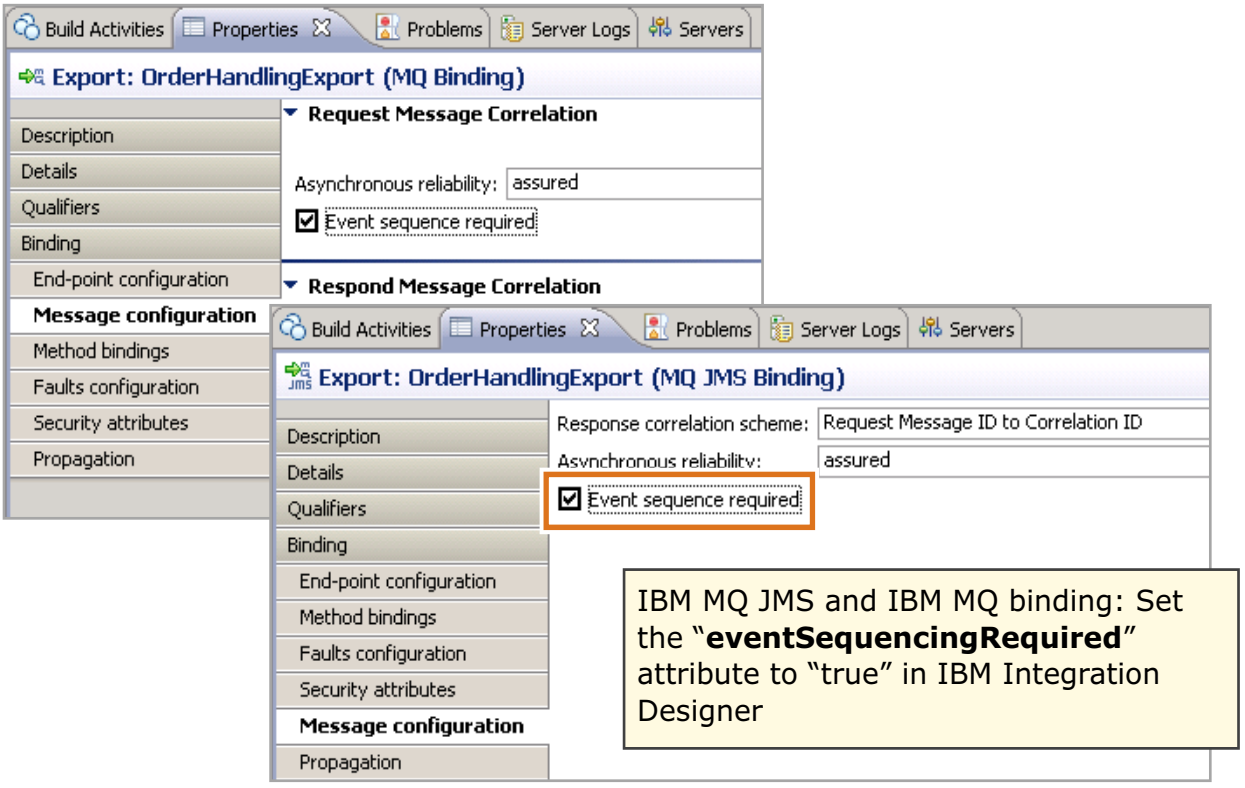
Figure 4-42. Enable event sequencing: JMS exports

© Copyright IBM Corp. 2017

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

IBM Training

Enable event sequencing: IBM MQ JMS and IBM MQ exports



© Copyright IBM Corporation 2017

Figure 4-43. Enable event sequencing: IBM MQ JMS and IBM MQ exports

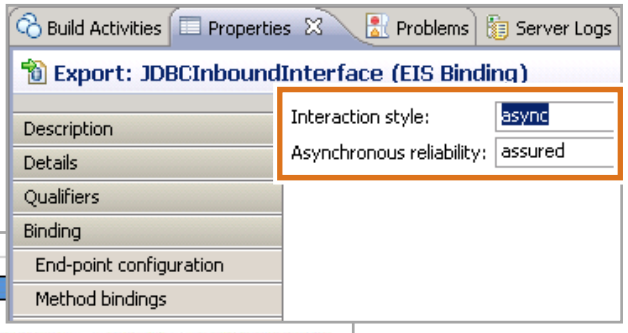
© Copyright IBM Corp. 2017

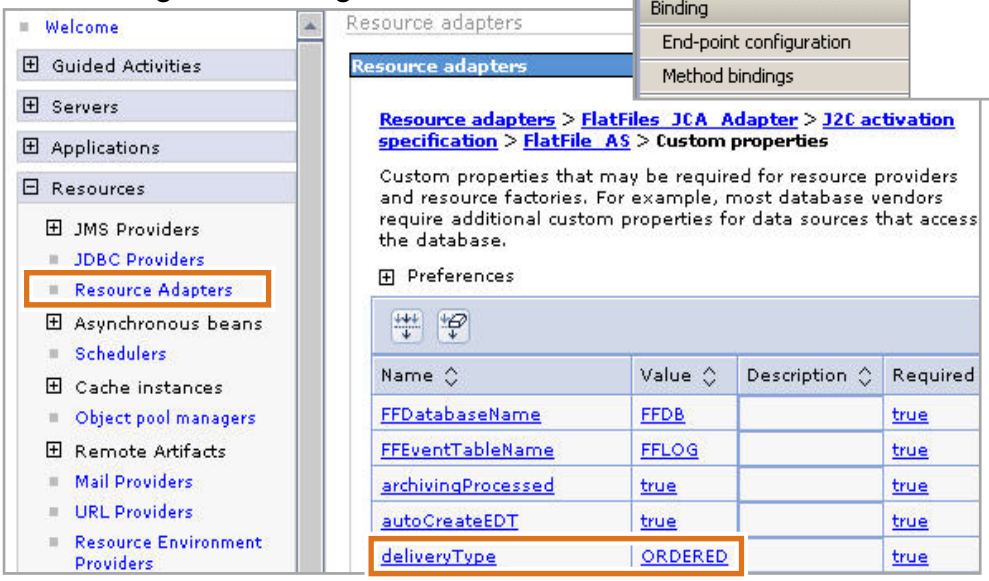
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Enable event sequencing: EIS export

- EIS binding
  - It is advised that you set the "**adapterInteractionStyle**" attribute to "async" in IBM Integration Designer





- Set "**DeliveryType**" property to "**ORDERED**" in the resource adapter ActivationSpec in the administrative console

© Copyright IBM Corporation 2017

Figure 4-44. Enable event sequencing: EIS export

With EIS binding, it is a good practice that you set the "adapterInteractionStyle" attribute to "async" in IBM Integration Designer.

Set the "DeliveryType" property to "ORDERED" in the resource adapter ActivationSpec in the administrative console.

In general, if event sequencing is required in a network deployment environment, the module that has the export must be moved:

- To a stand-alone server
- Or to a cluster that has only one active server that is enabled for high availability

© Copyright IBM Corp. 2017

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

IBM Training

esAdmin command-line interface

- The **esAdmin** command-line utility can be used to list locks, delete locks, and break deadlock:

```
$WAS/bin/esAdmin.sh[bat] -help
    Usage: esAdmin [-h hostname] [-p soapPortNumber] method
      [parameters]
    Available methods are:
     listAll
     listLocks moduleName
     listLocks moduleName componentName
     listLocks moduleName componentName methodName
     deleteLocks moduleName
     unlock lockId
```


© Copyright IBM Corporation 2017

Figure 4-45. esAdmin command-line interface

The `esAdmin` command can list and delete all locks currently managed by the lock manager. When listing locks, you can list all locks or a small subset that is filtered based on the module, component, or method. This command can also be used to release an active lock in a deadlock situation; after the lock is released, it is granted to the next queued request.

The `list` command lists lock table entries for given criteria.

The `delete` command deletes entries from the lock table; the server must *not* be running.

The `unlock` command deletes the current entry; it grants a lock to the next queued event.

The `MessageID` column can be used to correlate the rows with messages on destination.

BPEL limitations are as follows:

- To initiate the receive activities (`createInstance` set to yes), the event sequencing qualifier must be declared.
- If multiple receive activities are initiated, only one can have an event sequencing qualifier.
- Deadlocks are not detected statically or dynamically.
- The command-line utility is supported to break deadlocks.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain how the SCA runtime handles synchronous and asynchronous communication
- Describe the function of the Failed Event Manager
- Describe the SCA import and export bindings
- Identify the components of an SCA import or export
- Describe the role of SCA quality of service qualifiers in transmission priority and level of route reliability
- Explain the purpose and value of interface qualifiers, implementation qualifiers, and reference qualifiers

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-46. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: An administrator can resubmit an asynchronous business exception by using the Failed Event Manager.
2. True or False: The security permission qualifier is used to provide the identity under which a service implementation executes at run time.
3. True or False: Event sequencing for JMS export bindings is supported in a clustered environment only when the destinations are not partitioned.

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-47. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: An administrator can resubmit an asynchronous business exception by using the Failed Event Manager.
   The answer is False. An administrator ***cannot*** use the Failed Event Manager to resubmit an asynchronous business exception.
2. True or False: The security permission qualifier is used to provide the identity under which a service implementation executes at run time.
   The answer is False. The security permission qualifier allows access only to users who are granted for the particular security role.
3. True or False: Event sequencing for JMS export bindings is supported in a clustered environment only when the destinations are not partitioned.
   The answer is True.

SCA bindings, the SCA runtime, and SCA quality of service qualifiers

© Copyright IBM Corporation 2017

Figure 4-48. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Working with SCA bindings and qualifiers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 4-49. Exercise 3

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

IBM Training

Exercise objectives

- Implement a WebSphere MQ import binding in IBM Integration Designer
- Use the WebSphere Adapter for Flat Files to implement an EIS import binding
- Test bindings in the IBM Integration Designer integrated test environment

© Copyright IBM Corporation 2016

Figure 4-50. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

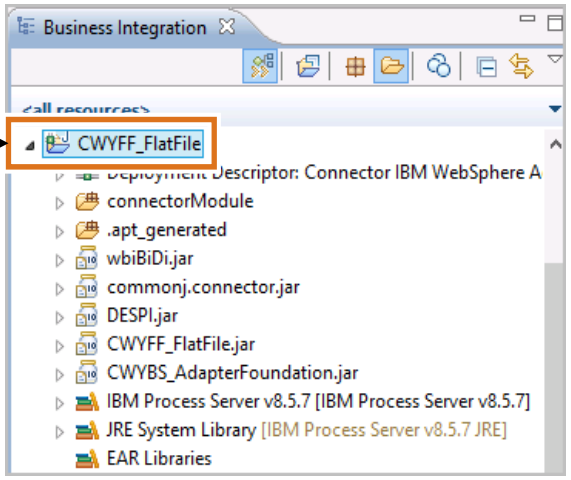
Components that are required for Exercise 3

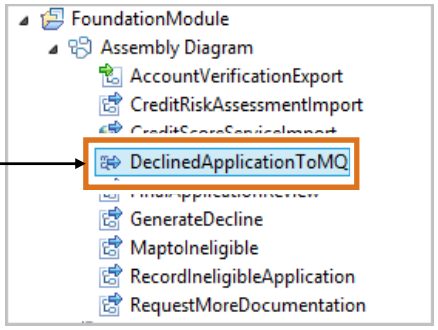
Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **AccountVerification BPEL**

New components that you create in the lab:

1. **DeclinedApplicationToMQ component**
2. **CWYFF_FlatFile adapter component**





© Copyright IBM Corporation 2017

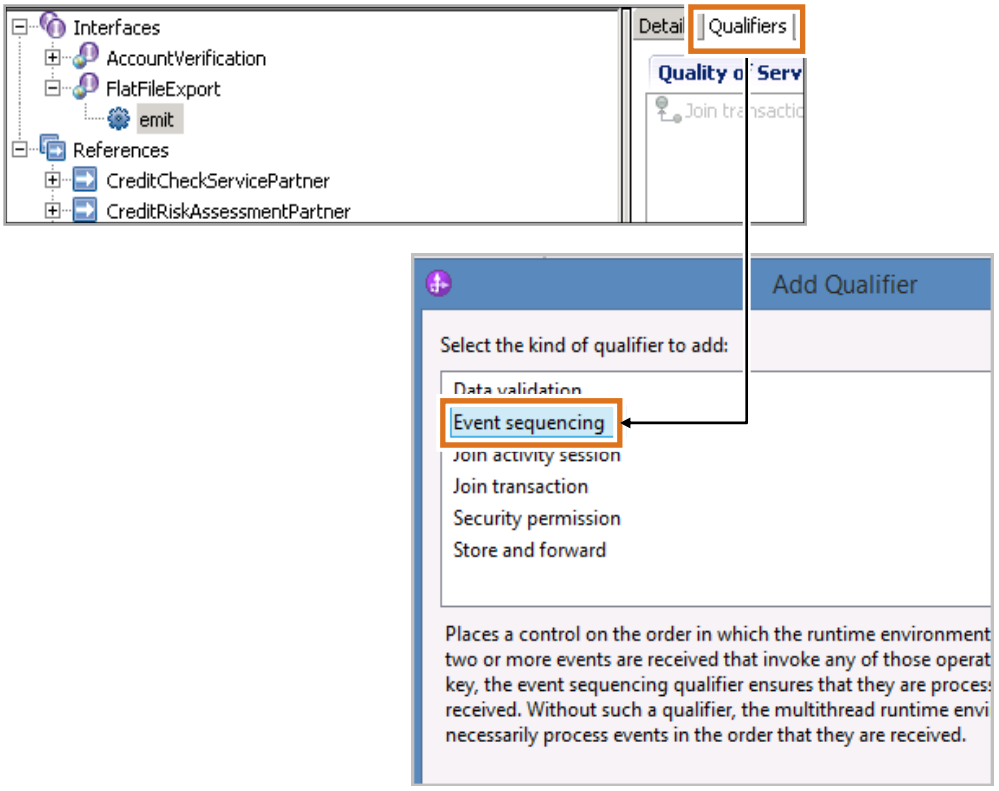
Figure 4-51. Components that are required for Exercise 3

© Copyright IBM Corp. 2017

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

IBM Training

Enable the event sequencing qualifier in Exercise 3



SCA bindings, the SCA runtime, and SCA quality of service qualifiers © Copyright IBM Corporation 2017

Figure 4-52. Enable the event sequencing qualifier in Exercise 3

© Copyright IBM Corp. 2017

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

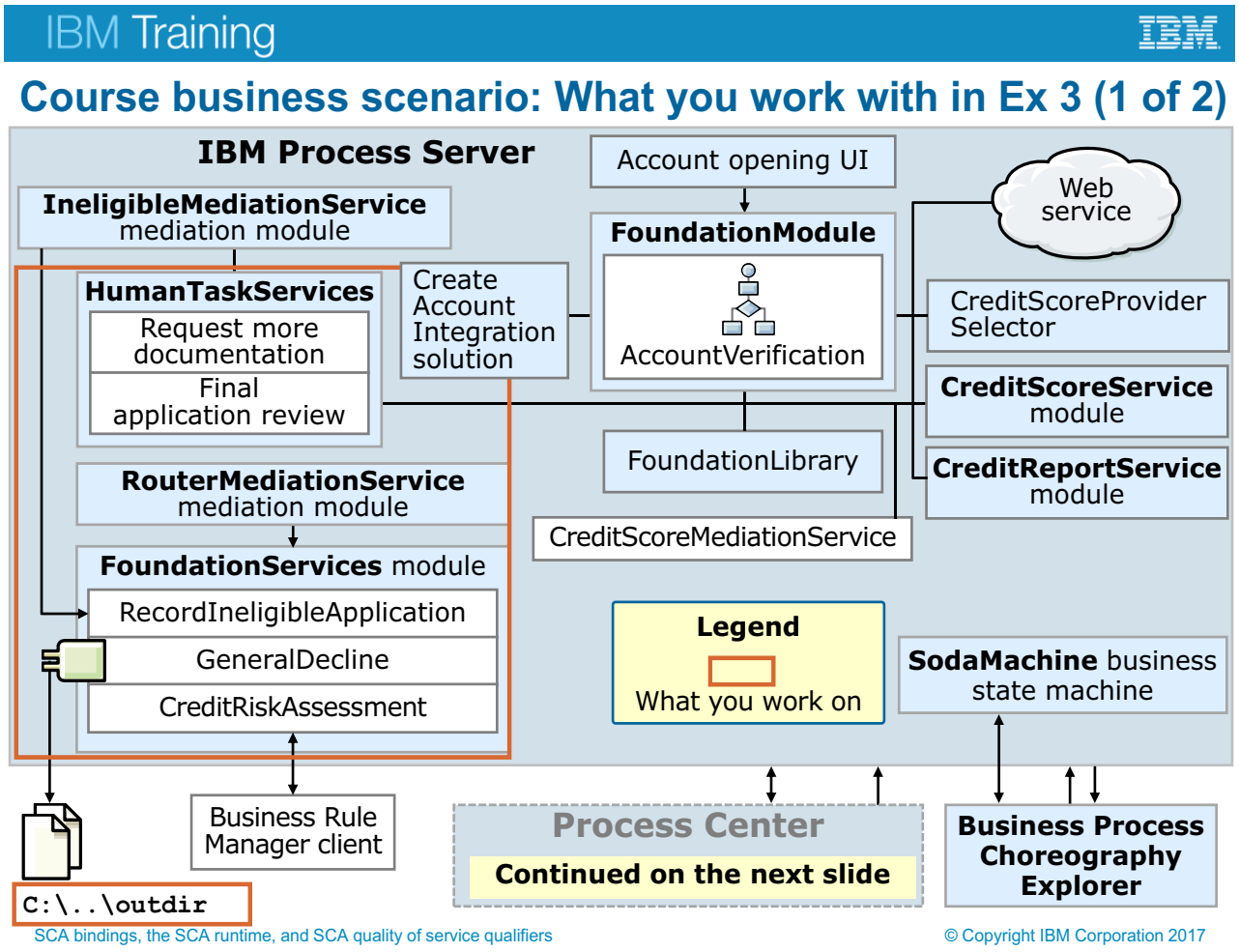


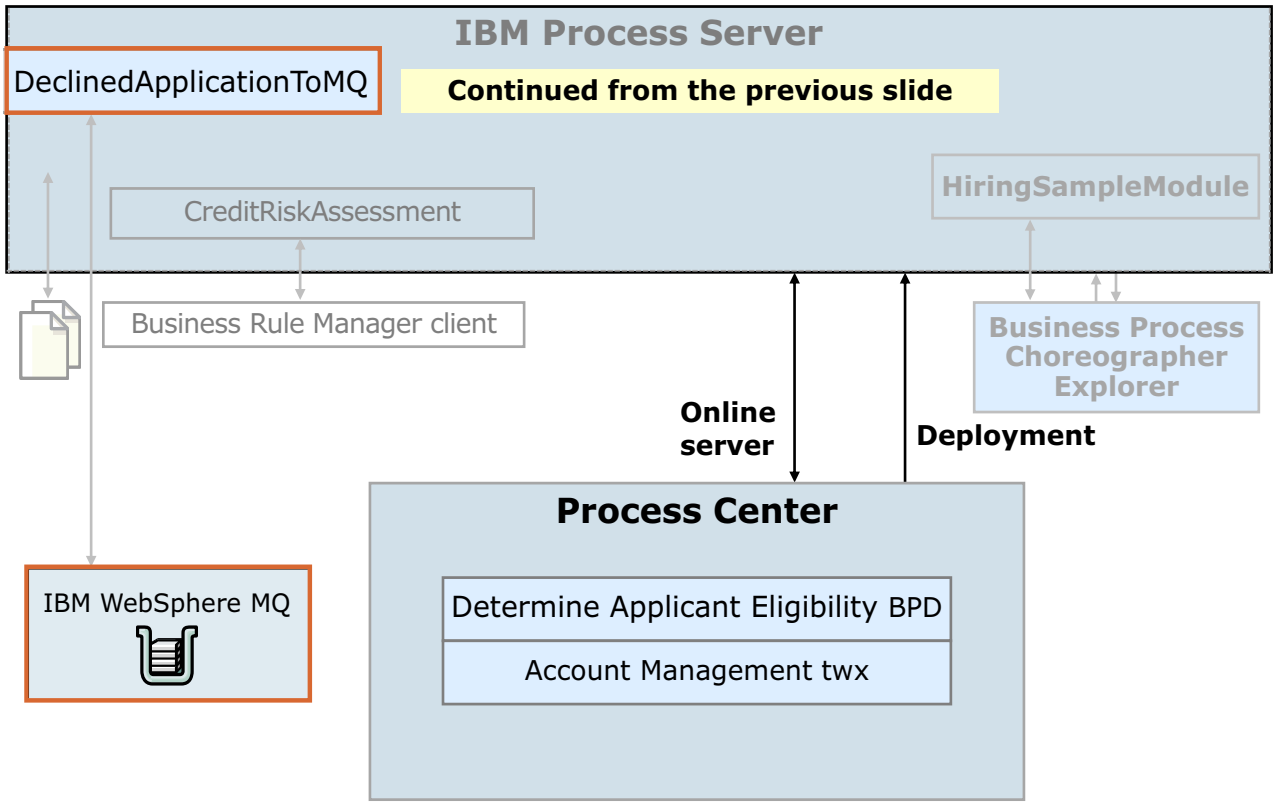
Figure 4-53. Course business scenario: What you work with in Ex 3 (1 of 2)

In this lab, you work with IBM Integration Designer.

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

IBM Training

Course business scenario: What you work with in Ex 3 (2 of 2)



© Copyright IBM Corporation 2017

Figure 4-54. Course business scenario: What you work with in Ex 3 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM WebSphere MQ. You do not work with Process Center.

© Copyright IBM Corp. 2017

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

Unit 5. Fault handlers

Estimated time

00:45

Overview

This unit explains the fault handler element in Web Services Business Process Execution Language (WS-BPEL), and handling of the Advanced Integration services faults.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain the purpose and business value of fault handlers
- Define activity, local, and process-level fault handlers
- Explain the error handling in IBM Process Designer
- Apply error handlers to Advanced Integration services (AIS)

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-1. Unit objectives

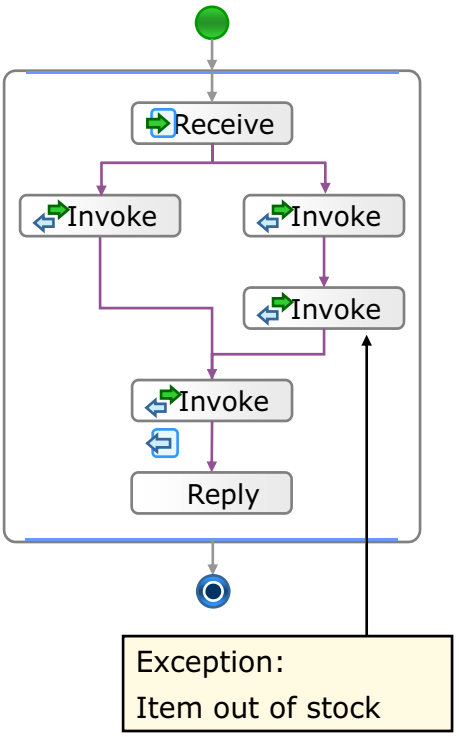
© Copyright IBM Corp. 2017

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

IBM Training

Exception handling in business processes

- Exceptions are planned and expected to happen occasionally:
  - Destination for shipment causes the customer to pay extra fees
  - Losing power in the warehouse is not expected to ever happen and cannot be handled
- Exceptions might force different steps in the business process:
  - Ordered item is no longer stocked; offer the customer a different item
- Exceptions can arise from service invocations:
  - Call a service that responds with a fault message



Fault handlers

© Copyright IBM Corporation 2017

Figure 5-2. Exception handling in business processes

If your BPEL invokes an external operation in a *synchronous* manner (request/response), and the operation fails, the Failed Event Manager tool does not handle it. The exception returns to the BPEL engine, and you must handle it within the business process logic itself. Like any application, it is important to have fault handling in your business process. Almost always, some business exception occurs during the lifetime of your application.

While exceptions in the execution of a business process are unexpected and not planned, many business processes have the capabilities to react to and handle exceptions. For example, consider an order business process where the total weight of a package might be too great for normal shipping costs. While this situation is not expected to happen during normal execution, it can be handled by charging the customer extra fees and sending the package to the customer. The order would not need to be stopped or canceled for any reason.

Exceptions in a business process might occur on individual steps where they can be handled, or for those situations where something is not completing correctly, the business process can end with an exception. In either case, the exception is returned to the caller, where it can be handled.

When modeling business processes, implementations of runtime exceptions are expected to be handled within the business process. While this implementation is possible, it creates a tight connection between the business process model and the implementation. Therefore, if the implementation changes, it might require updates to the business process. In most cases, you want

© Copyright IBM Corp. 2017

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

to keep the two separate to avoid these types of situations. For example, you might handle the breakdown of a cardboard packaging machine in your order process. However, if you change to a plastic shrink-wrap machine, you must change the business process to handle the breakdown of the shrink-wrap machine. Also, the breaking down of either machine does not qualify as a “planned” exception. If you must model these types of situations into your business processes, you might want to consider reevaluating the implementation and look for ways to make it more stable.

© Copyright IBM Corp. 2017

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

IBM Training

Fault types

- The business process engine recognizes two categories of faults:
  - *Built-in faults* that the engine throws to signal low-level failures in the system
  - *User-defined faults* that the business partner or the current process itself throws
- All business process faults have the following characteristics:
  - They must be declared with a unique qualified name as the fault name
  - They can optionally provide more fault information through a fault variable
- User-defined faults that the partners to a business process throw are shown as the business objects in the interface definition
  - WSDL fault message definitions describe the fault variable type

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-3. Fault types

The engine throws built-in faults, such as **join Failure Exception**, to signal low-level failures in the system. A business partner or the current business process itself throws the user-defined faults to trigger forward-recovering activities. An example of a user-defined fault might be a **ProductOutOfStock** fault. If your business is to sell a product, it is possible that you might receive an overwhelming number of orders during the holiday season. As a result, the product goes out of stock and must be ordered through the manufacturer. As a service provider, you specify your own reacting behavior in such cases (that is, you send an email to customers to notify them about the delay).

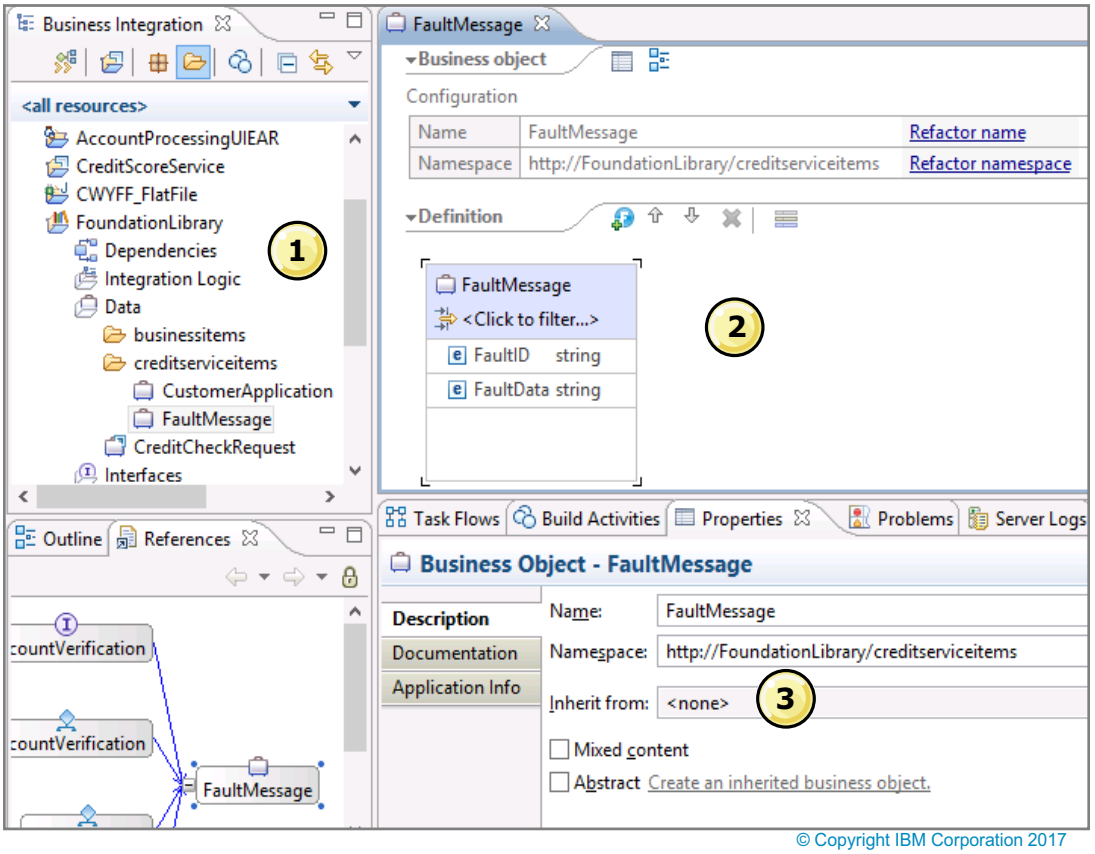
All process faults must have a unique qualified name (namespace plus the fault name). More fault information can be provided through a fault variable; however, it is optional. A WSDL might define an operation with an input message, output message, and *fault message*.

© Copyright IBM Corp. 2017

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

IBM Training

Examine user-defined fault types with the business object editor



© Copyright IBM Corporation 2017

Figure 5-4. Examine user-defined fault types with the business object editor

Creating a fault message business object is same as creating a regular business object to represent a business entity. This slide demonstrates the steps to define a fault message business object to hold detailed information about the exception.

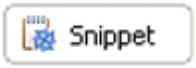
1. The ***Data*** category in the **Business Integration** view provides a list of all types that are defined in the interfaces that are saved within the business module.
2. View and modify fault messages that are defined in the interface by using the *business object* editor. In this example, **FaultMessage** can also hold a FaultData as its fault variable.
3. The fault name is a combination of the namespace declaration and the name of the business object.

© Copyright IBM Corp. 2017

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

IBM Training

Sources of user-defined faults



- A throw activity in a business process can raise a built-in or user-defined fault:
  - The *rethrow* activity takes the current fault in a fault handler catch element and raises it to a higher scope
- A Java throw statement within a Java snippet can also raise a fault:
  - Java snippets can be found in the *snippet* activity, transition conditions, link conditions, and expressions in looping or decision constructs
- An operation from a partner can return a fault reply message:
  - *Reply* activities can return a fault reply message to the business process caller
  - The business process engine raises a fault when a process receives a fault reply

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-5. Sources of user-defined faults

A fault can be thrown by using a **throw** activity. A fault that was caught and handled in a fault handler can be a **rethrow** so that the outer scope can further the process and react to the particular exception. Within a **snippet** activity, a fault can be thrown programmatically in Java.

Finally, if a particular fault cannot be handled within a particular business process, you might *reply* with a fault message to have the service requester handle it. The service requester knows that a service can respond with a fault message instead of an output by looking at its WSDL. You see an entry for the operation that states: `<wsdl:fault message= ...>`

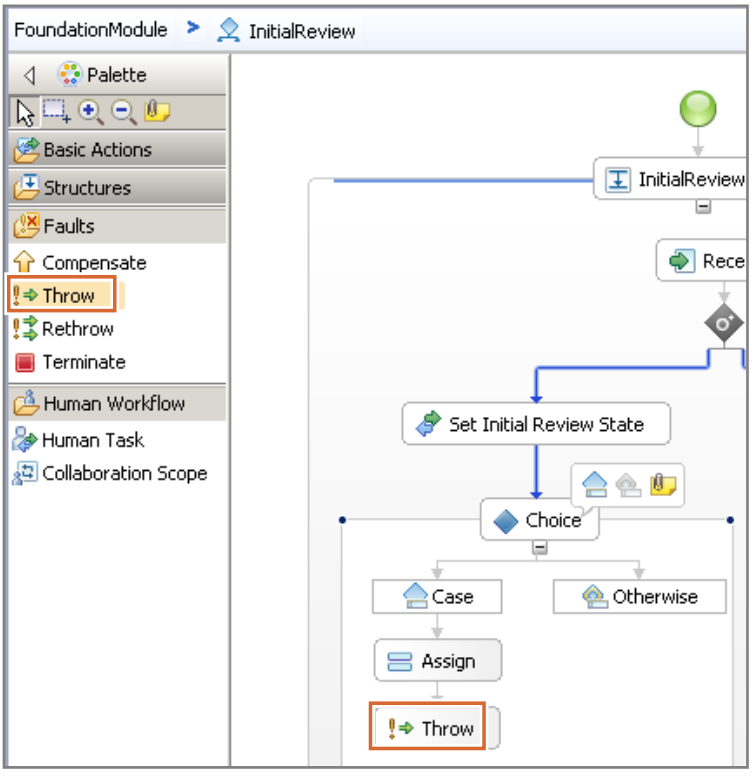
© Copyright IBM Corp. 2017

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

IBM Training

Raise a fault with the throw activity

1. In the process editor, select the *throw* activity from the palette
2. Place the *throw* activity onto the process
3. Specify the *namespace* and the *fault name* to identify the fault
   - Specify the correct namespace value
4. Optionally, pass a process variable as additional information stored in the fault variable
   - Use the *assign* activity to store fault information into the process variable



© Copyright IBM Corporation 2017

Figure 5-6. Raise a fault with the throw activity

This slide reviews how a fault can be raised from your business processes.

© Copyright IBM Corp. 2017

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

IBM Training

Raise a fault from a Java snippet

- Use the `raiseFault(...)` method to raise a business process fault from a Java snippet:
  - `raiseFault(QName faultName);`
  - `raiseFault(QName faultName, String variableName);`
  - `raiseFault(QName faultName, Object message);`
- The following Java code fragment raises a fault with a fully qualified fault name and fault variable:

```
"BOFactory factory = (BOFactory) ServiceManager.INSTANCE.locateService(
"com/ibm/websphere/bo/BOFactory");
    BOType boType = (BOType) ServiceManager.INSTANCE.locateService(
      "com/ibm/websphere/bo/BOType");
    Type type = boType.getType(
      "http://exceptions.widgets.training.ibm.com",
      "CustomerNotFoundExceptionType");
    DataObject faultMessage = factory.createByType(type);
    faultMessage.set("message", customer.get("customerId"));
    QName faultName = new QName(
      "http://exceptions.widget.training.ibm.com",
      "CustomerNotFoundException");
    raiseFault(faultName, faultMessage);"
```

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-7. Raise a fault from a Java snippet

Do not throw a `ServiceBusinessException` from a Java snippet. It might result in a `wpc:runtimeFailure`. You must use the `raiseFault()` method instead. A fault can be thrown from a snippet activity programmatically by using the `raiseFault()` method. In the Java code fragment, the **BOType** class returns the data type of the exception type. The **BOFactory** object returns an instance of the fault message as an SDO DataObject, which is based on the specified exception type. The **DataObject** `set(...)` method stores the fault message information. You must create a qualified name for the fault that is based on the name of the exception and the exception namespace. Finally, use the `raiseFault` method to raise a fault and its message in the business process.

- Step through the example code:

  ```
  BOFactory factory = (BOFactory)
  ServiceManager.INSTANCE.locateService("com/ibm/websphere/bo/BOFactory");
  ```

- Locate the BOFactory service:

  ```
  BOType boType = (BOType)
  ServiceManager.INSTANCE.locateService("com/ibm/websphere/bo/BOType");
  ```


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

- Locate the BOType service:

```
Type type = boType.getType("http://exceptions.widgets.training.ibm.com",
"CustomerNotFoundExceptionType");
```

- Define the type of fault business object instance:

```
DataObject faultMessage = factory.createByType(type);
```

- Instantiate a fault message business object:

```
CustomerNotFoundException business object.faultMessage.set("message",
customer.get("customerId"));
```

- Set the value for the message attribute for the **`faultMessage`** business object, which is an instance of the **`CustomerNotFoundException`** business object:

```
QName faultName = new QName("http://exceptions.widget.training.ibm.com",
"CustomerNotFoundException");
```

  - QName is a qualified name for your fault business object: namespace plus the fault name.

- Finally, throw a fault by using the **`raiseFault()`** method:

```
raiseFault(faultName, faultMessage);
```

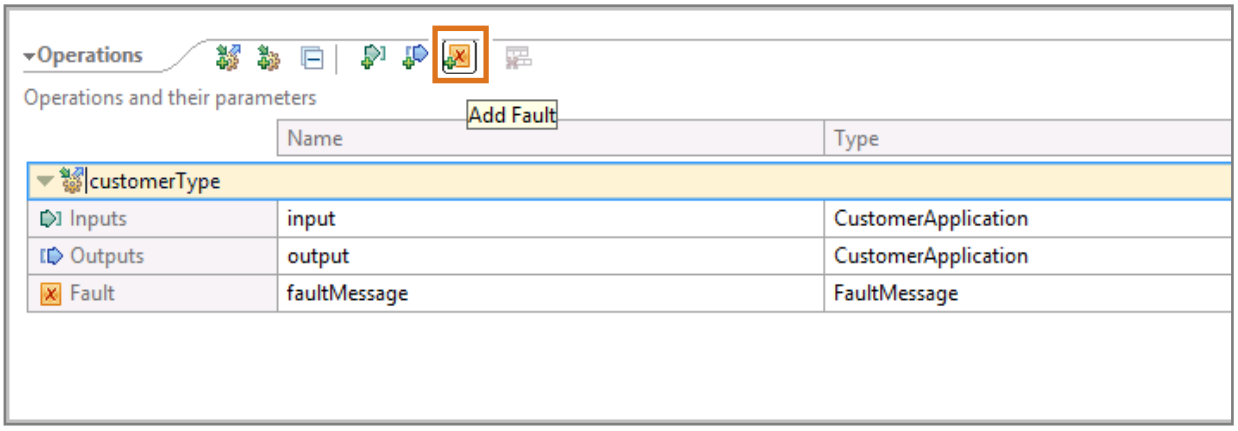
© Copyright IBM Corp. 2017

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

IBM Training

Return a fault by using the reply activity (1 of 2)

1. In the business process interface, add a fault reply to the operation that the business process represents:



2. Add a reply activity at the end of the business process

© Copyright IBM Corporation 2017

Figure 5-8. Return a fault by using the reply activity (1 of 2)

Use reply with fault when the fault is intended for the caller to handle or react. Use a throw activity when the fault must be handled within the business process. As a service provider, you might decide not to handle a particular case of fault. Instead, it can be more appropriate for your service requester to manage that. In this case, you can use the *reply* activity in BPEL. Normally, a service replies with an output message. However, if its interface is defined to *reply* with a fault message, it is possible to do so.

Steps 1 and 2 show how to modify the interface so that your WSDL can define the type of fault message.

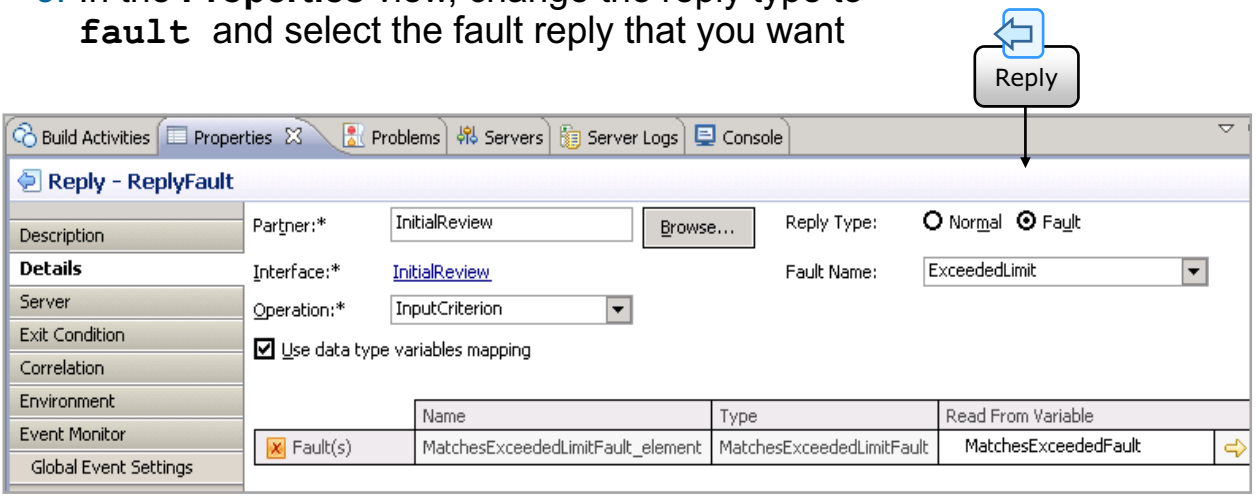
© Copyright IBM Corp. 2017

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

IBM Training

Return a fault by using the reply activity (2 of 2)

3. In the **Properties** view, change the reply type to **`fault`** and select the fault reply that you want



4. Map the fault reply variable to the process variable with more fault information, if required

© Copyright IBM Corporation 2017

Figure 5-9. Return a fault by using the reply activity (2 of 2)

Steps 3 and 4 show how to set the property of the *reply* activity when a fault message must be returned to a service requester.

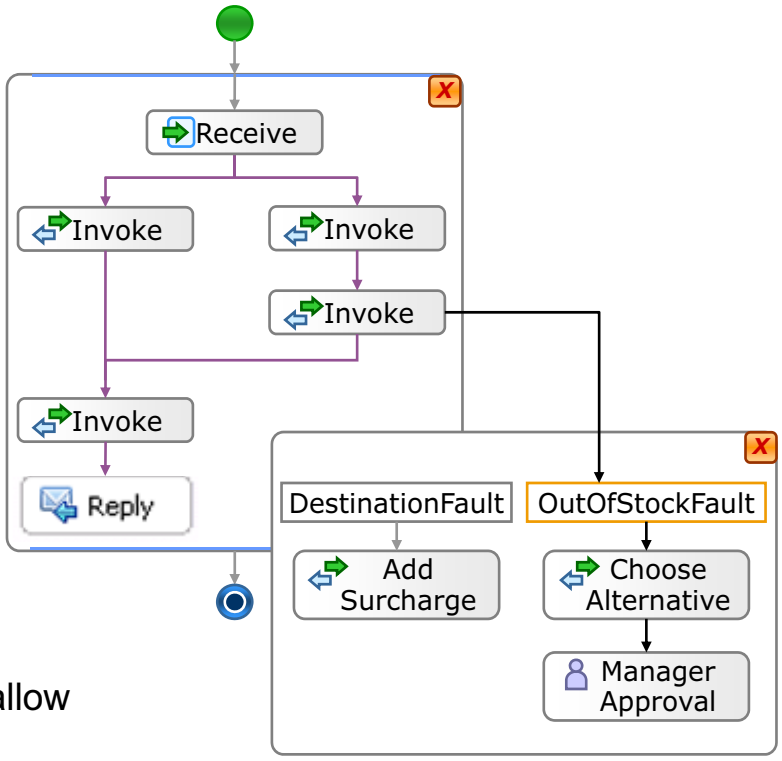
© Copyright IBM Corp. 2017

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

IBM Training

Manage faults with fault handlers

- Fault handlers manage exception conditions at a process level
- Handlers do not embed fault handler logic within service implementation
- Activities handle programming failures; they raise faults in business processes
- Fault handlers provide a *forward recovery* mechanism
- Handlers that complete without throwing a fault allow the business process to continue as if no error occurred



Fault handlers

© Copyright IBM Corporation 2017

Figure 5-10. Manage faults with fault handlers

Fault handling is built into the BPEL specification. Fault handlers run a series of activities that are designed to recover from the fault. A fault handler associates specific fault activities on either an invoke activity or a scope activity that runs when an invoke activity or an activity inside the scope activity throws the fault. In this case, the operation within the parent activity is not committed; therefore, it is possible to reverse the transaction easily. You can place activities on paths within the handler to either deal with anticipated faults, or attempt to find another way to complete the operation. A catch or a catch all element precedes each path within the fault handler.

© Copyright IBM Corp. 2017

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

IBM Training

Fault handling behavior

- When an activity throws a fault, the business process engine determines whether a fault handler is defined for the activity
- If a fault handler is defined:
  1. The engine tries to match the fault to a ***catch*** element
  2. If the fault handler has no suitable catch entries, the engine chooses the ***catch all*** element
  3. If the fault handler has no suitable catch entries, and no catch all element is defined, the engine runs the default fault handler
     - This fault handler triggers the compensation handlers of all enclosed activities that are completed successfully
  4. It then rethrows the fault to a higher scope
- If a fault handler is not defined, the engine runs the default fault handler
- The highest scope in a business process is the process scope
- If the fault is thrown beyond the process scope, the entire process ends in a failed state

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-11. Fault handling behavior

If a fault matches more than one fault handler, then the following rules determine which catch element processes the fault:

- If the fault has no associated fault data, a catch activity with the matching fault name is selected. Otherwise, the default "**catch all**" element is chosen.
- If the data is associated with the fault, then a catch element with matching fault name and variable values is selected. If the fault name is not specified, then a fault with a matching fault type is selected. Otherwise, the default "**catch all**" element is chosen.

An important point is that fault handling in business directly affects the quality of services. The explanation here is business-oriented. For example, in case a product is out of stock, your fault handling behavior might be to notify the customer about a delay of product shipment. For invalid credit card information, you might have to ask the customer to reenter the credit card number. In business, such types of faults must be handled. Catching a specific fault gives results in a better automated process even when exceptions occur.

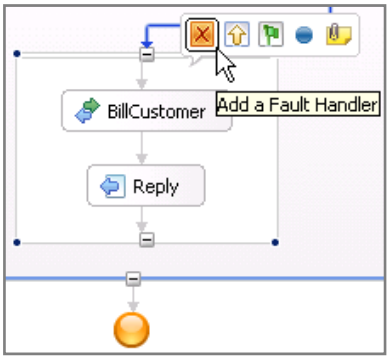
© Copyright IBM Corp. 2017

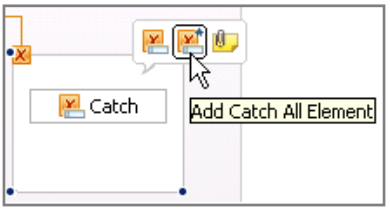
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

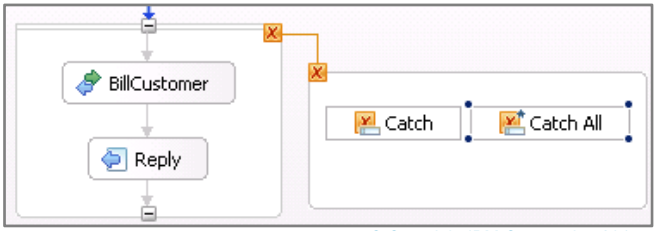
IBM Training

Add a fault handler to a business process

- In the process editor, select a scope or an activity to hold the new fault handler
- Click the **Add Fault Handler** icon from the action bar
  - An empty fault handler is shown with a sample catch element
- Select the fault handler and use the action bar to add the catch element:
  - Catch handles a particular type of fault
  - Catch all handles all faults; this construct is analogous to the otherwise case in a choice activity
- Add activities to the fault catch element







Fault handlers

© Copyright IBM Corporation 2017

Figure 5-12. Add a fault handler to a business process

When a business process throws a fault, the business process engine tries to match the fault to one of the catch blocks first. If no matching entries exist, the engine passes the fault to the catch all block. If the catch all block is not defined in the fault handler, the engine propagates the fault to a higher scope.

The fault handlers that have a variable to store the data must catch the faults with associated data. Even if a fault handler exists that matches the fault type or fault name, the handler does not catch the fault unless it has a local variable defined.

© Copyright IBM Corp. 2017

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

IBM Training

Catch a specific fault message

- The business process engine matches a fault to a catch block through one of three methods:
  - Match by fault name with no fault variable defined
  - Match by both fault name and fault variable type
  - Match by fault variable type for a fault with no fault name specified
- To catch a specific fault:
  1. Select a **catch** element in the fault handler
  2. Choose either a built-in or user-defined fault type to catch
  3. For a user-defined fault, choose the namespace and fault name
  4. Add a variable name to access the fault variable data in the catch element
  5. Map the fault variable to a specific data type or fault definition in an interface

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-13. Catch a specific fault message

This slide shows how to specify a particular fault to be caught in the IBM Integration Designer tool.

The variable name and type declare a new variable local to the fault handler. This variable holds the data that is associated with the fault.

Later, in an exercise that follows this unit, you create a user-defined fault, Fault Message. You also catch a specific fault by giving a fault name (FaultMessage), variable name (faultMessage), and data type (FaultMessage).

© Copyright IBM Corp. 2017

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

IBM Training

Continue processing upon unhandled faults feature

Continue processing upon unhandled faults: ◉ Same as Process ○ Yes ○ No

Determines how the process can proceed when fault is not caught on the enclosing scope, such as in an *invoke* or a *snippet*

- **Same as process**:
  - Matches the setting to that of the process
  - If default (process-level) setting changes, this behavior also changes
- **Yes**:
  - The business process engine creates a fault to represent the *invoke* activity failure
  - The engine also propagates the fault to a higher scope, since no fault handler is defined for the activity
- **No**:
  - The business process engine places the activity in a **stopped** state
  - The engine creates an administrative task; this administrator can repair the activity by trying again or completing the action

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-14. Continue processing upon unhandled faults feature

This slide explains the purpose of the **Continue processing upon unhandled faults** option for invoke activities.

The **Continue processing upon unhandled faults** setting is one of the IBM extensions to the WS-BPEL specification. This setting specifies how the execution of a process proceeds when a fault is not caught on the enclosing scope. If the check box is clear, then the activity is put into the stopped state so that it can be repaired. If it is enabled, then the fault is sent to the fault handler of the enclosing scope of the activity. Finally, it is now rethrown to the next enclosing scope.

Although the activity is placed in a **stopped** state, the process itself remains in a **running** state. The **Continue processing upon unhandled faults** setting allows a long-running process to continue even if one of the subprocesses or partners failed (meaning that the partner replied with a fault message). Instead of terminating the entire process, the administrator can manually inspect and try the invoke operation again. By default, the **continue processing upon unhandled faults** setting is set to true. As mentioned before, do not try to handle runtime exceptions within your business process because this approach disrupts the separation between your design and implementation. If a runtime exception occurs, setting **continue processing upon unhandled faults** to **True** allows you to manually inspect the problem and resolve it.

© Copyright IBM Corp. 2017

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

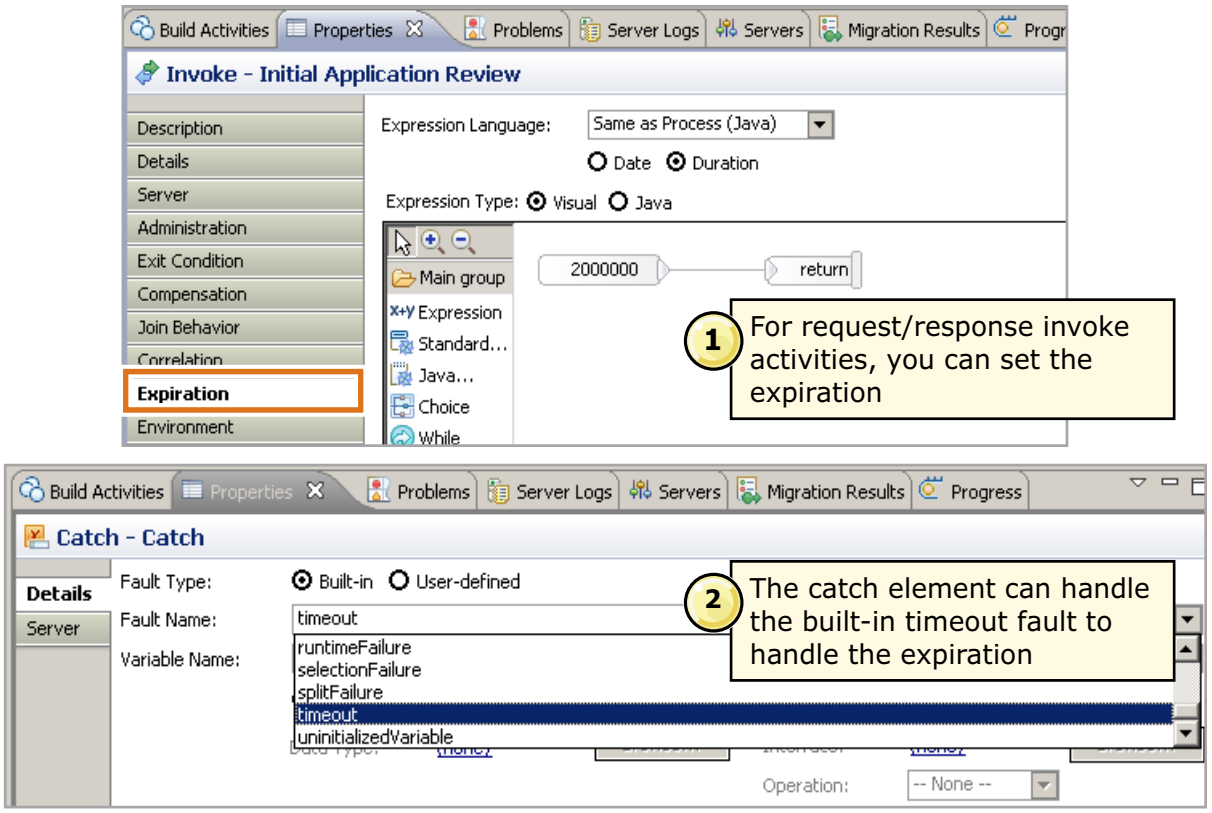
When the **Enable persistence and queries of business-relevant data** property is enabled, the details and history of the execution of this activity are stored in the runtime environment for later reference. When it is disabled, no data is kept. By default, this setting is enabled.

© Copyright IBM Corp. 2017

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

IBM Training

Expirations for long-running service invocations



Fault handlers

© Copyright IBM Corporation 2017

Figure 5-15. Expirations for long-running service invocations

For request/response type communications, the process instance goes to sleep until the response comes back. If some failure causes the response to not be returned, the particular process instance ends up hanging. To avoid such cases, you have the option of setting the expiration. Within the fault handler, you can specify the logic to resubmit the request, invoke an alternative service, or you might want to handle it.

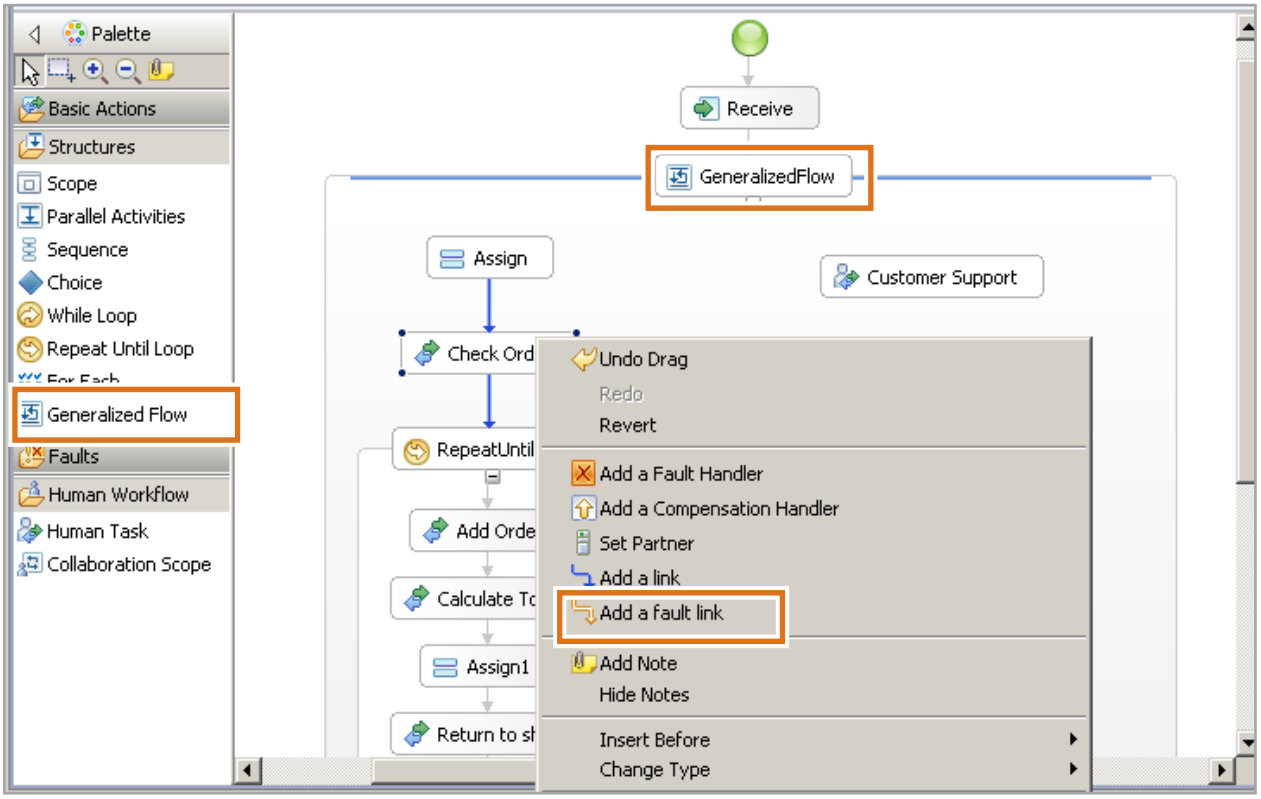
1. If you expect a service to respond within a time period, you can set the **Expiration** on the invoke activity. For example, the slide sets the **Type Validation Invoke** activity to expire if the response does not come back after 5 seconds.
2. The fault handler of the enclosing scope must catch this timeout by using the built-in timeout fault. This approach prevents a particular process instance from hanging when a failure occurs.

© Copyright IBM Corp. 2017

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

IBM Training

Generalized flow to model faults graphically (1 of 3)



Fault handlers

© Copyright IBM Corporation 2017

Figure 5-16. Generalized flow to model faults graphically (1 of 3)

When faults occur in your business process, fault handlers are typically engaged to deal with the fault. The **generalized flow** activity offers a simplified procedure to handle fault handle. From any scope or basic activity (excluding the **Throw** and **Rethrow** activities), you can add one or more “fault links”. If a specified fault occurs while an activity is running, a fault link in the activity is followed. You can define “catch” fault links for various conditions, or you can create a “catch all” fault link, which is followed when any fault occurs that the catch fault link does not cover. If multiple fault links are modeled for the same activity, best match decides which fault link is followed. The rules to catch faults are the same as for fault handlers.

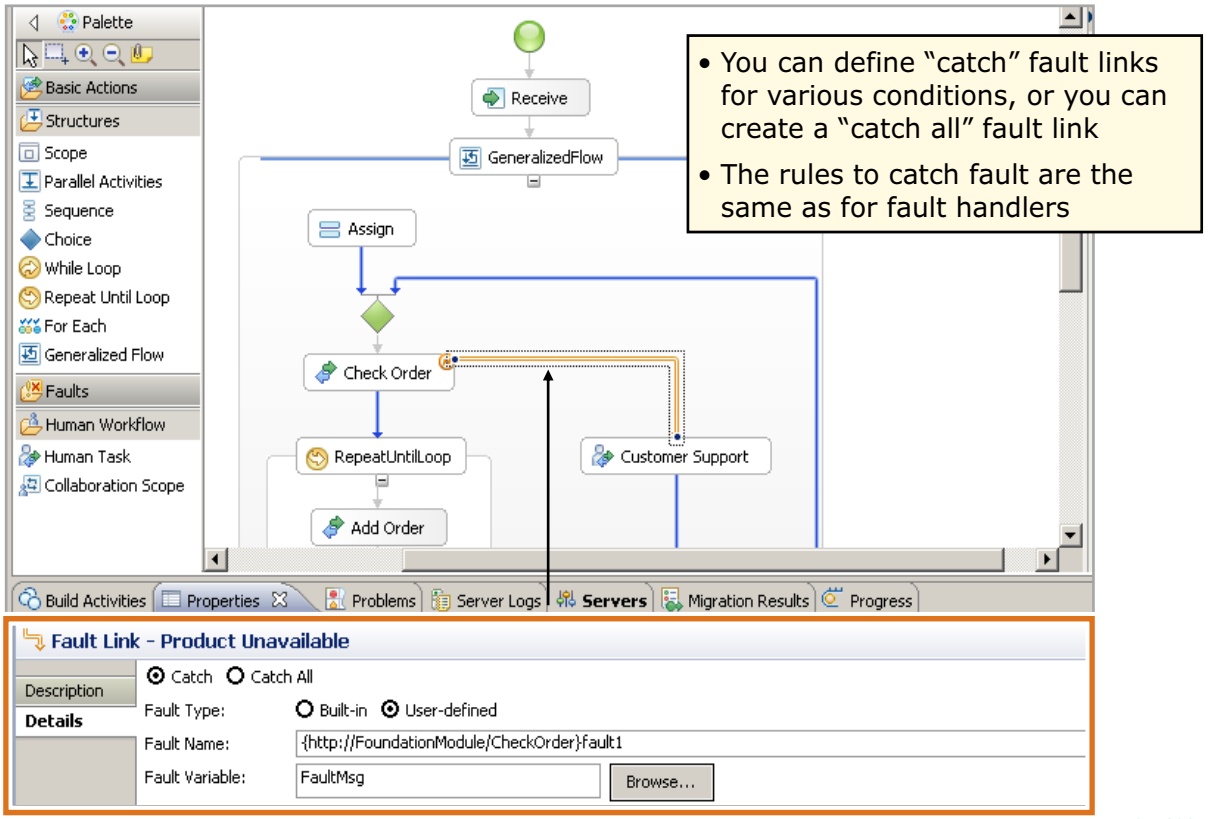
To create a fault link, right-click the activity and select **Add a fault link**. Drag the cursor over the canvas. A black line is now attached to the cursor, and when you hover over a valid incoming link icon or valid gateway, the crossed-out circle disappears, and the link snaps into place. Click the activity or gateway that you want to link to. The fault link becomes a striped red line and connects the two activities.

© Copyright IBM Corp. 2017

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

IBM Training

Generalized flow to model faults graphically (2 of 3)



© Copyright IBM Corporation 2017

Figure 5-17. Generalized flow to model faults graphically (2 of 3)

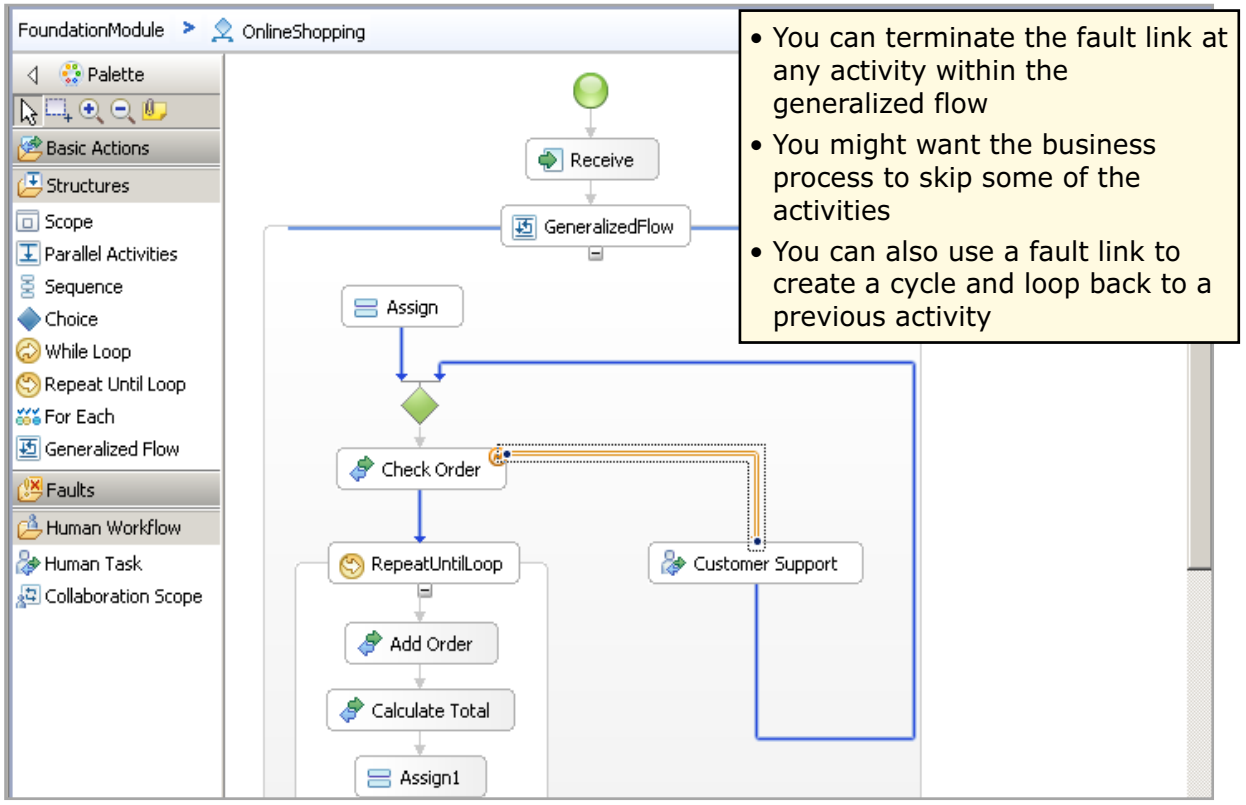
You can define "catch" fault links for various conditions, or you can create a "catch all" fault link. No more than one fault link can catch an occurred fault.

© Copyright IBM Corp. 2017

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

IBM Training

Generalized flow to model faults graphically (3 of 3)



Fault handlers © Copyright IBM Corporation 2017

Figure 5-18. Generalized flow to model faults graphically (3 of 3)

You can terminate the fault link at any activity within the generalized flow. For example, you might choose one of the following options:

- To direct the fault link to a terminate activity to have the business process, stop if a fault occurs.
- You might want the business process to skip the rest of the steps in the generalized flow and exit to the next activity. In this case, you would terminate the fault link at the last activity in the generalized flow.
- You can also use a fault link to create a cycle and loop back to a previous activity.

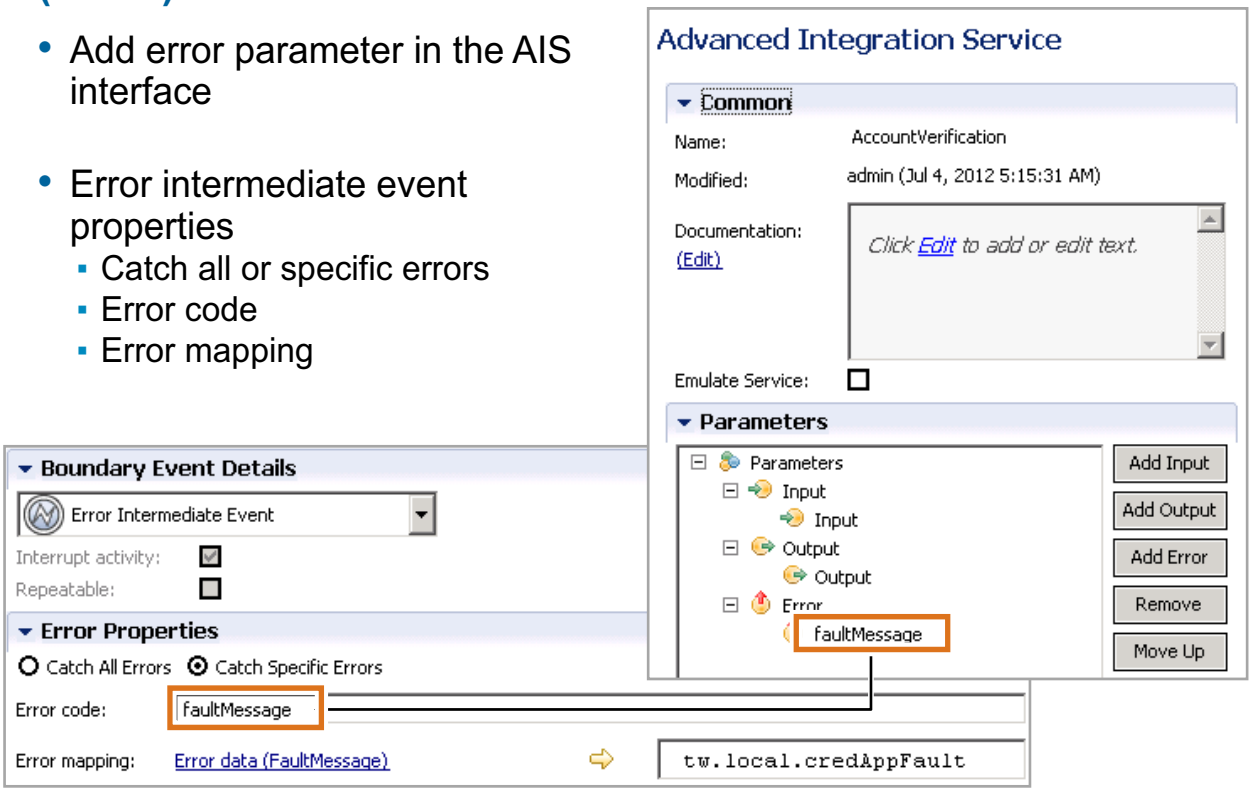
© Copyright IBM Corp. 2017

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

IBM Training

Catch Advanced Integration services (AIS) exceptions (1 of 2)

- Add error parameter in the AIS interface
- Error intermediate event properties
  - Catch all or specific errors
  - Error code
  - Error mapping



© Copyright IBM Corporation 2017

Figure 5-19. Catch Advanced Integration services (AIS) exceptions (1 of 2)

To catch errors that are thrown from the AIS, the AIS interface must have the **Error** parameter defined. An error parameter identifies an error or fault that the service that is designed in IBM Integration Designer might throw. If you want to catch a specific error by using an error event in your process model, the error name that is entered here is used for the error code in the catching error event.

An error intermediate event that is attached to any activity in the BPD or service has the following properties:

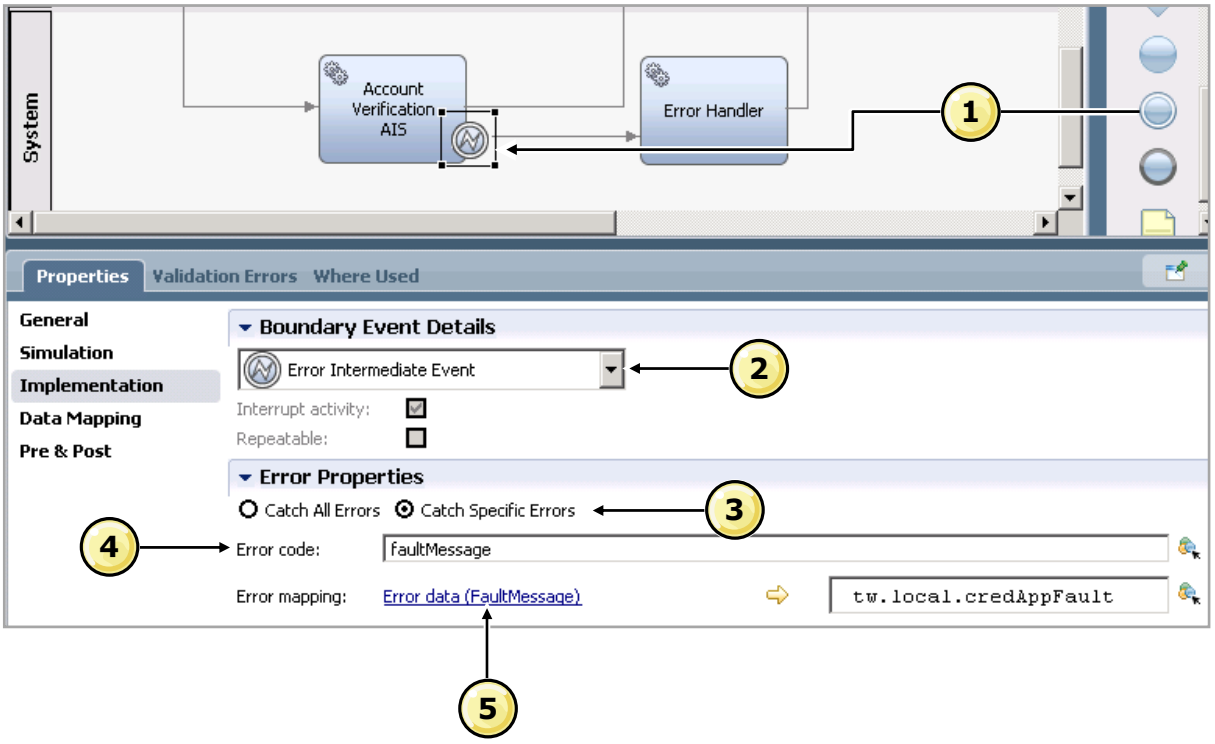
- **Catch All Errors** or **Catch Specific Errors**: You must make a selection here whether you want to filter the errors that are received or catch all errors by using this option. If you select the **Catch Specific Errors** option, then you can select the **Error code** and **Error mapping**. If you are catching specific errors, you can select the error code, map the error data, or both.
- **Error code**: To filter the errors that are caught, select an error code from a list of all thrown errors. Observe that these codes are the same name that you specified in the AIS interface in the **Error** parameter.
- **Error mapping**: To map the error data into a variable, select an error mapping variable that is defined on the Variables tab.

© Copyright IBM Corp. 2017

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

IBM Training

Catch Advanced Integration services (AIS) exceptions (2 of 2)



© Copyright IBM Corporation 2017

Figure 5-20. Catch Advanced Integration services (AIS) exceptions (2 of 2)

To catch AIS faults in a BPD process, you must do the following steps:

1. Attach Intermediate Event from the palette to the AIS activity.
2. Select the attached Intermediate Event and open the **Implementation** tab in **Properties.** In the **Boundary Event Details** section, select **Error Intermediate Event** from the menu.
3. Select the **Catch All Errors** or **Catch Specific Errors** option.
4. If the **Catch Specific Errors** option is selected, select the error code for which you must catch errors.
5. Error mapping is to map the error data to a variable that is defined on the BPD variables.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain the purpose and business value of fault handlers
- Define activity, local, and process-level fault handlers
- Explain the error handling in IBM Process Designer
- Apply error handlers to Advanced Integration services (AIS)

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-21. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: If a fault is thrown to the highest scope, a business process instance ends in the FAILED state.
2. True or False: Fault handlers are designed to handle business faults, not system faults.
3. Name the sources to raise faults in the BPEL process.
4. True or False: Interface with faults can be used in the mirrored library

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-22. Review questions

Write your answers here:

1.

2.

3.

4.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: If a fault is thrown to the highest scope, a business process instance ends in the FAILED state.
   The answer is True.
2. True or False: Fault handlers are designed to handle business faults, not system faults.
   The answer is True.
3. Name the sources to raise faults in the BPEL process.
   The answer is throw, rethrow, snippet, and reply.
4. True or False: Interface with faults can be used in the mirrored library.
   The answer is True.

Fault handlers

© Copyright IBM Corporation 2017

Figure 5-23. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Applying fault handlers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 5-24. Exercise 4

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

IBM Training

Exercise objectives

- Implement a fault handler in a BPEL process
- Test fault handlers in the IBM Integration Designer integrated test environment

© Copyright IBM Corporation 2016

Figure 5-25. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

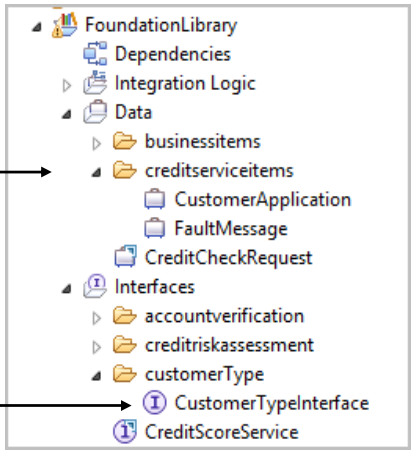
Components that are required for Exercise 4

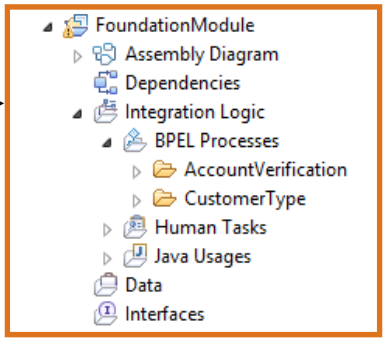
Prebuilt components that are imported in the lab:

1. **CustomerApplication** business object
2. **FoundationLibrary** library
3. **FoundationModule** module
4. **AccountVerification** process

New components that you create in the lab:

1. **CustomerApplication** business object
2. **FaultMessage** business object
3. **CustomerTypeInterface** interface
4. **CustomerType** BPEL process
5. **IsValidType** boolean variable
6. **FaultMessage** variable of FaultMessage data type
7. **CustomerTypePartner** partner reference
8. Several fault handlers





Fault handlers

© Copyright IBM Corporation 2017

Figure 5-26. Components that are required for Exercise 4

The following business objects are used in this exercise:

- **CustomerApplication**: The simple (non-hierarchical) business object that describes the characteristics of a particular customer application record. This generic business object (GBO) is instantiated from the user input through the human task.

The following interfaces are used along with the **CustomerApplication** business object:

- **CustomerApplicationDelivery**: Contains a single one-way operation that passes a single part, named **inputCustomerApplication** of type **CustomerApplication**, on the request. This interface is used on the definition of the business process and exposed as part of the exported SCA binding, which the other SCA components call.
- **CustomerApplicationRequest**: Contains a single request/response operation that passes a single part, named **inputCustomerApplication** of type **CustomerApplication**, on the request; and a single part, named **outputCustomerApplication** of type **CustomerApplication**, on the response. With the use of this interface, it is expected that any changes that are made to **CustomerApplication** on the receiving end are passed back to the process. This interface is used to define reference partners in the business process. Using the same interface simplifies the data exchange between the processes.

© Copyright IBM Corp. 2017

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

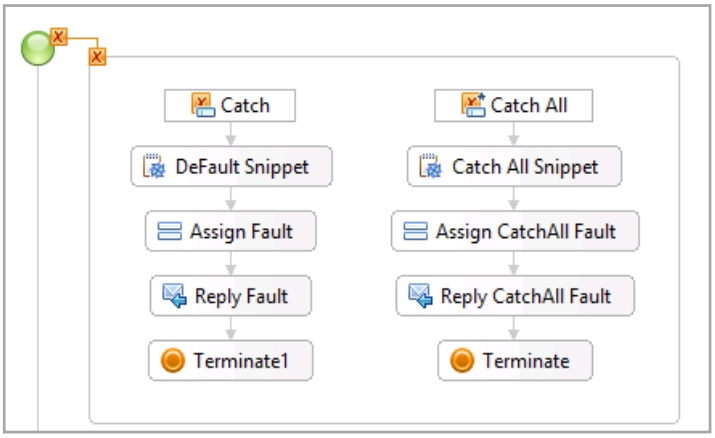
- You also build a new **FaultMessage** data type to encapsulate two string attributes, **FaultID** and **FaultData**, which are used to provide more information to the **AccountVerification** business process that catches the returned faults.

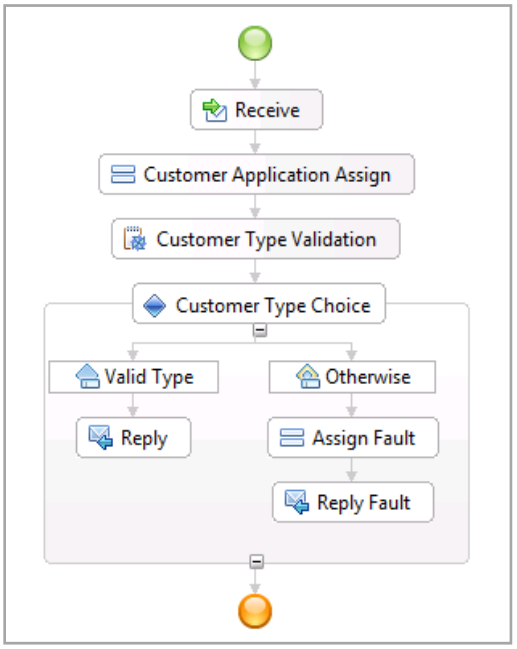
© Copyright IBM Corp. 2017

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

IBM Training

Create fault handlers to catch and throw faults in Exercise 4





Fault handlers

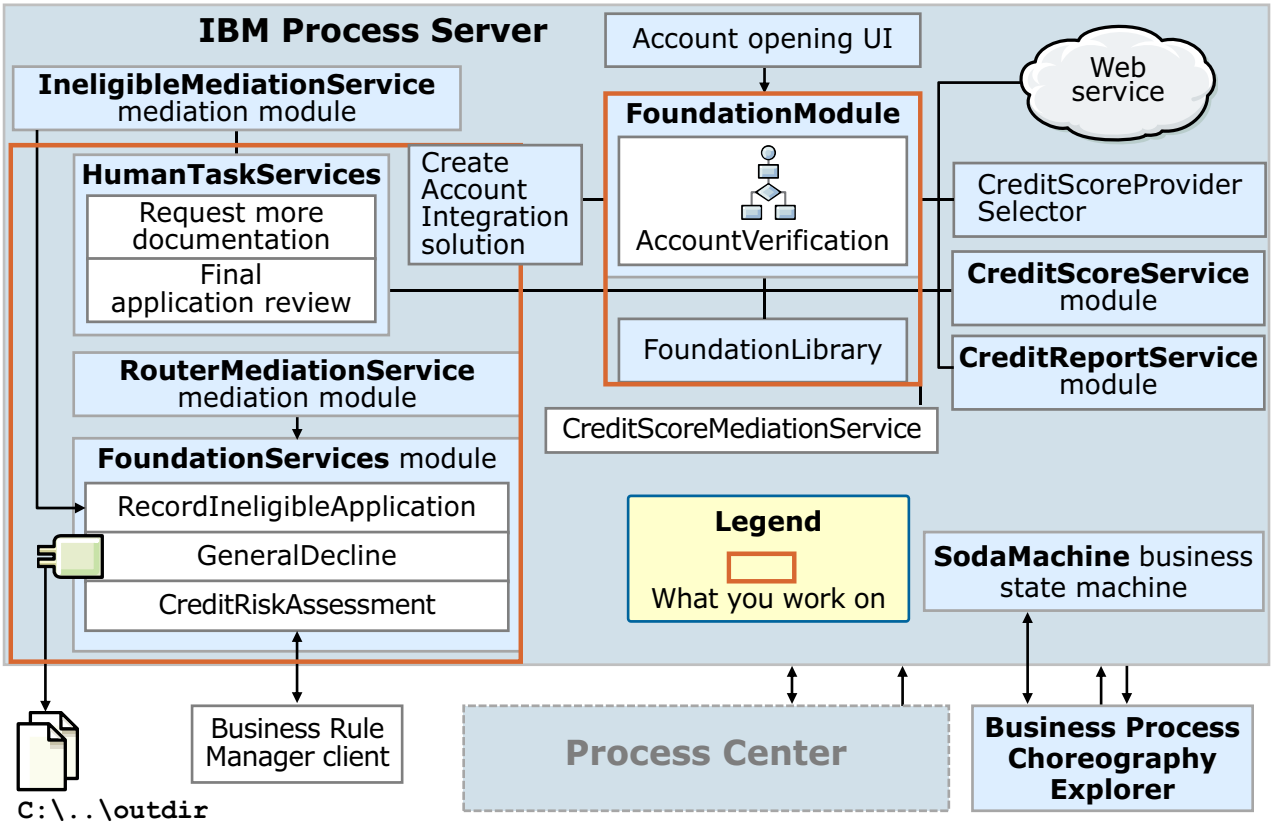
© Copyright IBM Corporation 2017

Figure 5-27. Create fault handlers to catch and throw faults in Exercise 4

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

IBM Training

Course business scenario: What you work with in Ex 4



Fault handlers

© Copyright IBM Corporation 2017

Figure 5-28. Course business scenario: What you work with in Ex 4

In this lab, you work with IBM Integration Designer.

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

Unit 6. WS-BPEL compensation and event handlers

Estimated time

00:45

Overview

This unit explains the compensation handler and event handler elements in WS-BPEL.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain the purpose and business value of compensation handlers and event handlers
- Define compensation handlers and compensation activities
- Explain the purpose of event handlers
- Explain the lifecycle of event handlers

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

The need for compensation

- Fault handlers attempt to reverse partial or unsuccessful operations which the fault signals
  - If the fault handler does not throw any faults, the business process continues forward until completion
- What if certain operations within the scope are already committed?
  - For example, a factory completes an order for 100 widgets, but the client changes the specifications before delivery
- What if certain operations are within the scope that the rollback of the current transaction cannot undo because of the nature of the operation?
  - For example, after emailing a promotion to its clients, the company wants to change the date and details of the promotion

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-2. The need for compensation

Fault handling is considered to be a forward recovery of unsuccessful operations. Some exceptions are expected to happen, and the developer must implement an alternative processing path through the fault handler. Compensation is an unexpected fault. In some situations, the business process instance might have to roll back. In long-running business processes, the transaction provides durability. It means that successfully committed transactions cannot be easily rolled back. A good example is an email service. After an email is successfully sent to recipients, it is not pushed to roll back. Compensation is a way to handle such a scenario.

© Copyright IBM Corp. 2017

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

IBM Training

Characteristics of transactions: ACID test

- **A**tomicity
  - Promises that the system does either all of the updates or none of the updates
- **C**onsistency
  - Promises that the system remains in a legal state when the transaction begins or ends.
- **I**solation
  - Ensures that changes that are made within the transaction are not visible to operations outside the transaction until it is committed
- **D**urability
  - Promises that if a transaction is committed, the updates that the transaction makes are not undone, even after system failure

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-3. Characteristics of transactions – ACID test

The following are the characteristics of **transactions** (ACID):

- **A**tomicity promises that the system does either all of the updates or none of the updates.
- **C**onsistency promises that the system remains in a legal state when the transaction begins or ends.
- **I**solation ensures that changes that are made within the transaction are not visible to operations outside the transaction until it is committed.
- **D**urability promises that if a transaction is committed, the updates that the transaction makes are not undone, even after system failure.

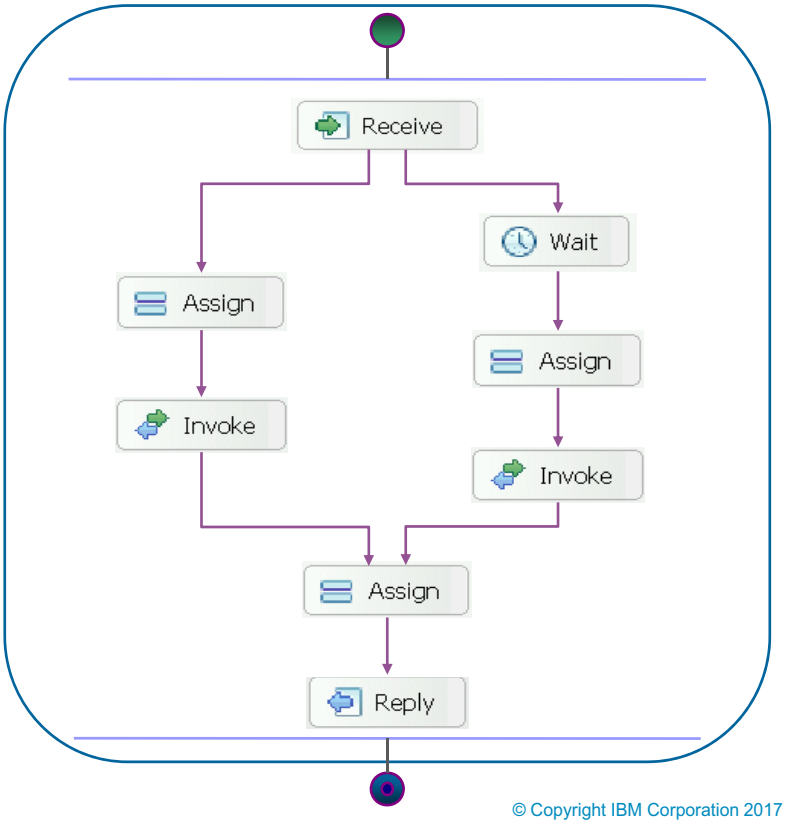
© Copyright IBM Corp. 2017

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

IBM Training

Business process execution modes: Long-running processes

- Long-running processes
  - Consist of a series of chained transactions
  - Provide true parallelism
  - Intermediate state is persisted in Business Process Choreographer database



© Copyright IBM Corporation 2017

Figure 6-4. Business process execution modes: Long-running processes

Business processes execute in two modes: long-running processes and microflows. A microflow is a short-running process. It is an IBM extension to the WS-BPEL specification, which considers all processes to be long-running. Compensation behavior is different for the two execution modes.

In a long-running process, each transaction is committed individually. To set up compensation, you must specify how to deal with each transaction, and whether it is required to be compensated if the process fails. In a microflow, the entire process executes within a single transaction. To set up compensation, you must store the original properties for each invoke activity within the microflow so that the original properties can be restored if the process cannot be committed and must be rolled back.

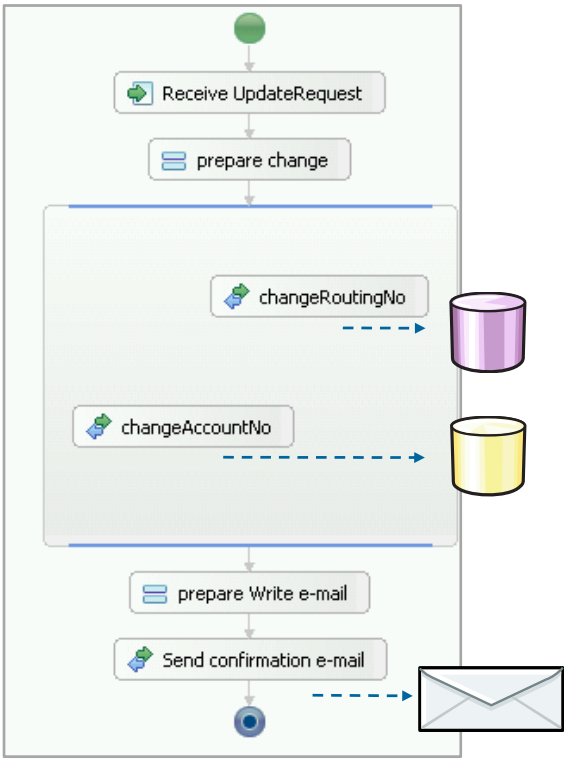
© Copyright IBM Corp. 2017

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

IBM Training

Business process execution modes: Microflows

- Microflows
  - Run entirely in one transaction and in one thread (no true parallelism)
  - Short-lived processes; no long-running activities are allowed
  - State of a microflow instance is not persisted in Business Process Choreographer database



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-5. Business process execution modes: Microflows

Microflows are known as short-running processes, and they are an IBM extension to the WS-BPEL specification. The ways to handle compensation behavior are different for long-running business processes compared to microflows because of their different natures.

© Copyright IBM Corp. 2017

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

IBM Training

Compensation and execution mode

- Compensation allows business processes to define activities that return the business to a **balanced state** when the process cannot be reversed
  - A balanced state is an outcome that is acceptable to both the client and the business
- The runtime behavior and objective of compensation differ depending on the execution mode of the process:
  - **Compensation in a long-running process** attempts to restore balance after one or more transactions are committed
  - **Compensation in a microflow** attempts to restore balance when the process involves activities whose effects are not undone if the current transaction is rolled back
  - Sending an email is an example of a service that is done independently of the outcome of the current (ACID) transaction

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-6. Compensation and execution mode

Before you can create a compensation for a process, you must identify:

- The potential condition that would put the process into an unbalanced state and where this unbalanced state might happen in the process
- The steps necessary to deal with it and return the process to an equitable state

Compensation does not necessarily mean “to undo a committed transaction.” Often, compensation is thought of as a means to undo a committed transaction. However, this definition is not entirely correct.

- Not all committed operations can be undone. For example, a customer who paid cash might receive a store credit when returning merchandise. This circumstance places both parties at a balanced state, although it is not the same as the original state.
- If the client and vendor cannot agree on a compensation method, then the business compensation is unsuccessful. Compensation in a microflow strives to maintain the integrity of the transaction, even if an operation within the transaction cannot be rolled back.
- If an email invitation was sent and the customer read it, you cannot cancel the message. In a compensation operation, you can send out another email that cancels the invitation.

© Copyright IBM Corp. 2017

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

- If a customer purchased some personalized merchandise, it cannot be returned to the store shelf if it is returned. A simple rollback operation does not work because the product cannot be returned to inventory. In this instance, a compensation operation might place the item into a “discount” inventory.

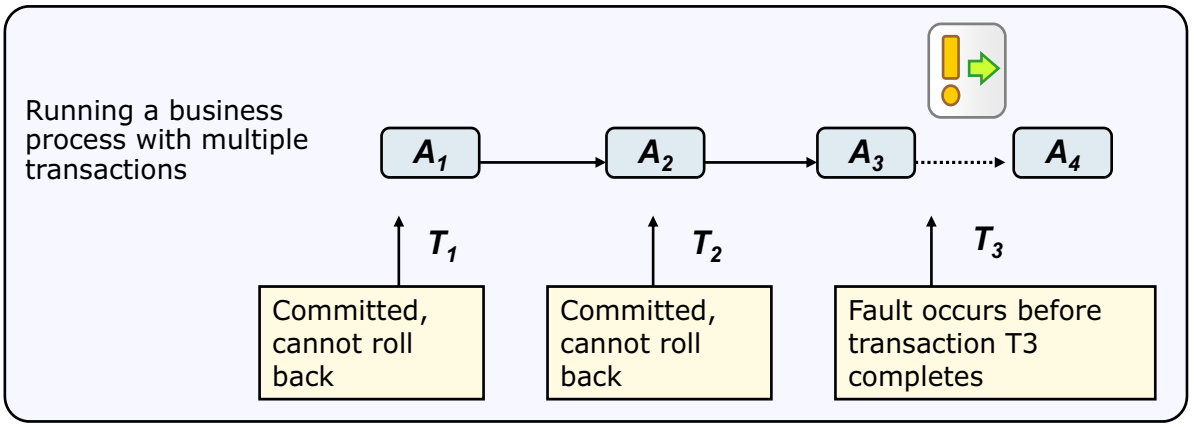
© Copyright IBM Corp. 2017

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

IBM Training

Compensation in long-running processes

- By using compensation in long-running processes, you can update committed transactions within a business process:
  - If a fault occurs in an interruptible business process, the engine cannot roll back already committed transactions
  - For example, a customer might be billed for a purchase, but the inventory item that is ordered is sold out
  - Even if no problems occur, new information might arrive **after** transactions are committed but **before** the process finishes



© Copyright IBM Corporation 2017

Figure 6-7. Compensation in long-running processes

With long-running business processes, each activity in the process is a complete step and can be considered a committed act without the possibility of undoing it. Consider a business process, for example, ordering a product. The order is received from the customer and is filled. While the order is being filled, the customer is billed. However, before the order is shipped to the customer, a cancellation notice is received from the customer, which is considered a business process-level failure. A number of completed activities cannot be easily undone now. In many cases, another business process or set of activities must be completed to undo the activity. With compensation, you can specify a service that is called or a business process end in a failure. The service does some type of undoing to compensate the completed activity.

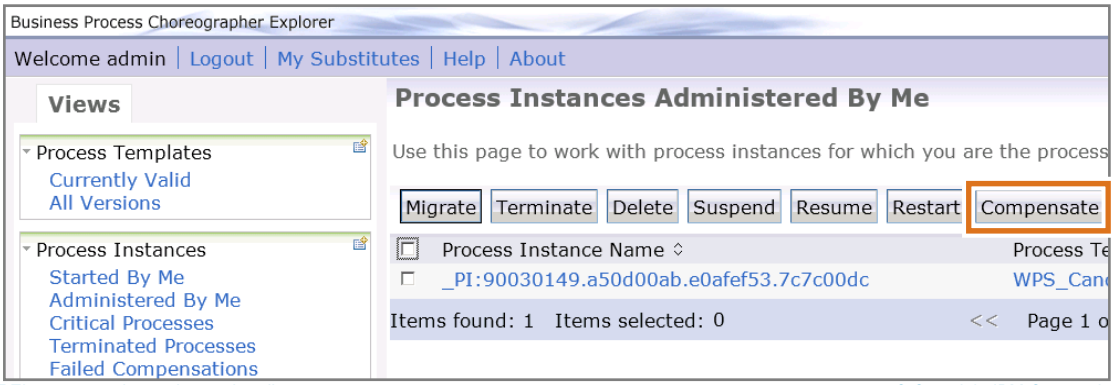
© Copyright IBM Corp. 2017

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

IBM Training

When does compensation occur in a long-running process?

- Compensation occurs during these circumstances in a long-running process:
  - The current transaction is marked for **rollback**
  - A ***compensate* activity** within the business process explicitly triggers compensation
  - The administrator triggers compensation through Business Process Choreographer Explorer
- Compensation occurs at the scope level
  - Either part of the process or the entire process can be compensated



© Copyright IBM Corporation 2017

Figure 6-8. When does compensation occur in a long-running process?

The `forceTerminate(...)` method is not the same as a terminate activity. The terminate activity always ends process execution *without* triggering fault or compensation handlers.

It is sometimes necessary for someone with process administrator authorization to terminate a top-level process instance that is known to be in an unrecoverable state. For example, when an application is invoked and fails, it does not return to a dormant state. Because a process instance terminates immediately, without waiting for any outstanding subprocesses or activities, the process instance must be terminated only in exceptional situations.

© Copyright IBM Corp. 2017

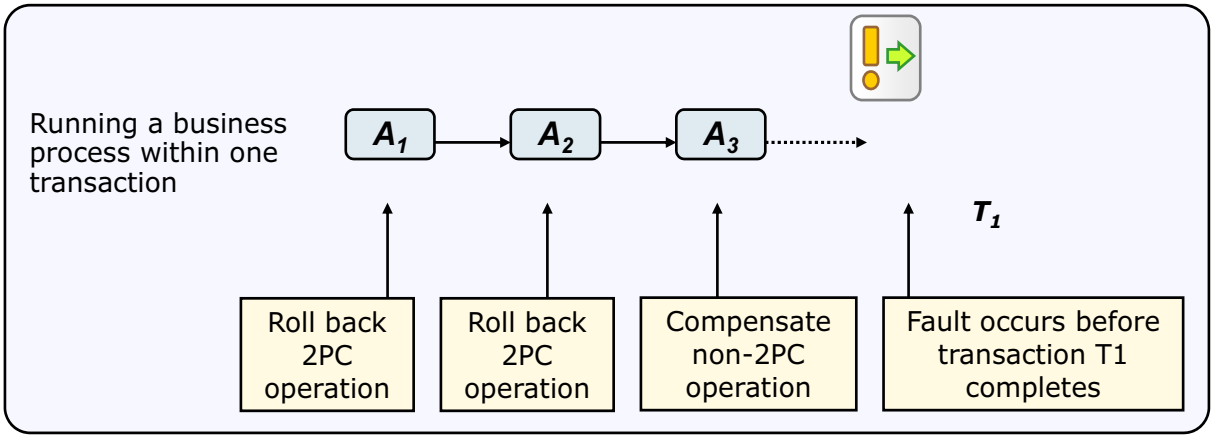
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Compensation in a microflow

Using compensation in a microflow, you can associate a compensation operation with a two-phase commit (2PC) resource

- A rollback of a 2PC resource invokes compensation operations
- The compensation operation restores work that the non-transaction activities do to a balanced state
  - For example, after sending email invitations, send another email to cancel the invitation



© Copyright IBM Corporation 2017

Figure 6-9. Compensation in a microflow

Recall that a microflow runs under one transaction. However, it can associate with an XA resource, which might trigger a rollback to occur in the microflow. Although a microflow always runs in a single transaction, you cannot control the transaction boundaries of partners. The example that is mentioned here is that you might invoke a partner service to send an email. Your process is running in a single transaction; however, the partner service might not participate in your process transaction. Thus, you still must be concerned about compensating completed activities in a microflow.

© Copyright IBM Corp. 2017

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

IBM Training

When does compensation occur in a microflow?

- Compensation occurs in a microflow during transaction rollback
  - Since a microflow runs under one transaction, the business process is always compensated as a whole
- The engine executes compensation operations for activities that:
  - Are completed successfully ***and***
  - Did not participate in the transaction of the microflow
- The Service Component Architecture (SCA) quality of service qualifiers of the *process* and the *called service* determine whether a service participates in the transaction of the microflow
- If an error occurs during compensation, manual intervention is required
  - Process administrators can manage errors through the Business Process Choreographer Explorer

WS-BPEL compensation and event handlers © Copyright IBM Corporation 2017

Figure 6-10. When does compensation occur in a microflow?

In a microflow, the entire process runs within a single transaction. To set up compensation, you store the original properties for each invoke activity within the microflow so that they can be restored if the process cannot be committed and must be rolled back. If this microflow fails, the runtime engine crawls back through the invoke activities in reverse order, and restores each activity to its previous state.

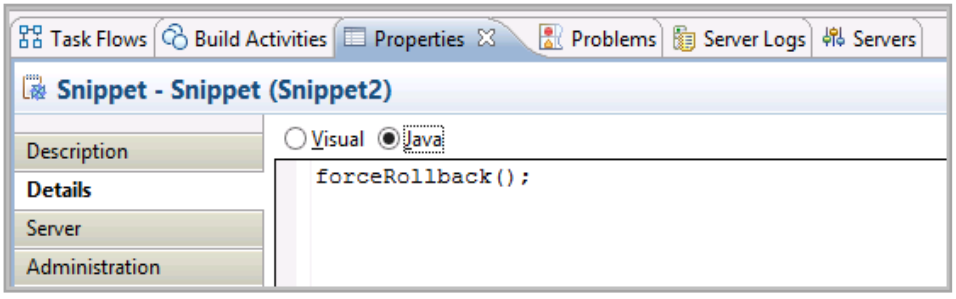
© Copyright IBM Corp. 2017

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

IBM Training

Compensation and fault handlers

- When an unhandled fault reaches the default fault handler for the process, the process engine marks the current transaction for rollback only
  - A rollback only setting ensures that the engine runs compensation
- To trigger compensation in a **microflow** with a fault handler defined, use the generic process API `forceRollback()` method
  - This method forces the transaction to roll back, and triggers compensation
  - This method is available in all Java snippets in a microflow
  - This method has no effect when called in a long-running business process



© Copyright IBM Corporation 2017

Figure 6-11. Compensation and fault handlers

This slide shows the use of `forceRollback()` in the fault handler by using snippet activity.

If you terminate the process instance with compensation, the compensation handler that is defined for the process template is called. If the process template compensation handler is not defined, the default compensation handler is called. If you terminated the process instance without compensation, the process instance is terminated immediately without waiting for activities to end normally.

The force terminate request does not affect the applications that the process starts. If these applications must be terminated, you must add statements to your process application to explicitly terminate the applications that the process starts.

© Copyright IBM Corp. 2017

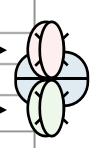
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Building blocks for compensation (1 of 2)

- Define a **compensation pair** to enable compensation for the activity

	Name	Type
BillCustomer		
Input(s)	customerId	int
	orderId	int
Output(s)	success	boolean
Fault(s)	CustomerNotFound	CustomerNotFoundException
UndoBillCustomer		
Input(s)	customerId	int
	orderId	int



Primary operation
Bill customer account for order

Compensation operation
Refund charged amount to account

- Primary operation
  - Performs normal actions of the activity
- Compensation operation
  - Allows updates after primary operation commits its changes
  - Executed as a one-way operation; output messages are ignored
  - Runs only after primary operation is completed

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-12. Building blocks for compensation (1 of 2)

Whether the compensating action is a reverse action or another business process, it is a service that is defined with an interface. The compensating service is associated with the primary activity or service in a one-to-one mapping. The two services make up what is known as a compensation pair. During normal execution, the primary service is executed and completed. When something goes wrong in the business process and if the primary service is completed, then only the corresponding compensating service can be called. The compensating service cannot be invoked without the completion of the primary service.

The compensating service must be synchronous. If a compensating service has an output message that returns information, it is ignored. The compensating service runs as a one-way operation.

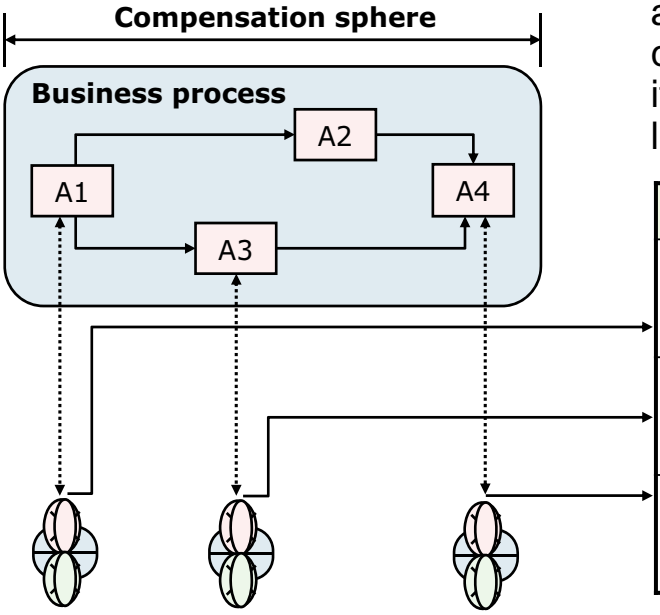
© Copyright IBM Corp. 2017

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

IBM Training

Building blocks for compensation (2 of 2)

- When a process starts, if compensation is enabled, a **compensation sphere** is created, which is a list of activities that are completed successfully thus far
- During execution, when an activity with an associated compensation service completes, its **undo** activity is added to the list in the compensation sphere



Compensation sphere		
A1	Compensating service information	Input, output, messages
A3	Compensating service information	Input, output, messages
A4	Compensating service information	Input, output, messages

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-13. Building blocks for compensation (2 of 2)

If a fault reaches the top process level, compensation is triggered.

- Registered compensating services are called in reverse order.
- The business process instance is set to state `COMPENSATING`
- Upon completion, the process instances are marked `COMPENSATED`.

When a business process is started (as a compensation scope provider), it creates a compensation sphere and activates the compensation sphere. A compensation sphere defines the scope for a set of compensable activities that the business process does. When the business process runs, each compensation pair registers with the compensation sphere. It then runs its primary services and stores the information that is needed to run the compensation service if the business process must be compensated. The information is called a “proclet” and is stored in the business process database. This information is stored such that in a long-running business process, the information can be retrieved even after the server restarts. This information is needed at the time when the compensating service is registered.

The input and output messages that are sent to and received from an activity are part of the proclet. The messages are part of the scope of the activity in regard to the overall business process and the call to the activity. The input and output messages are the only items to which the activity has access at the time of running the activity. It is from this information that the activity can be

© Copyright IBM Corp. 2017

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

compensated. No other information about the business process is needed or is available to the compensating service.

During execution, when an activity with an associated compensation service successfully completes, its **undo** activity is added to the list in the compensation sphere.

Each undo entry contains two pieces of information:

- The service that does the compensation activity
- Input values for the undo processing; for example, the reservation number

An activity that defines the compensation and returns a fault message (has a fault terminal) does not register the compensation activity with the compensation sphere. Since the activity is returning a fault message, it is assumed that the error and “undoing” were already done within the activity. Also, the output message contents would not be available as the fault message were returned instead.

If one of the undo operations fails, the business process instance is marked as IN DOUBT and needs manual resolution. The web client offers compensation repairs. Finally, if an undo service has no explicit input, the input is created implicitly by using the input and output data of the activity to which the undo activity is associated. The parts of the input and output message are merged, and if both messages contain parts with the same name, the part from the output message is taken.

© Copyright IBM Corp. 2017

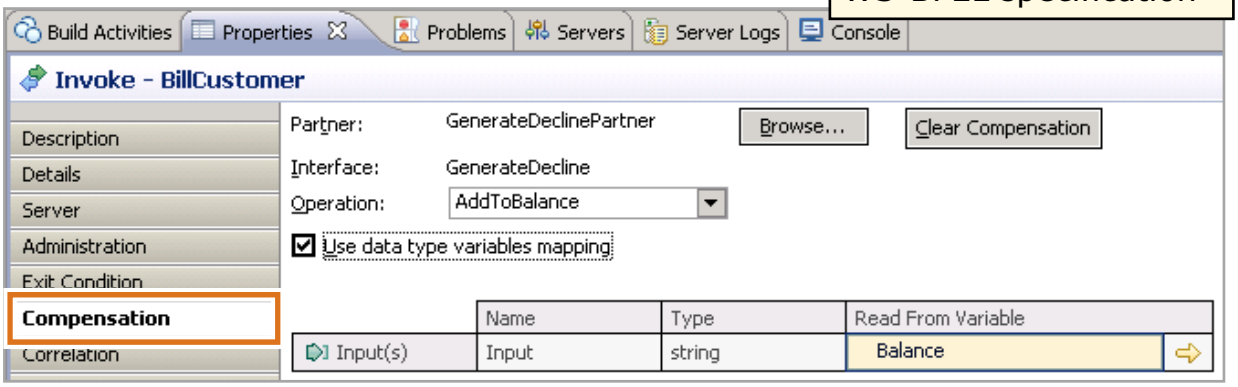
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Add a compensation operation to an invoke activity in microflows

1. Select an *invoke* activity in the business process
2. Select the **Compensation** tab from the **Properties** view
3. Choose the compensation operation from the partner
4. Include additional information that is needed to run the compensation operation by using the **Input Variable** field
   - Compensation operations can be defined for both long-running processes and microflows

Compensation is an IBM extension to WS-BPEL specification



© Copyright IBM Corporation 2017

Figure 6-14. Add a compensation operation to an invoke activity in microflows

The steps for adding a compensation operation to a Java snippet or human task activity are similar to the steps in this slide. When selected, both of these latter types of activities have a **Compensation** tab in the **Properties** view.

To create compensation logic for an invoke activity in a microflow, you must define a location to store the details of this state. To define, proceed as follows:

1. To begin, you must make this process a *microflow***.** Click an empty area of the canvas, click the **Details** tab in the **Properties** area, and clear the **Process is long-running** check box.
2. Choose a setting for **Compensation Sphere**. You have the following two options:
   - ***OptionDescriptionSupports:*** Use this setting when this microflow can run without a compensation service.
   - ***Required:*** Use this setting when this microflow needs a compensation service.

For each invoke activity in this microflow, proceed as follows:

1. Click the **Compensation** tab in the **Properties** area.
2. Browse to a reference partner, and select an appropriate operation to store the original condition of the activity.
3. Browse to an input variable to store the original value of the activity.

© Copyright IBM Corp. 2017

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

If the microflow must be compensated, it looks at these values and restores them to the activity.

If the larger long-running process calls this microflow, the compensation values for each activity are promoted to the parent process and stored there.

If it fails, the process as a whole must be compensated.

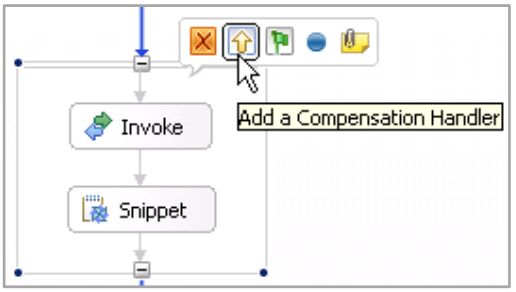
© Copyright IBM Corp. 2017

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

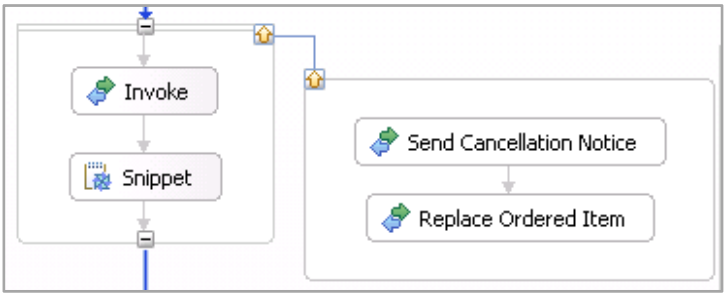
IBM Training

Add a compensation handler to a long-running process

- Compensation handlers represent a group of activities that are run during process compensation
  - Handlers provide an alternative to pairing compensation operations to a particular *invoke* operation
  - Handlers can be associated with a particular *invoke* activity or a local scope in the process



- To add a compensation handler:
  - With a scope or *invoke* activity selected, choose **Add a Compensation Handler** from the action bar
  - Add activities within the compensation handler



© Copyright IBM Corporation 2017

Figure 6-15. Add a compensation handler to a long-running process

A compensation handler runs only if the activity with which it is associated completes successfully. To compensate a process, proceed as follows:

1. To begin, make the process as a long-running process. Click an empty area of the canvas, click the **Details** tab in the **Properties** area, and select the **Process is long-running** check box.
2. Drop an invoke activity or a scope activity onto the canvas. If you are creating a compensation handler for a single activity, use invoke. If it is for a group of activities, use a scope.
3. Click this activity to start the action bar.
4. In the action bar, click the **Add compensation handler** icon as shown in this image.
5. Populate the compensation handler with the activities that are needed to create the compensation logic.

If the process fails any time after the invoke or scope activities were successfully committed, then the associated compensation handler is executed.

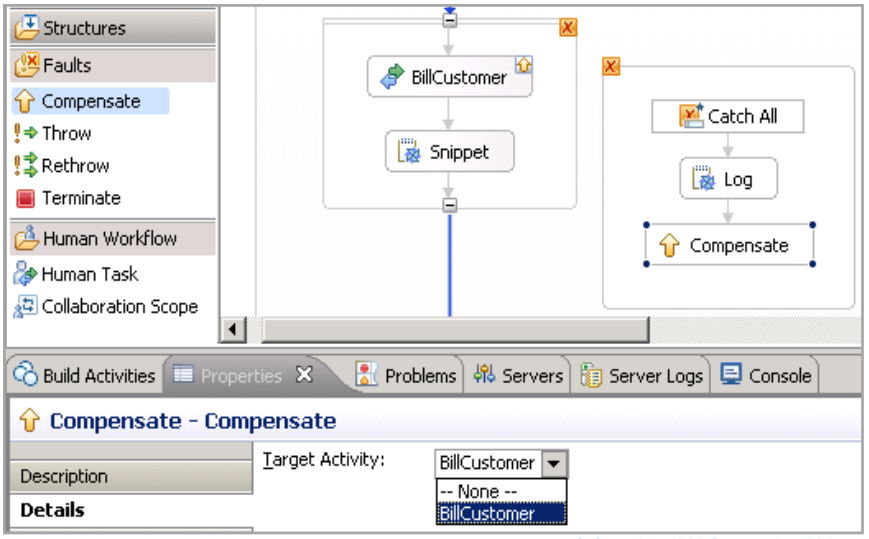
© Copyright IBM Corp. 2017

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

IBM Training

Trigger compensation explicitly

- Use the *compensate* activity to explicitly trigger a compensation handler or a compensation operation within a business process
- To add a *compensate* activity:
  1. In the process editor, select the *compensate* activity from the palette
  2. Place the *compensate* activity onto the process
  3. With the *compensate* activity selected, choose the **Set compensate** link from the action bar
  4. Draw a link to the appropriate compensate *invoke* activity within scope



© Copyright IBM Corporation 2017

Figure 6-16. Trigger compensation explicitly

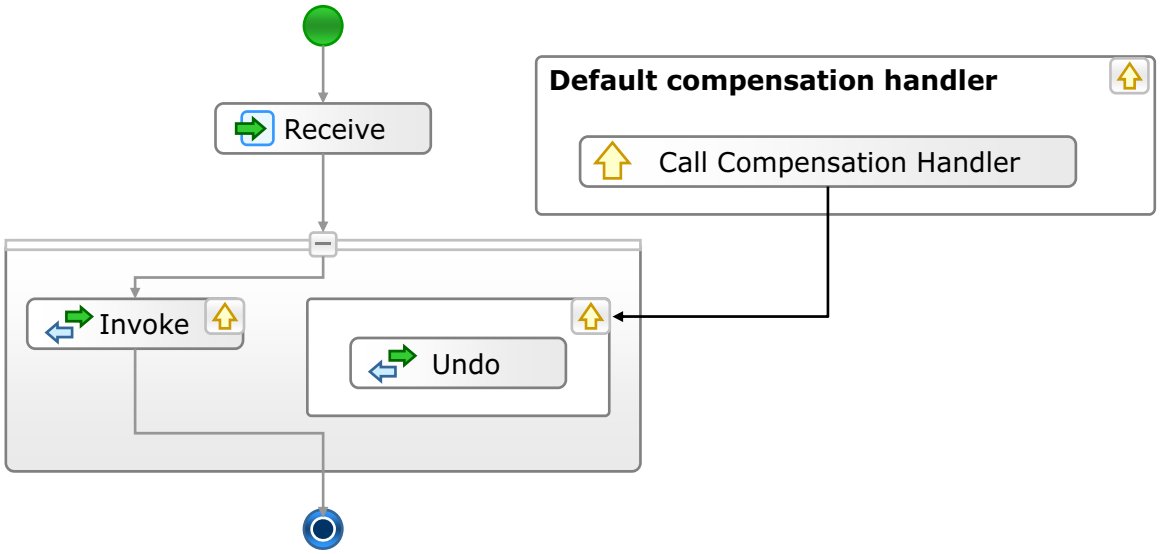
© Copyright IBM Corp. 2017

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

IBM Training

Default compensation handler

- A scope that does not define a compensation handler has an implicit default compensation handler
- The default compensation handler compensates all immediately enclosed and successfully completed subscopes in reverse order of their completion



© Copyright IBM Corporation 2017

Figure 6-17. Default compensation handler

If no compensation handler is explicitly defined for a scope, an implicit compensation handler exists by default.

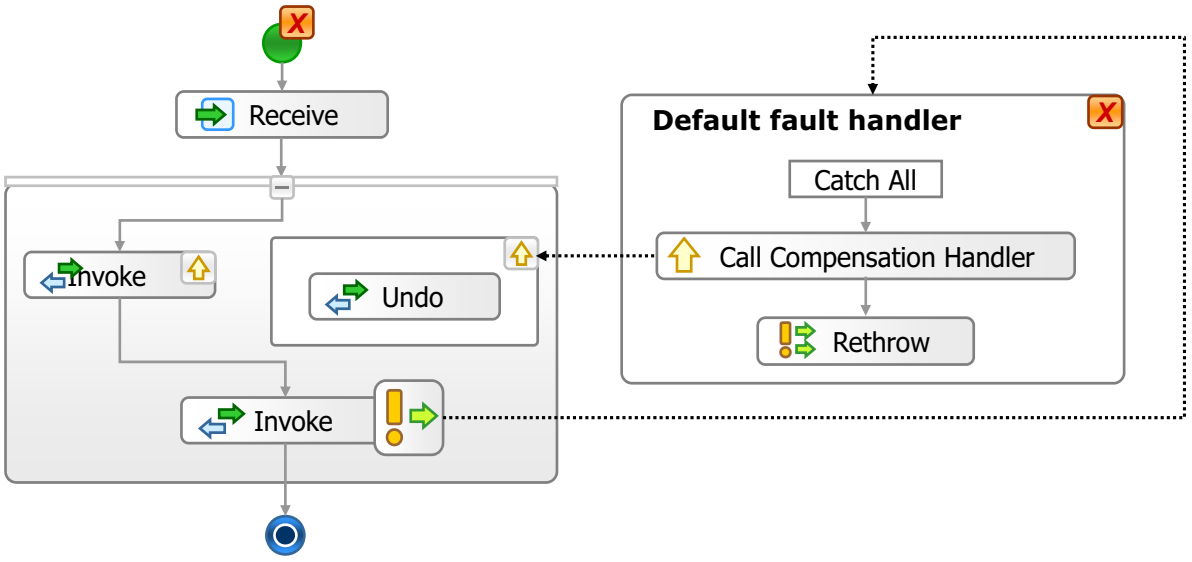
© Copyright IBM Corp. 2017

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

IBM Training

Default fault handler

- Each scope has an implicit default fault handler, which handles all faults that another fault handler did not catch
- Compensates all immediately enclosed successfully completed subscopes in reverse order and rethrows the fault



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-18. Default fault handler

Each scope also has an implicit default fault handler to handle all faults that the other fault handler cannot catch.

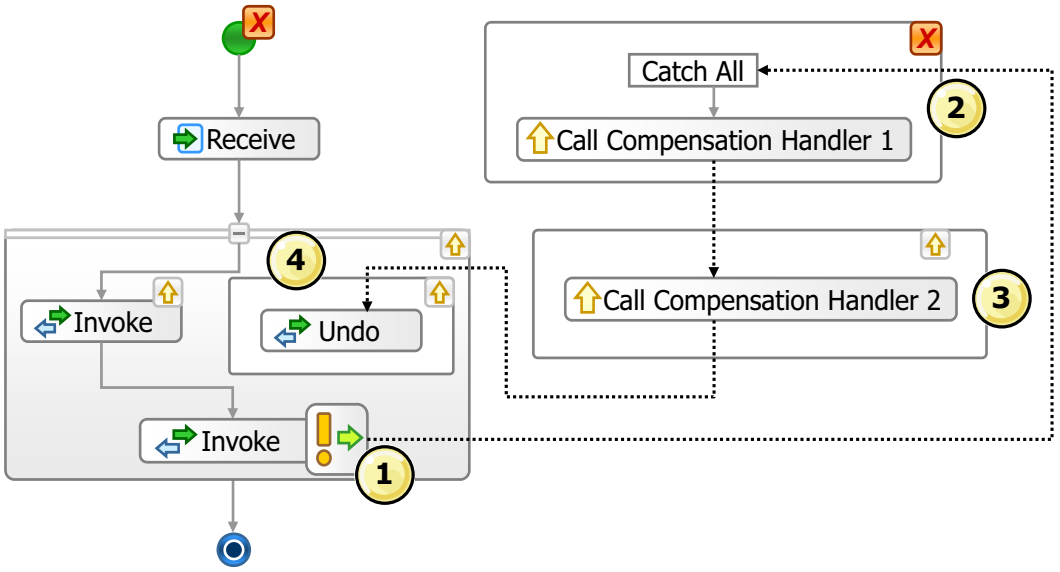
© Copyright IBM Corp. 2017

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

IBM Training

Fault handlers and compensation handlers

- A fault handler can trigger the compensation handler of any immediately enclosed scopes that are completed successfully
- In turn, a compensation handler can trigger any compensation handlers of any immediately enclosed, successfully completed scopes



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-19. Fault handlers and compensation handlers

The following example explains the behavior of compensation handlers at a local scope and activity level. *A compensate* activity triggers each compensation handler explicitly.

1. A *throw activity* near the end of the business process throws a fault.
2. The *fault activity* escalates to a user-defined fault handler at the process level. The *catch all* handler accepts the fault and invokes a compensation operation within one of the local scopes in the process.
3. The *compensation handler* for the local scope calls a compensation handler for the *invoke* activity within the scope.
4. The *activity level compensation handler* invokes a service to restore the situation to a balanced state.

See the *Fault handling behavior* slide in the previous unit for an explanation about default fault handlers.

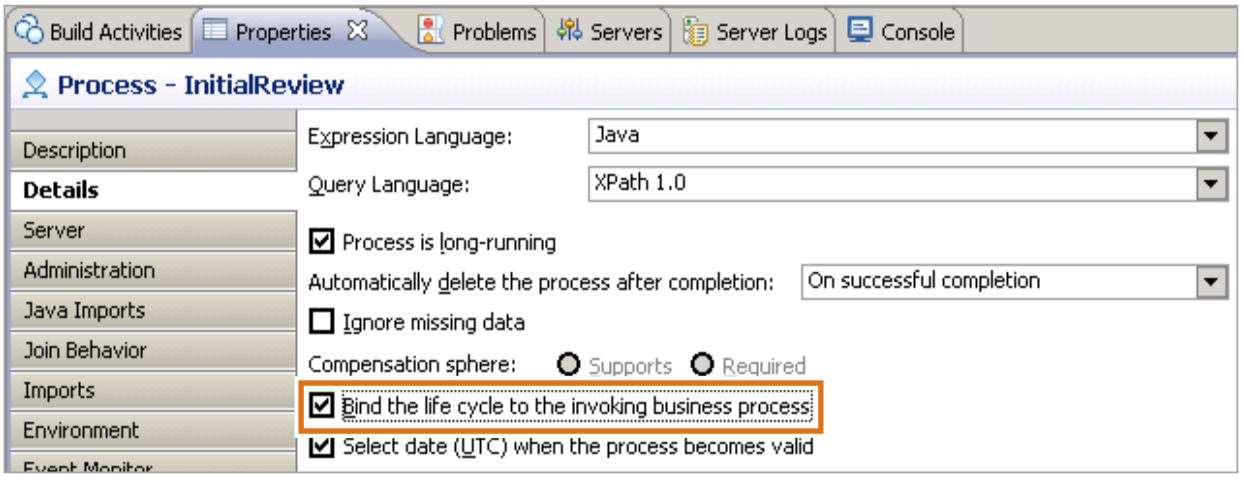
© Copyright IBM Corp. 2017

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

IBM Training

Compensation and subprocesses (1 of 2)

- A subprocess is a process that another process calls, and its lifecycle is bound to an invoking process
- When the *Bind the lifecycle to the invoking business process* property is selected, the process runs as a child process and ends when the invoking business process ends



© Copyright IBM Corporation 2017

Figure 6-20. Compensation and subprocesses (1 of 2)

The *Bind the lifecycle* property sets the process as a child subprocess when it is set.

A subprocess is automatically compensated when:

- A fault handler of the scope that encloses the parent activity is entered
- The compensation handler of the scope that encloses the parent activity is entered

Subprocesses are compensated before the actual handler is run.

A subprocess must complete successfully to be compensated.

If a scope contains multiple subprocesses, the compensation order corresponds to the reverse order of the forward execution.

Subprocess compensation cannot be turned off.

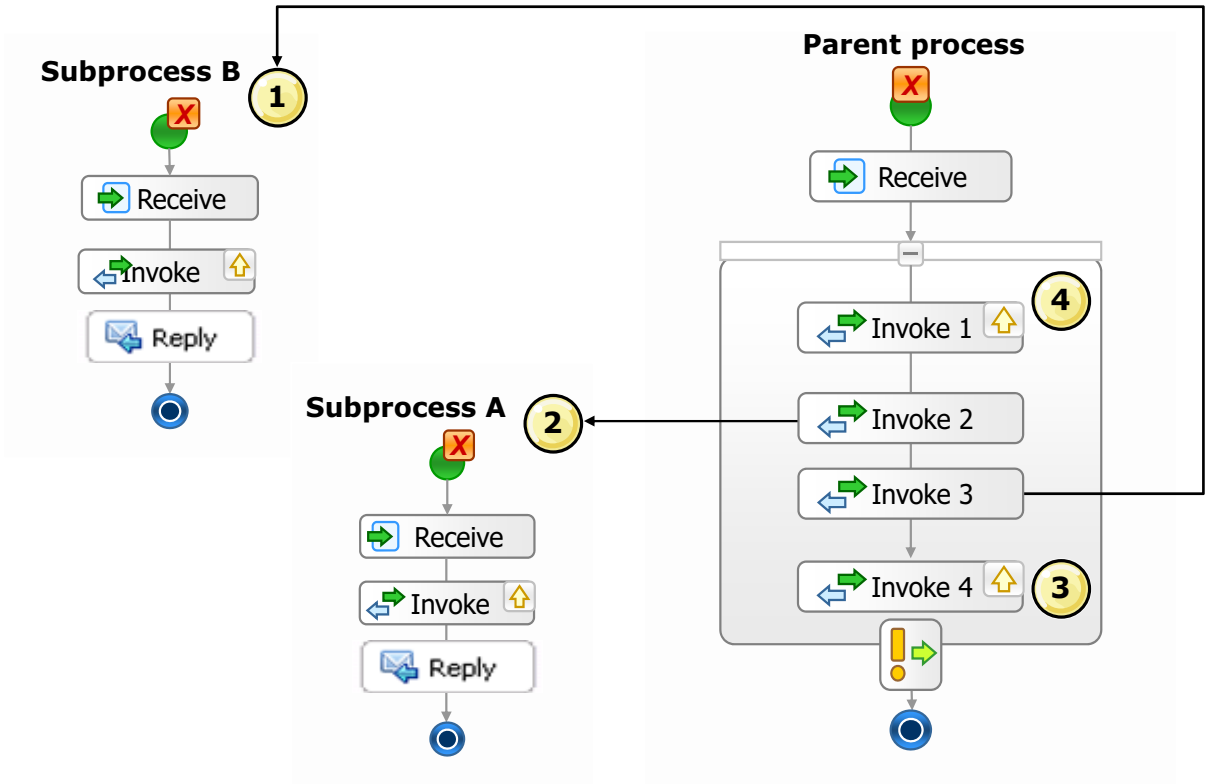
Compensating a long-running subprocess means to run all compensation handlers of top-level scopes in reverse order of their completion.

© Copyright IBM Corp. 2017

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

IBM Training

Compensation and subprocesses (2 of 2)



© Copyright IBM Corporation 2017

Figure 6-21. Compensation and subprocesses (2 of 2)

The execution order of the invoke activities is **Invoke 1 > Invoke 2 > Invoke 3 > Invoke 4**. If Invoke 4 is the last successfully executed activity, then Invoke 4 must compensate first. However, Invoke 2 and Invoke three call subprocesses. In such a case, subprocesses are compensated first. For multiple subprocesses, the order of compensation is the reverse order of their normal execution. Therefore, subprocess B must be compensated first.

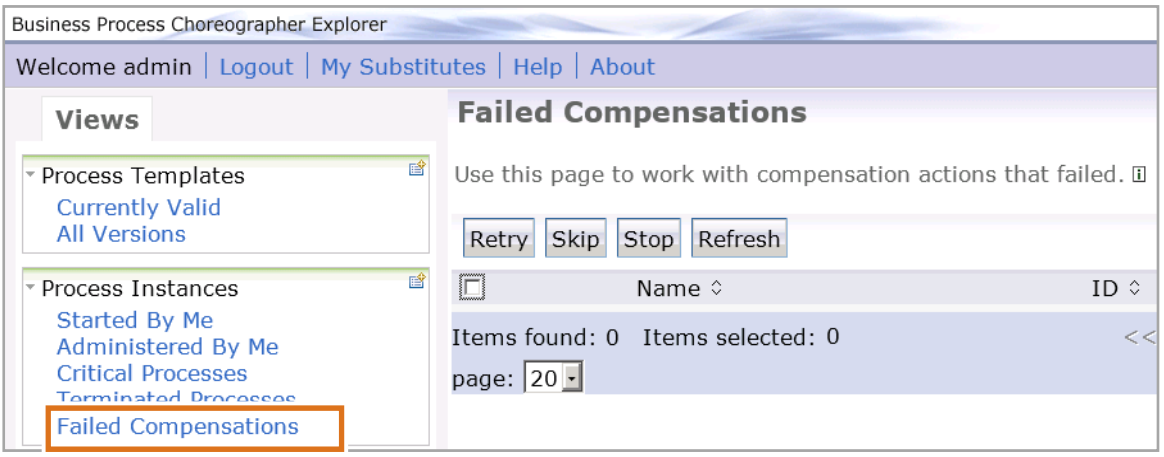
© Copyright IBM Corp. 2017

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

IBM Training

Failure during the compensation

- If a compensation fails, you must work with the failed instance manually through the Business Process Choreographer Explorer



© Copyright IBM Corporation 2017

Figure 6-22. Failure during the compensation

Use the Failed Compensation action in the Business Process Choreographer Explorer to resolve failed compensations.

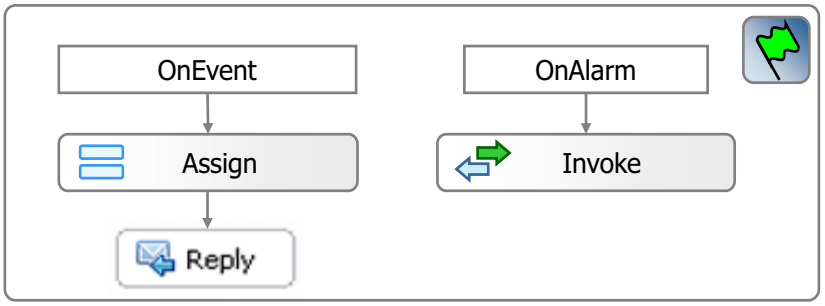
© Copyright IBM Corp. 2017

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

IBM Training

Reacting to changes outside of the business process

- A business process must be able to react to changes outside of its own environment:
  - For example, a client might modify a loan application after submitting the application to the bank
  - The system notifies the bank manager about outstanding loan applications after two weeks
- Event handlers allow processes to accept new messages from the client or timed events without disturbing the design of the process:
  - A particular message from the client triggers a *message event*
  - An elapsed time triggers an *alarm event*



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-23. Reacting to changes outside of the business process

Unlike fault handlers or compensation handlers, event handlers are considered to be a part of a *normal activity* flow in your process.

This slide is an overview of event handlers. With event handlers, you can have a *milestone pattern* in your business processes. In the **milestone** pattern, an activity can be done only when a certain **milestone** is reached and cannot be done after the milestone expires. Stated differently, an activity can run any number of times between the occurrences of enabling and disabling events. Using the event handlers, you can do just that.

An event handler is associated with a scope, and the scope is the **milestone** of the event handler. After the milestone is reached, and before it expires, the event handler can receive events. It can receive a specific event through **OnEvent** or set a timer through **OnAlarm**. The timer begins when the execution path enters the scope.

Needless to say, the event handlers apply to long-running processes only, since a microflow runs in a single thread. The process editor in IBM Integration Designer must allow the addition of an event handler only if it is set to be a long-running process.

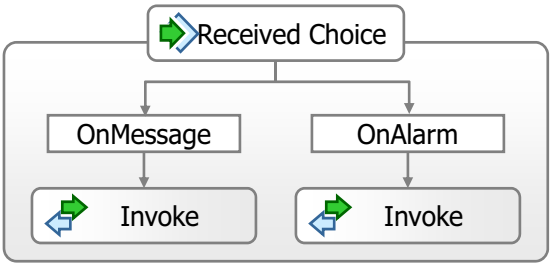
© Copyright IBM Corp. 2017

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

IBM Training

Why use event handlers?

- The *received choice* (*pick*) activity can wait for an incoming message or timed event, but it has the following limitations:
  - Every *received choice* activity must have at least one *OnMessage* case defined; therefore, it cannot be used to model a *milestone* pattern
  - *Received choice* supports only an exclusive choice pattern; it cannot execute two cases simultaneously



- Complicated design patterns obscure the main business process itself:
  - Contrast the *received choice* (*pick*) activity implementation and the *event handler* implementation for the *milestone* pattern

WS-BPEL compensation and event handlers © Copyright IBM Corporation 2017

Figure 6-24. Why use event handlers?

The *receive choice* (pick) activity limitations can be overcome by using concurrent execution paths within a parallel activities (flow) structure.

A wait activity in a separate execution path within a parallel activities (flow) structure can mimic the behavior of an event. To handle multiple cases concurrently, you can place any number of wait and receive activities, each in its own execution thread within a flow. However, this type of design makes the business process complex. Event handlers allow designers to separate these execution paths from the main business process.

Using the *receive choice* activity in WS-BPEL, you can receive events in the middle of a process. However, the main difference between receive choice and event handler is that your business process instance is in a halt state until a specific message arrives, or it times out. Meanwhile, event handlers do not stop the executions of normal activities and fire off another thread upon event arrival. Thus, multiple paths can be executed simultaneously.

© Copyright IBM Corp. 2017

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

IBM Training

Event handler characteristics

- Event handlers are associated with a scope within the process
  - Unlike fault and compensation handlers, event handlers are considered part of the normal operation of the scope
  - Faults that occur within an event handler are handled in the same manner as a fault that any activity throws in the same scope
  - Fault handlers act on faults from the scope and any event handlers that are associated with the scope
- The event handler can run any activity, such as a sequence or a flow, except for the *compensate* activity
- Event handlers react to two types of events:
  - *OnEvent* handlers allow the process to use messages from the client
  - *Timeout* (*OnAlarm*) handlers allow the process to do a task after a specified length of time

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-25. Event handler characteristics

The WS-BPEL 2.0 specification specifies time-based event handlers within an *OnAlarm* element. Within IBM Integration Designer, this type of event handler is known as a *timeout* handler.

Each scope, including the process scope (global scope), can have a set of event handlers. These event handlers can run concurrently and are invoked when the corresponding event occurs. The child activity within an event handler *must* be a scope activity. Events have two types. First, events can be inbound messages that correspond to a WSDL operation of your BPEL. Second, events can be alarms that go off after user-set timers. An event handler *must* contain at least one *OnEvent* or *OnAlarm* element.

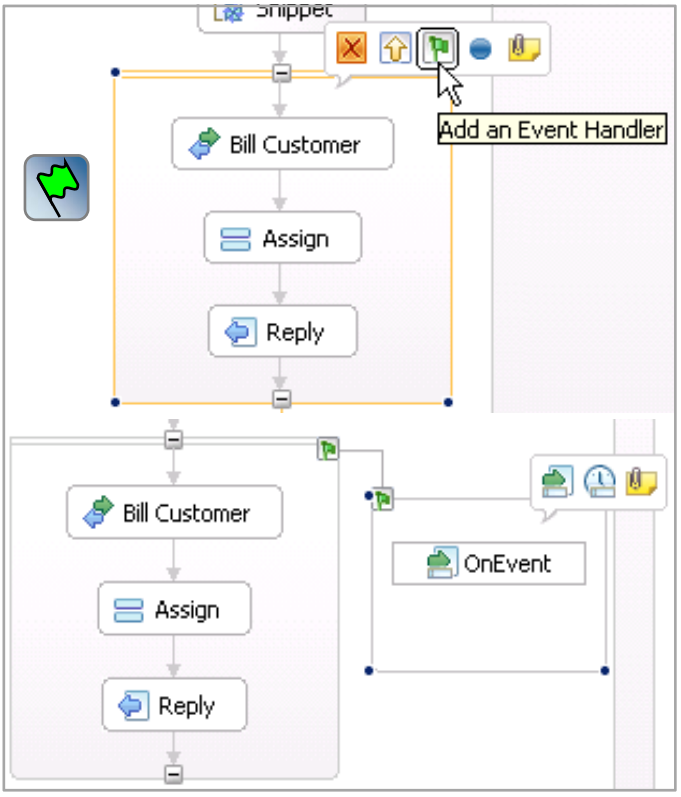
© Copyright IBM Corp. 2017

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

IBM Training

Adding an event handler to a scope

1. Select the scope in the process editor
2. In the action bar, select the **Add an Event Handler** icon
3. Add *OnEvent* and *Timeout* handlers to the event handler in the same manner as a process-level event handler
4. A small event handler icon appears in the upper right corner of the scope
5. Click this symbol to reveal or hide the event handler



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-26. Adding an event handler to a scope

This slide demonstrates the steps to define an event handler on a scope. The steps are very much the same even when you are adding an event handler on the global scope. By default, the editor adds a blank ***OnEvent*** message handler to the event handler. There must be at least one handler that is defined in the event handler.

© Copyright IBM Corp. 2017

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

IBM Training

Message event handler

- An *OnEvent* handler is dispatched when both conditions are met:
  - The process container enables the *OnEvent* handler
  - The client executes the operation that is associated with the *OnEvent* handler
- The *OnEvent* handler works with a copy of the request message:
  - The process container copies the contents of the request message into a local variable that the *OnEvent* handler specifies
  - Each instance of the event handler works with its own private copy of all variables and control information local to the event handler instance, which is not shared with the other instances
- Multiple instances of the same *OnEvent* handler can be dispatched simultaneously if it is associated with a one-way operation

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-27. Message event handler

The **OnEvent** element indicates that the specified event waits for a message to arrive. The interpretation of this element and its attributes is similar to a receive activity. A message arrives through an operation that is defined in your WSDL.

The variable attribute, if it exists, identifies a variable local to the event handler that contains the message that is received from the client application.

As stated in the slide, multiple instances of the same OnEvent handler can be dispatched simultaneously if it is associated with a one-way operation, and the scope is active.

What makes a scope active? As soon as the first activity that is defined in the scope gets executed, the scope is active.

What makes a scope expire? As soon as the last activity in the scope completes its execution, the scope expires.

© Copyright IBM Corp. 2017

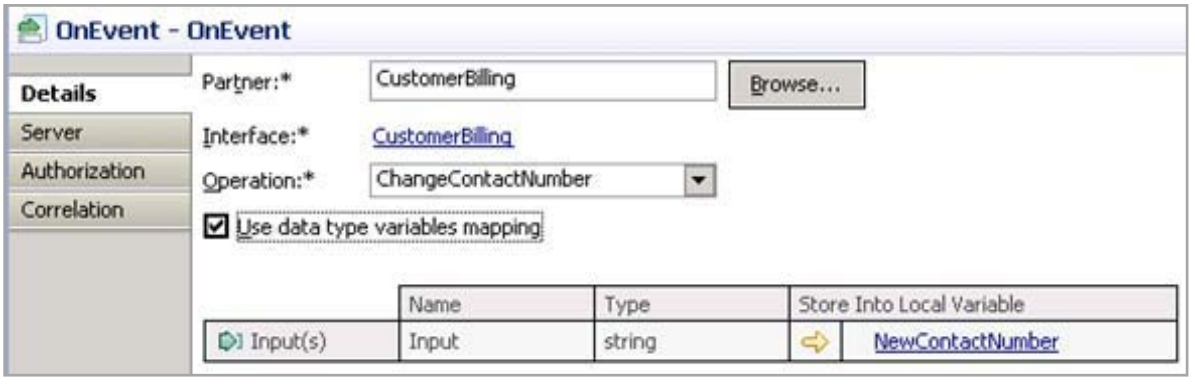
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Adding and configuring an OnEvent handler

The following instructions describe how to listen to messages from the **`ChangeContactNumber`** operation from the client partner:

1. Open an event handler in the business process
2. Add an *OnEvent* handler for the **`ChangeContactNumber`** operation from the client partner
3. Enter the name of the local variable that represents the request message



© Copyright IBM Corporation 2017

Figure 6-28. Adding and configuring an OnEvent handler

This slide demonstrates how IBM Integration Designer supports the creation of event handlers. Each instance of an OnEvent handler promises to have a copy of the request message in a private process variable. The name of that process variable is defined in the **variable name** field of the **OnEvent** handler.

As mentioned in the earlier slides, an OnEvent element is added upon the creation of an event handler by default. However, you might add more than one OnEvent element, and you can have both OnEvent and OnAlarm elements.

OnEvent is where a message comes into the process. To receive an event, your process must have the matching operation.

© Copyright IBM Corp. 2017

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

IBM Training

Timeout event handler

- A timeout handler is dispatched when both conditions are met:
  - The process container enables the timeout handler
  - The alarm event that is associated with the timeout handler occurs
- Use either a ***date*** or ***duration*** to specify when the alarm event occurs
  - A date specifies a deadline in the future
  - A duration specifies the length of time from when the container enables the timeout handler to the time when the alarm event occurs
- Timeout handlers can represent a one-time event or a recurring event until a set deadline
- Only one instance of a timeout handler can be dispatched

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-29. Timeout event handler

The OnAlarm (timeout) element marks a time-driven event, which can be either a specific date or duration event. The clock for the duration starts at the point in time when the parent scope (the scope that directly encloses the event handler) starts. The date expression specifies the specific point in time when the alarm is fired. **Timeout** handlers can represent a one-time event or a recurring event. The repeat expression also specifies a duration (for example, repeat every 5 minutes).

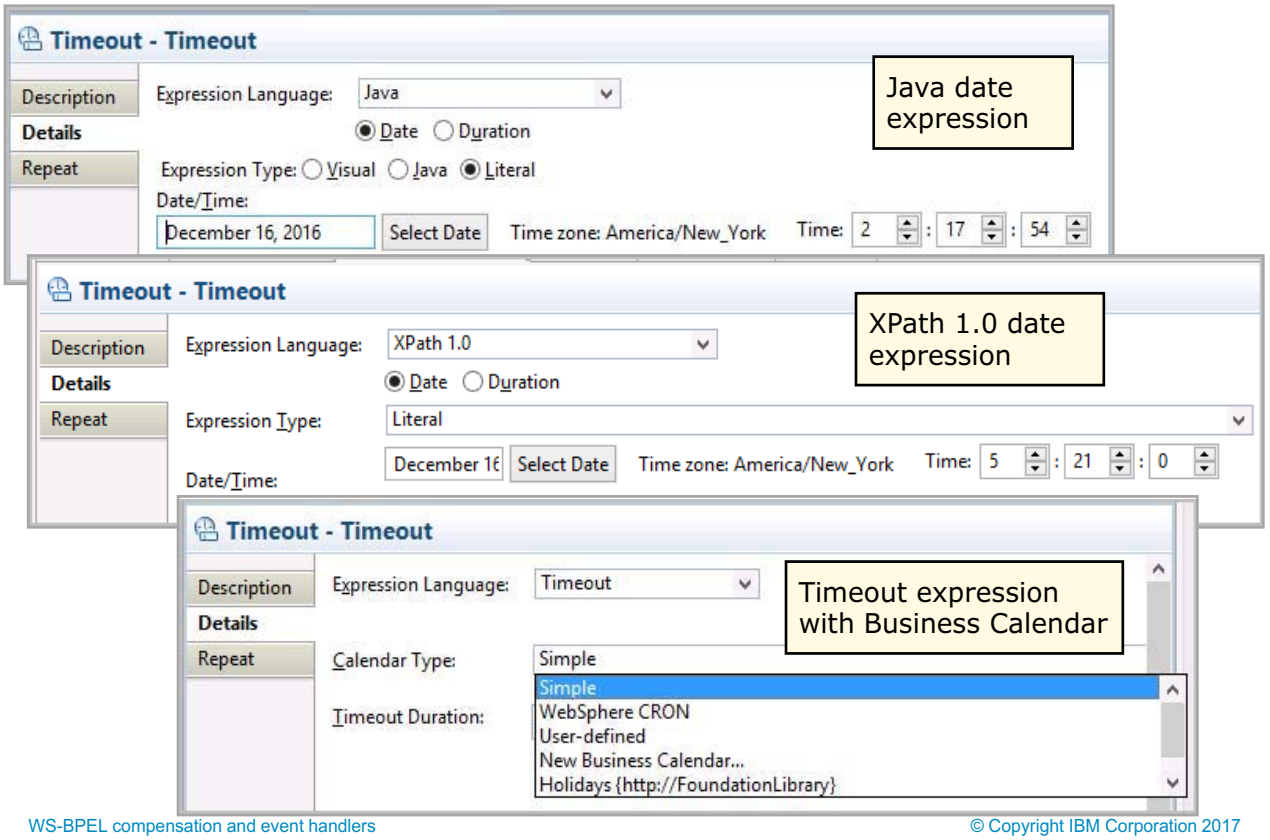
The WS-BPEL specification calls a time-based event **OnAlarm**; however, IBM Integration Designer tool calls it **Timeout**.

© Copyright IBM Corp. 2017

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

IBM Training

Specifying a date-based timeout value



© Copyright IBM Corporation 2017

Figure 6-30. Specifying a date-based timeout value

You can use IBM Integration Designer to set the timeout value in a calendar date expression.

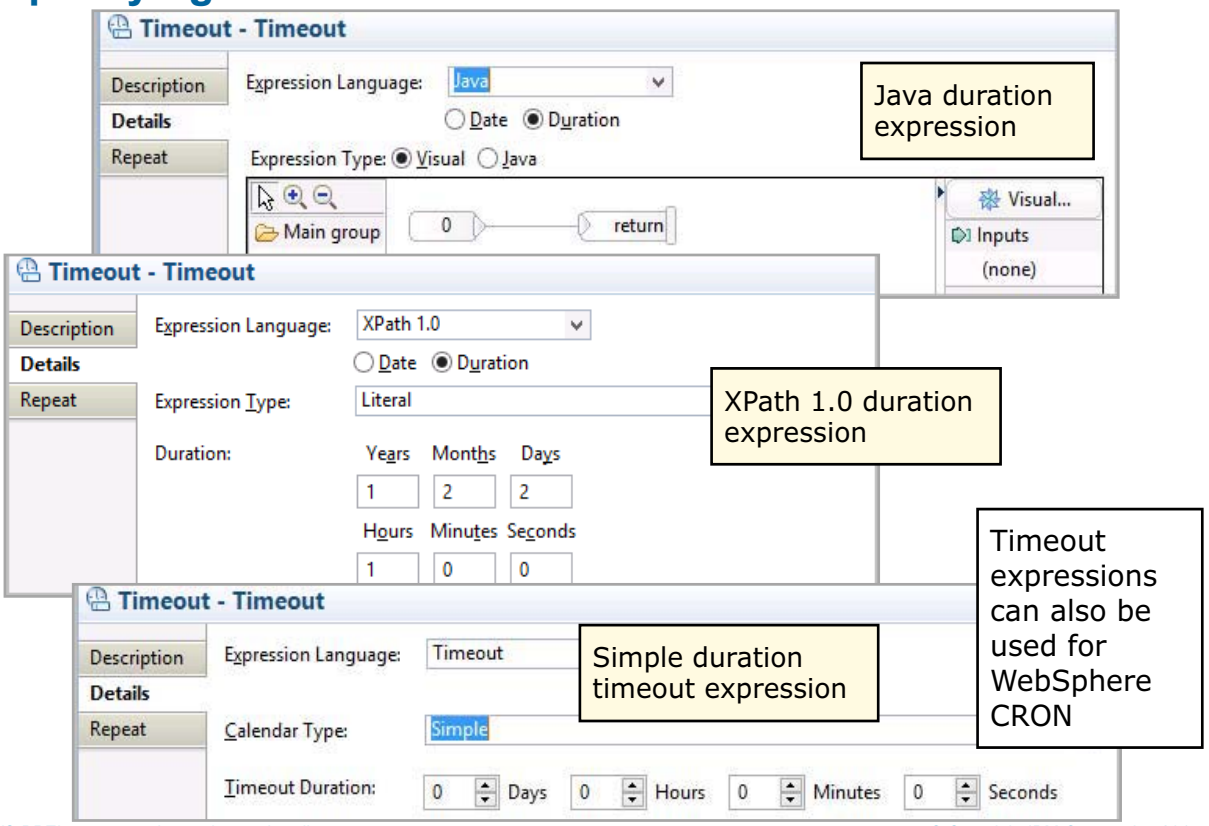
For the `java.util.Calendar` class, the months start from 0 and count to 11. Therefore, January has a value of 0, not 1.

© Copyright IBM Corp. 2017

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

IBM Training

Specifying a duration-based timeout value



© Copyright IBM Corporation 2017

Figure 6-31. Specifying a duration-based timeout value

Java durations are expressed in seconds. XPath 1.0 itself does not support date or time values. The WS-BPEL specification relies on the XML schema *duration* data type to express a time duration.

© Copyright IBM Corp. 2017

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

IBM Training

Event handler lifecycle

- Event handlers are ready to act upon events when their associated scope begins:
  - The process container enables all event handlers that are associated with the process-level scope after it receives the incoming message
  - This design allows the event handler to use information from the incoming message
- All message handlers are enabled during their associated scope
- All event handlers that are associated with a scope are disabled when the scope finishes normal processing
  - If any active event handlers remain after the process finishes all activities within a scope, it waits until the active event handlers are finished

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-32. Event handler lifecycle

An event handler is associated with a particular scope, and that scope is the *milestone* for the event handler. While the scope is active, the event handler can fire off another thread to run a sequence of activities. When the last activity exits the scope, that is when the scope expires. Therefore, if any active event handler is running upon the scope expiration, the scope waits until the active event handlers complete first.

The life of an event handler is the life of its associating scope.

When a fault occurs within the inbound message operation that is specified in OnEvent itself or its associated scope, the fault *must* be propagated to the associated scope first. In other words, if the fault handler is defined in the associating scope, then for any fault that was raised within the event handler, the same fault handler must handle it. As mentioned earlier, the event handler is considered to be a part of *normal* processing.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain the purpose and business value of compensation handlers and event handlers
- Define compensation handlers and compensation activities
- Explain the purpose of event handlers
- Explain the lifecycle of event handlers

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-33. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: A compensation handler reverses operations and tries them again so that a business process can run until completion.
2. True or False: You can attach both a fault handler and a compensation handler to a scope.
3. True or False: A receive activity can serve only one request at a time while an event handler can serve multiple requests.
4. True or False: An event handler can be disabled in a running business process.

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-34. Review questions

Write your answers here:

1.

2.

3.

4.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: A compensation handler reverses operations and tries them again so that a business process can run until completion.
   The answer is False. A compensation handler handles the fault according to the implementation.
2. True or False: You can attach both a fault handler and a compensation handler to a scope.
   The answer is True.
3. True or False: A receive activity can serve only one request at a time while an event handler can serve multiple requests.
   The answer is True.
4. True or False: An event handler can be disabled in a running business process.
   The answer is False. An event handler cannot be disabled.

WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-35. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Applying a compensation handler to WS-BPEL

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 6-36. Exercise 5

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

IBM Training

Exercise objectives

- Implement a compensation handler in a business process
- Test compensation handlers in the IBM Integration Designer integrated test environment

© Copyright IBM Corporation 2016

Figure 6-37. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

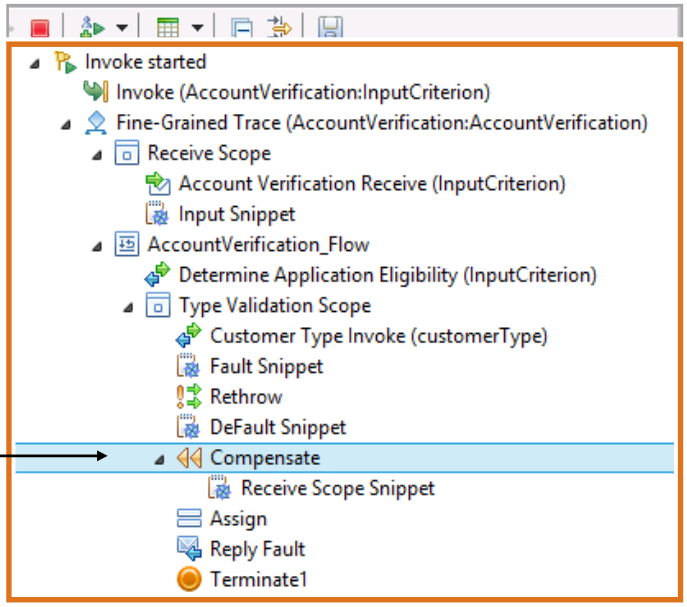
Components that are required for Exercise 5

Prebuilt components that are imported in the lab:

1. **AccountVerification BPEL**
2. **Default snippet**

New components that you create in the lab:

1. **Default Compensate** activity
2. **Compensation handler** for **Receive Scope** activity
3. **Compensation handler** for **Type Validation Scope** activity
4. **Compensation handler** for **Customer Type Scope** activity
5. **Receive Scope** activity
6. **Type Validation Scope snippet**
7. **Input Snippet**
8. **Status Choice** activity



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-38. Components that are required for Exercise 5

In a long-running process, each transaction is committed individually. If the process fails, a compensate activity specifies how to deal with each transaction that needs to be compensated. The compensation handler of a scope, or a process, triggers the compensation.

AccountVerification is already checking the customer type information through a service call to the **CustomerType** process. You now add logic to the **AccountVerification** process to also check the customer status and raise a fault for a detected inactive customer status setting. The logic checks the status of the customer. If the customer is active, no further action is necessary (other than a status message to the server logs or console).

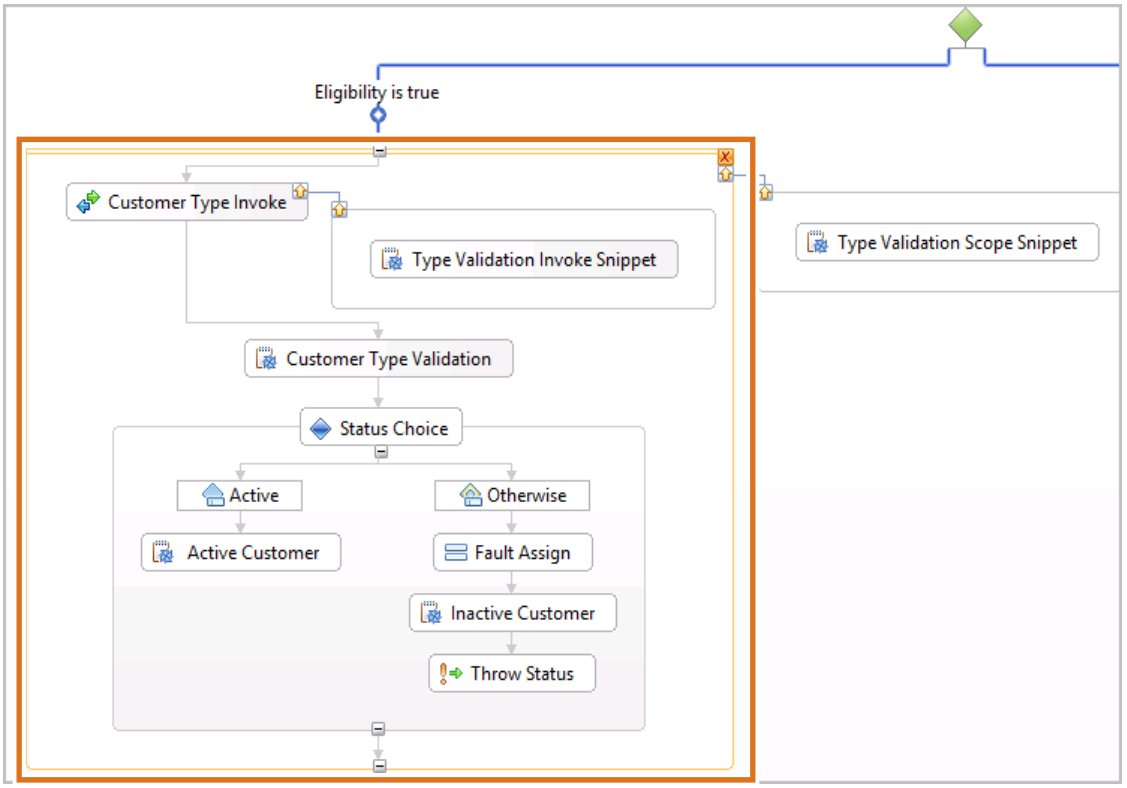
For inactive customer status, a fault is thrown.

© Copyright IBM Corp. 2017

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

IBM Training

Building specific blocks of the AccountVerification process in Exercise 5



© Copyright IBM Corporation 2017

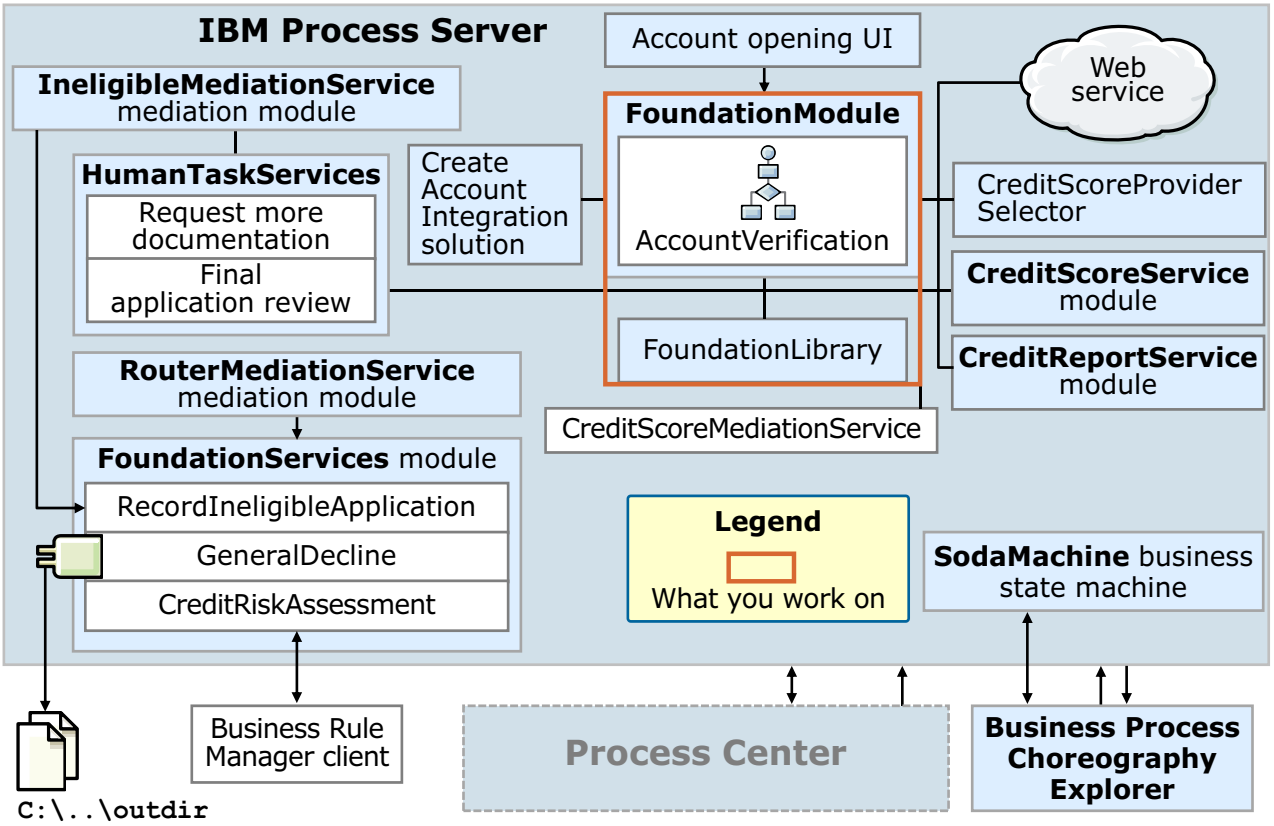
Figure 6-39. Building specific blocks of the AccountVerification process in Exercise 5

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 5



WS-BPEL compensation and event handlers

© Copyright IBM Corporation 2017

Figure 6-40. Course business scenario: What you work with in Ex 5

In this lab, you work with IBM Integration Designer.

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

Unit 7. Business state machines

Estimated time

01:00

Overview

This unit defines the concept of business state machines and describes their value in process integration applications. The unit also lists and defines the elements of a business state machine and describes its runtime behavior.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Define the purpose and business value of using a business state machine
- List the components of a business state machine
- Explain the runtime behavior of a state machine
- Describe the differences between state machines and business processes

Business state machines

© Copyright IBM Corporation 2017

Figure 7-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Introduction to business state machines
- Elements of state machines

Business state machines

© Copyright IBM Corporation 2017

Figure 7-2. Topics

© Copyright IBM Corp. 2017

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

7.1. Introduction to business state machines

© Copyright IBM Corp. 2017

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

IBM Training

Introduction to business state machines

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

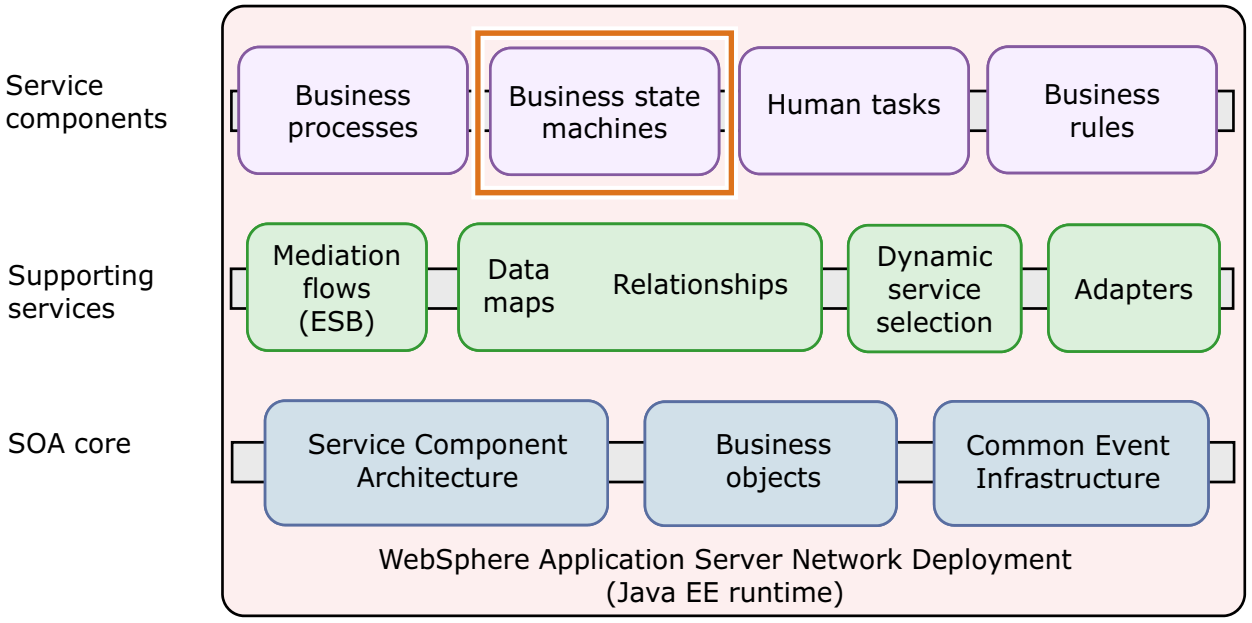
Figure 7-3. Introduction to business state machines

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

IBM Training

Business state machines are service components

- Business state machines are part of the service component layer



Business state machines

© Copyright IBM Corporation 2017

Figure 7-4. Business state machines are service components

In the first row, service components represent artifacts that contain business logic. These artifacts are run on IBM Process Server, which provides a WS-BPEL compliant process engine. This engine is also used to run state machines as business processes. Business state machines allow another way of modeling business processes through states and events, as opposed to the graph-oriented, sequential approach of business processes.

© Copyright IBM Corp. 2017

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

IBM Training

States and state machines

- A state machine model describes dynamic behavior of an application or business process
  - Focuses on the events that cause a transition from one state to another
- A business state machine is an implementation of a business process model that “executes” (moves from one state to another) based on real-time events
- Based on the Unified Modeling Language concept of a state machine
  - XML representation of a state machine is contained in a SACL (State Adaptive Choreography Language) file
  - The SACL file is used to generate a BPEL representation
- State machines are useful for preserving and querying process state information

Business state machines

© Copyright IBM Corporation 2017

Figure 7-5. States and state machines

The concept of a state machine introduces the idea that your process is in a unique state at any particular time and asks what can you do while in that state. For example, if a DVD is out for viewing, you cannot rent it. While it is in stock but reserved, you cannot rent it either. The state of the DVD determines the actions that can be performed. When you develop a business process, your focus is the flow of individual activities, which typically occurs in sequence. If the focus of your process shifts toward the different states that your business process can be in throughout its lifecycle, then a business state machine might be a more suitable choice. If you choose a state machine as the implementation type for your process, be sure to consider the following points:

1. What are the states of your process?
2. What are the transitions from one state to another?
3. What are the conditions for each transition?
4. What are the possible events that trigger state changes?
5. Which events cause which transitions?
6. What business objects are used?
7. What interfaces provide the events for the transitions?
8. With which other SCA components or partners must you collaborate?

© Copyright IBM Corp. 2017

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

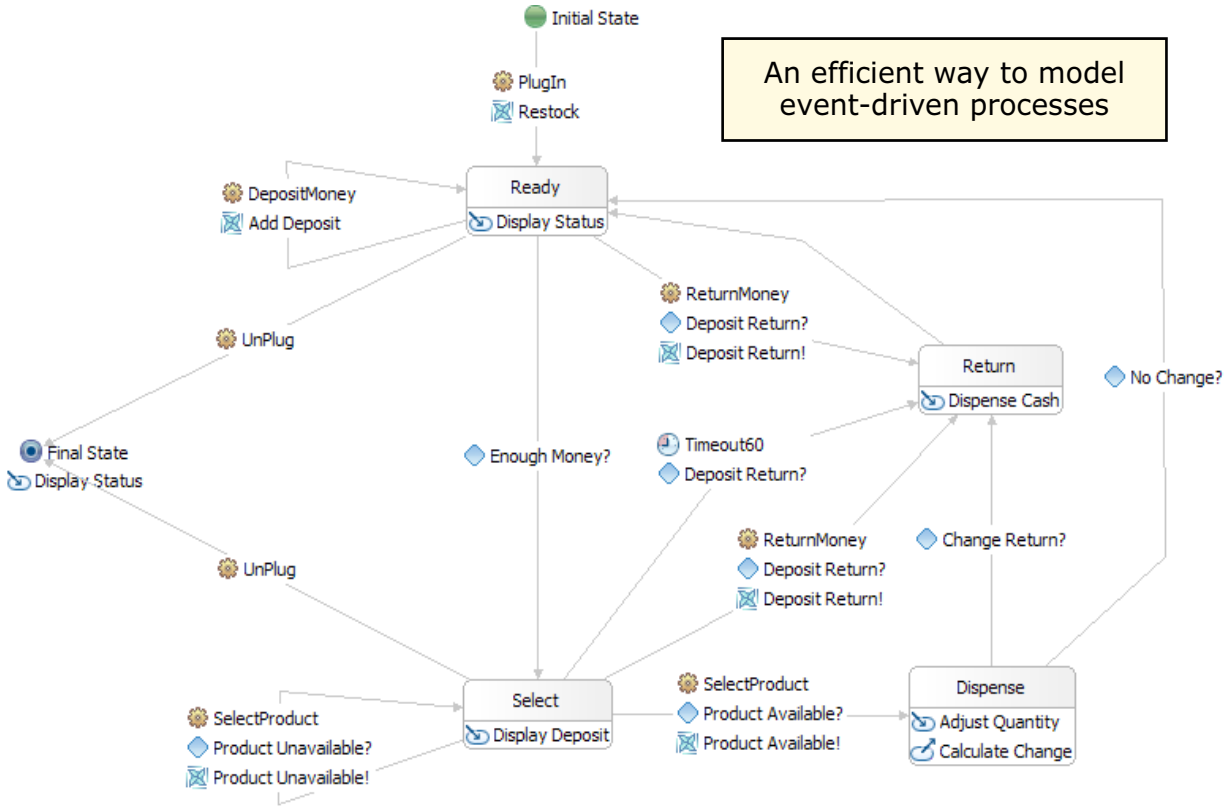
After the business state machine is successfully built, you can see its `.bpel` file in the physical resources view. There is also a `.sacl` file that IBM Integration Designer uses to show the business model in the state machine editor. Do not attempt to edit or customize either of these files.

© Copyright IBM Corp. 2017

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

IBM Training

What is a business state machine?



Business state machines

© Copyright IBM Corporation 2017

Figure 7-6. What is a business state machine?

A state machine is an event-driven business application in which external operations trigger changes that guide the state machine from one discrete mode to another. Each mode is an individual state, and this mode determines what activities and operations can occur.

The IBM Integration Designer tools were designed so that users can easily compose integrative business solutions without programming skills. To this end, you can easily create and develop business state machines in an intuitive graphical programming environment that is called the business state machine editor.

In UML modeling, a state machine is a specification of the dynamic behavior of individual class objects, use cases, and entire systems. Except for operations, when you create a state machine, the object to which you attach the state machine becomes the owner of the state machine. When you create a state machine for an operation, the operation becomes the owner of the state machine. A blank state machine diagram opens when you create a state machine. A state machine diagram is a graphical representation of the sequence of states of an object. The events that cause a state to move from one state to another, and the actions that result from a change in state, are also considered. You can add diagrams to a state machine to describe different behavioral aspects of an object.

You can create state machines to describe classes and systems that have significant behavior. Not all objects require state machines. If the behavior of an object is basic, if it stores or retrieves data,

© Copyright IBM Corp. 2017

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

the behavior of the object might not be important to you, and its state machine might be of little interest. State machines can also contain nested states that represent different hierarchical state levels. You can use nested states to examine complex state changes in objects.

State machines are useful modeling aids for developing real-time or event-driven systems because they show dynamic behavior. You can develop state machines during all phases of a software project and for business modeling. You can use state machines in the following situations:

- During business modeling, you can create state machines to model a use case scenario.
- During analysis and design, you can model event-driven objects that react to events outside the context of an object.
- During analysis and design, you can use several state machine diagrams to show different aspects of the same state machine and its behavior.

© Copyright IBM Corp. 2017

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

IBM Training

State machines are implemented in BPEL

- State machines and business processes are both implemented in BPEL
  - State machines are long-running business processes
  - The Business Process Choreographer container handles the state machines
- Most BPEL components apply to state machines:
  - Invoke partner links
  - Correlation
  - Persistence
  - Transactions
  - Interfaces
- The BPEL processes can represent the state machines and *vice versa*
  - Transitions = BPEL activities
  - States = sequence links

Business state machines

© Copyright IBM Corporation 2017

Figure 7-7. State machines are implemented in BPEL

The business process container included with IBM Process Server manages and runs business processes and state machines since both are BPEL implementations. In IBM Integration Designer, the container is automatically installed with the IBM Process Server test environment. The container is a Java EE application that provides APIs that you can use to create clients.

The business process container is configured to use a Derby database in IBM Integration Designer, which is adequate for development and testing. An enterprise database system such as DB2 can be used for production environments. Database creation scripts are provided. The data store holds the business process templates, business process instance, and state information. In addition, a set of messaging resources are started and configured in the IBM Integration Designer environment.

The overall configuration in IBM Integration Designer can be viewed as a sample type of setup. It is not meant to emulate a full production environment. In production, your security is also activated, so you must configure the business process container security constraints. Referenced partners represent external services (SCA components) that the state machine calls at run time. Referenced partners in state machines function in the same manner as in a business process. Referenced partners can be called from actions (transition, entry, and exit) and conditions. Actions and conditions are described later in this unit.

© Copyright IBM Corp. 2017

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

If the operation that was started has a response, the output is received and the values can be assigned to global variables. When calling external partners, avoid long-running synchronous service invocations. If you need to start another SCA component or web service from a state machine, make it asynchronous so the state machine can be ready for the next event.

The correlation properties of a business state machine uniquely identify a running instance of that state machine. The correlation set in a state machine and the correlation set in a long-running business process work in the same manner. A correlation set consists of:

- The names of the variables
- The runtime contents of the specified variables

© Copyright IBM Corp. 2017

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

IBM Training

Business state machines versus business processes

Business processes and state machines have different attributes

	Business process	Business state machine
Event handling	Some capability to handle events that the handler provides	Most transitions depend on events
Sequential processing	Most steps tend to occur in sequential order	Automatic transitions can provide sequential processing
Looping	Process steps cannot form a loop, except in a *while*, *repeat until, for each*, *generalized flow*, or *collaboration scope* activity	Process steps can lead back to a previous state
Process state	Process state can influence transitions in a parallel activity (flow)	Reaction to events depends on process state
Handlers	Support for fault and compensation handlers	State machines do not support fault or compensation handlers

Business state machines

© Copyright IBM Corporation 2017

Figure 7-8. Business state machines versus business processes

Business processes and business state machines have a number of differences.

© Copyright IBM Corp. 2017

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

IBM Training

State machines in IBM Process Server (1 of 2)

- As BPEL processes, state machines are instantiated from templates
- *Process templates* describe the business process model
  - The template is used to create instances of the state machine
  - Use the administrative console to start, stop, and manage templates
- *Process instances* are entities that exist at run time
  - Instances represent running state machine processes
  - Each instance has specific data and status information that is associated with it
  - Manage state machine instances by using BPEL Process Choreographer Explorer
- Templates and instances are persisted to a database (BPEDB)
  - In IBM Integration Designer, the database is created in DB2 Express
  - For production environments, database creation scripts are provided
- Template's valid-from date can be used to support template versions
  - The valid-from date is used to decide which template to use
  - Instances use that version of the template unless they are migrated

Business state machines

© Copyright IBM Corporation 2017

Figure 7-9. State machines in IBM Process Server (1 of 2)

In IBM Process Server, process templates define state machines. At run time, they are instantiated as process instances.

© Copyright IBM Corp. 2017

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

IBM Training

State machines in IBM Process Server (2 of 2)

- Applications containing state machines are installed and uninstalled like applications that contain business processes
- The test server runs in “development mode” by default
  - Applications are installed and uninstalled easily by using add and remove projects
  - Development mode does not require you to terminate and delete the process instances or stop the process template
- On a production server, applications that contain business processes require manual intervention before they are uninstalled
  - All process instances must be terminated (or allowed to complete) and must be deleted
  - The process template must be stopped (done automatically during uninstall)
- Applications containing processes can be removed without manual intervention by using an included ***Jacl*** script, even though instances and templates still exist in various states
  - Should be used only during testing

Business state machines

© Copyright IBM Corporation 2017

Figure 7-10. State machines in IBM Process Server (2 of 2)

Upgrading a state machine means uninstallation of an old application and installation of a new version. It is important to understand the steps necessary to uninstall an enterprise application (`.EAR` file) which contains a service component that is implemented in BPEL.

Business processes, as deployable artifacts, are contained within an EAR file and exposed as SCA components. They are installed during the application installation process. No additional steps are necessary when installing an application that includes a business process. If you must uninstall an application that contains a business process, you must first stop all running instances of that business process. Doing so prevents the loss of state from a long-running business process by accidentally removing the application. Uninstalling a business process is a three-step process:

1. Ensure that all instances are completed and that they are removed.
2. Stop the template that represents the business process.
3. Uninstall the application.

The uninstallation is made much easier in the development environment by using “development mode,” in which you can uninstall and update the application without requiring you to terminate all business process instances and templates. You can enable this feature for testing, but it must not be enabled in a production environment.

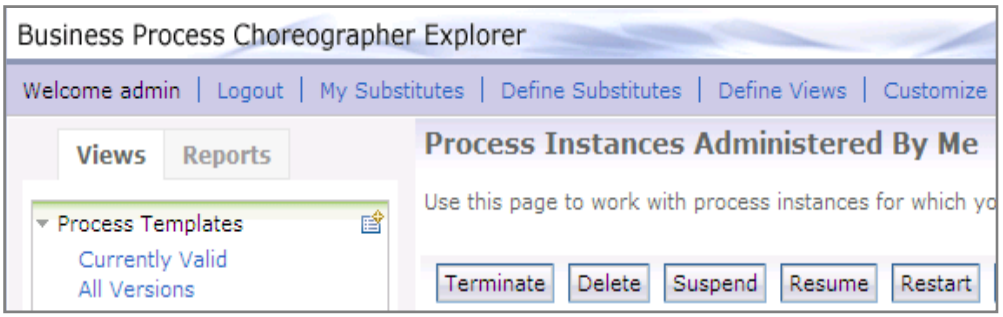
© Copyright IBM Corp. 2017

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

IBM Training

Business Process Choreographer Explorer

- Business Process Choreographer Explorer is included for basic process administration
  - View installed state machine process templates
  - State machine process instances can be viewed, started, terminated, deleted, compensated, tried again, and suspended
- Can be started in IBM Integration Designer from:
  - The servers view
  - **`http://localhost:9443/bpc`**



© Copyright IBM Corporation 2017

Figure 7-11. Business Process Choreographer Explorer

The Business Process Choreographer Explorer client provides basic administrative capabilities and is included with IBM Process Server. You can use the Business Process Choreographer Explorer client to view state machine templates and process instances.

Security can be enabled for this application to limit who can work with or administer an instance. Business Process Choreographer Explorer is based on JavaServer Faces, so many of the views are implemented with JSF tags or components and can be reused in your own custom applications and administrative clients. With a little effort towards understanding the JSF tags, you can build your own clients that extend upon the Business Process Choreographer Explorer.

When using the Business Process Choreographer Explorer client to delete completed process instances and their associated data, you can use selective criteria:

- Process instances that are based on a particular process template
- Process instances that a particular user initiates
- Process instances that are finished before a date or time
- Process instances in a particular state: finished, terminated, or failed

© Copyright IBM Corp. 2017

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

7.2. Elements of state machines

© Copyright IBM Corp. 2017

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

IBM Training

Elements of state machines

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-12. Elements of state machines

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

IBM Training

Overview: Elements of a state machine (1 of 2)

Icon	Element description
	1. **Transition**: A transition is the movement that occurs through the recognition of a triggering event; a transition can contain an action, a guard, or a timeout
	2. **Event** (operation): An event triggers a transition from one state to another
	3. **Action**: An action is an activity that is started on a transition from one state to another
	4. **Guard** (condition): A condition guards the transition and allows execution only when and if it evaluates to “true” • Otherwise, the current state is maintained
	5. **Timeout**: A timeout is an expiration that is imposed upon a state to ensure that it is not maintained indefinitely, waiting for an operation that might never occur

© Copyright IBM Corporation 2017

Figure 7-13. Overview: Elements of a state machine (1 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Overview: Elements of a state machine (2 of 2)

Icon	Element description
	6. **State**: A state is one of several individual stages that represent a business transaction
	7. **Entry action**: An entry action is an activity that starts when a state is entered
	8. **Exit action**: An exit action is an activity that starts when a state is exited
	9. **Composite state**: A composite state is an aggregate of two or more states

© Copyright IBM Corporation 2017

Figure 7-14. Overview: Elements of a state machine (2 of 2)

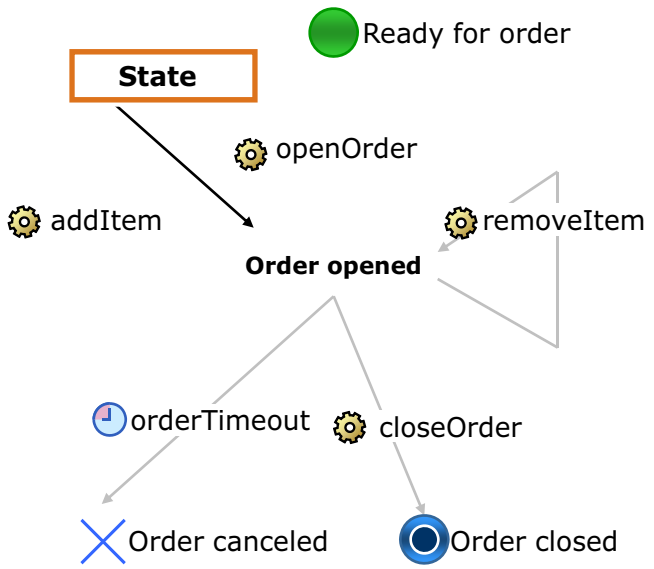
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine components: States

- A state is one of several discrete individual stages that represent a business transaction
- Several different kinds of states exist:
  - ***Initial state***: The first state and starting point of any state machine or composite state
  - ***State***: Use a state to represent an individual discrete stage in your state machine diagram
  - ***Final state***: The final state is where the state machine comes to a normal end
  - ***Terminate***: Use this state to bring a halt to all activity within the state machine



Business state machines

© Copyright IBM Corporation 2017

Figure 7-15. Business state machine components: States

You can use several different types of states in a business state machine.

- **Initial states:** When used in a state machine, an initial state has one outbound transition that defines the operation that starts the state machine. Only one outbound transition can exist, and it cannot be guarded.
- **States:** A simple state can define both an entry and an exit action, and when it is entered, it activates all of its outbound transitions.
- **Final states:** A final state can define an entry action, but it cannot have an exit action.
- **Terminate states:** Terminate states are an abnormal end to a state machine. A terminate state ends all processing. A terminate state can define an entry action, but it cannot have an exit action.

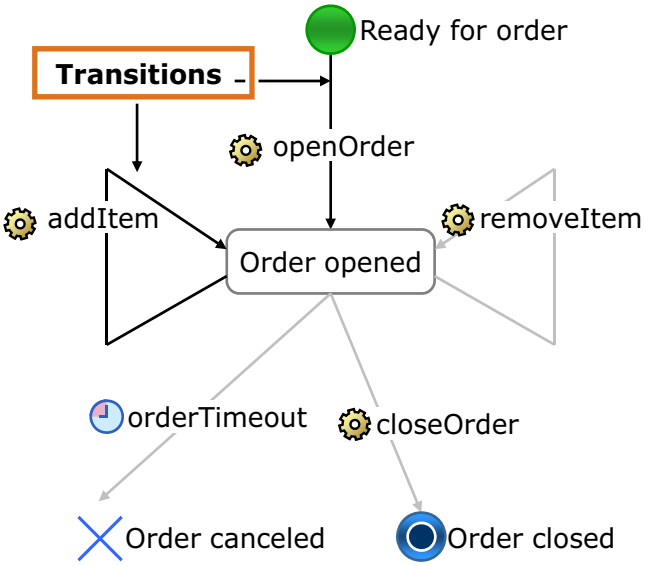
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine components: Transitions

- The transition type (represented as an arrow) determines which transition the state machine follows
- Several types of transitions exist:
  - ***Automatic*** transitions are automatically followed
  - ***Operation*** (or event) transitions are triggered by calling an interface operation, so a message that is sent from outside of the state machine triggers the transition
  - ***Duration*** transitions are triggered after a time that is set in a timeout
  - ***Expiration*** transitions are triggered after a deadline elapses (also by using a timeout)



Business state machines

© Copyright IBM Corporation 2017

Figure 7-16. Business state machine components: Transitions

An automatic transition is a transition with no operation or timeout. An automatic transition is what provides sequential processing in a state machine. Since the state machine always follows the automatic transition, no other outgoing transition can come from a state with an automatic transition. Events are the data that is coming from outside of your state machine. Therefore, the interface of your state machine must define an operation to receive data. In the example that is shown above, the state machine has an interface that contains `openOrder`, `addItem`, `removeItem`, and `closeOrder` operations.

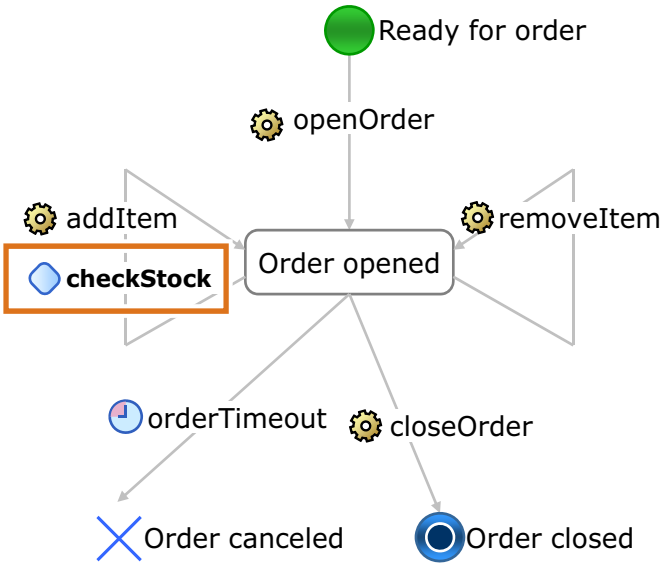
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine components: Conditions

- A ***condition***, or ***guard***, determines whether a state machine should follow a transition that an event triggers
- A condition can be a:
  - Boolean Java expression
  - Visual snippet
  - Service invocation
- If the condition evaluates to false, transitions cannot be followed
  - Returns to the previous state



Business state machines

© Copyright IBM Corporation 2017

Figure 7-17. Business state machine components: Conditions

With conditions, you can model two transitions that accept the same operation (event) but lead to two different states. If a transition condition uses invoke (operation call) as its implementation, the operation that is called must return a Boolean value. A condition (guard) must return a Boolean value regardless of the expression type: Java, visual snippet, or service invocation. If the condition is not met, the state machine remains in the same state. In this example, the client request to add an item to the order is rejected if the item is not in stock.

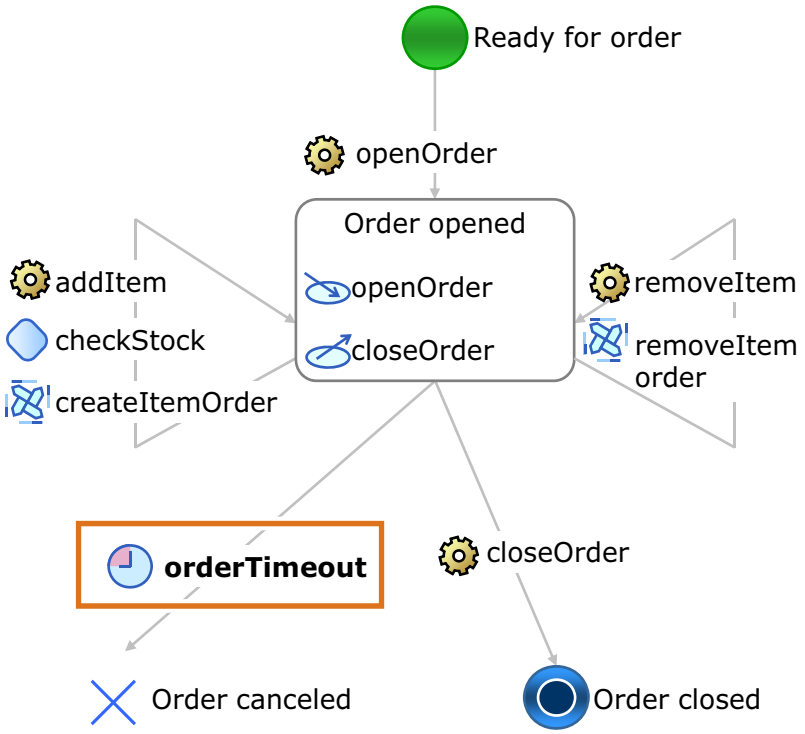
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine components: Timeouts

- A ***timeout*** can cause a transition
  - Durations (for example, one day)
  - Expiration dates (for example, Dec 31, 2018)
- A process should not wait forever for someone to cancel
  - Provides an "escape mechanism" so the process does not run indefinitely



© Copyright IBM Corporation 2017

Figure 7-18. Business state machine components: Timeouts

A timeout can be set to wait for a certain duration of time, or set a specific expiration date before a state moves to another state. Durations or expirations can be expressed by using visual snippets, Java, or literals. For example, if you define a duration on a timer, you can use either a visual or Java snippet to access the attribute in a variable.

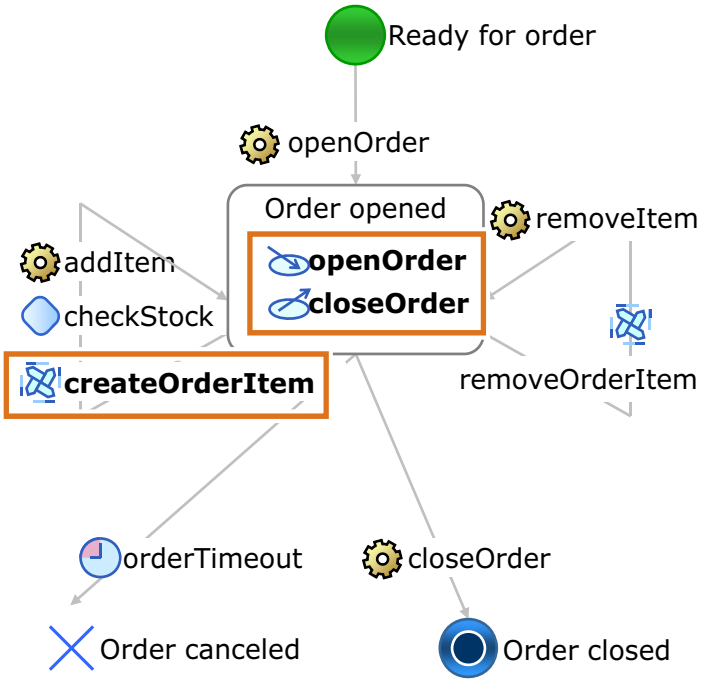
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine components: Actions

- An ***action*** is an activity that the state machine starts
- Three types of actions exist:
  - ***Entry action***: An activity that the state machine starts when entering a state
  - ***Exit action***: An activity that the state machine starts when leaving a state
  - ***Transition action***: An activity that is started during the transition from one state to another
- The implementation of an action can be one of the following types:
  - Visual snippet, Java code, and Service invocation



Business state machines

© Copyright IBM Corporation 2017

Figure 7-19. Business state machine components: Actions

Transitions are logical concepts, not physical events; thus, IBM Process Server supports “actions” that can be done as a side effect of the transition. You can use a visual snippet, Java, or a service invocation to define the action. In a service invocation, you invoke a service (an operation) which a reference partner provides. In this example, an order item is added after a createOrderItem action if sufficient stock exists.

Every state can have at most one entry action, one exit action, or both. When the process enters a state, an entry action runs some logic. Before leaving a state, an exit action might be run. Just like the actions on transitions, you can choose Java, visual snippets, or service invocations to implement entry and exit actions. Entry and exit actions are independent of which transition enters or exits a state. **Entry** and **exit actions** are implemented the same way as **transition actions.** You can add an entry to intermediate states, terminate states, or final states. You can add an exit action to intermediate states only. You cannot define an entry action on the initial state or add an exit action to the final state.

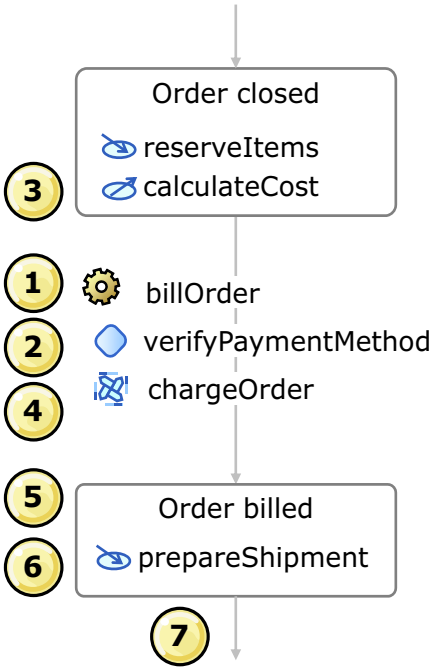
© Copyright IBM Corp. 2017

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

IBM Training

Business state machine component flow at run time

1. The state machine receives an incoming message through an interface operation
   - The container uses one or more correlation properties to retrieve the correct instance and current state
2. For each transition that supports the operation (event), checks which transition condition evaluates to true
   - If no conditions are true, return to the previous state
3. Process the state exit action, if defined
4. Process the transition action, if defined
5. Change the current state to the next state
6. Process an entry action of the new state, if defined
7. Follow any automatic transitions out of the new state or wait for the next event



Business state machines

© Copyright IBM Corporation 2017

Figure 7-20. Business state machine component flow at run time

The numbers in the list match the numbers on the diagram to the right. The diagram shows a process that is waiting in the “Order closed” state until a new `billOrder` event is received.

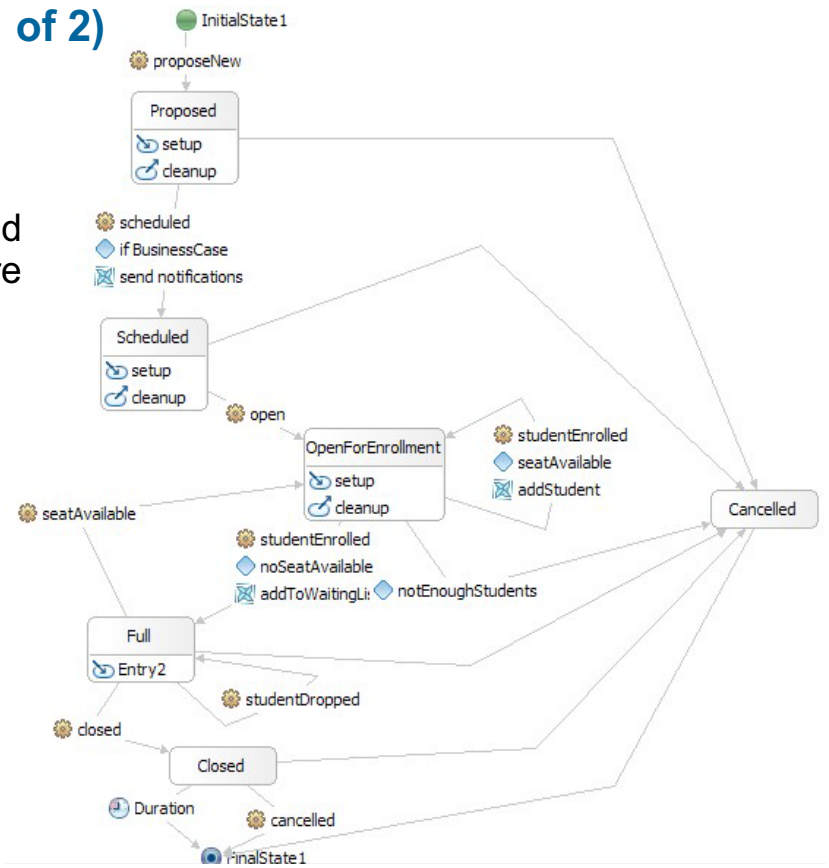
© Copyright IBM Corp. 2017

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

IBM Training

Composite states (1 of 2)

- This example shows a state machine representation of the creation of a course and its enrollment procedure
- **Problem**: Too many transitions all lead to the canceled state
- **Result**: A messy state machine that is:
  - Hard to understand
  - Difficult to maintain
- **Solution**: Composite state: A group of states nested within a state machine



Business state machines

© Copyright IBM Corporation 2017

Figure 7-21. Composite states (1 of 2)

Problem:

Business state machines can have a complex structure that can be difficult to understand and to maintain. For example, a business state machine can end up having too many transitions that are going to the same state, making it hard to read and understand.

Solution:

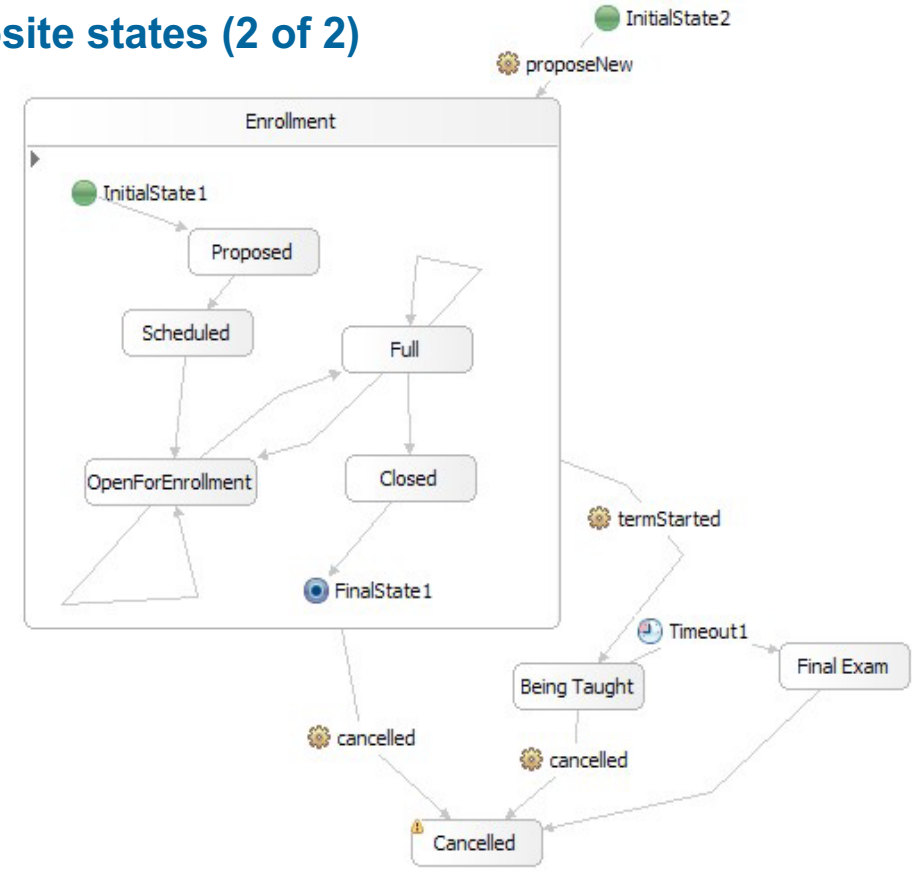
A composite state machine is a state machine nested within a state machine. A composite state is an aggregate of one or more states. Use one or more of them to decompose a complex state machine diagram into an easy-to-comprehend hierarchy of state machines.

© Copyright IBM Corp. 2017

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

IBM Training

Composite states (2 of 2)



Business state machines

© Copyright IBM Corporation 2017

Figure 7-22. Composite states (2 of 2)

The ***composite state*** groups a set of related states into an embedded state machine.

- This construct can be used to break down a complex state machine into a hierarchy of state machines.
- The composite state includes all states that leads to a common state.
- Transitions that enter a composite state begin at the initial state.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Define the purpose and business value of using a business state machine
- List the components of a business state machine
- Explain the runtime behavior of a state machine
- Describe the differences between state machines and business processes

© Copyright IBM Corporation 2017

Figure 7-23. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: An action is always started because it has no condition that is attached.
2. True or False: A guard condition allows the transition to be followed only if a specific condition is true.
3. Which of the following list is not a transition type?
   A. Automatic
   B. Expiration
   C. Timeout
   D. Operation
4. True or False: A condition can be used to start a service.

Business state machines

© Copyright IBM Corporation 2017

Figure 7-24. Review questions

Write your answers here:

1.

2.

3.

4.

5.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: An action is always started because it has no condition that is attached.
   The answer is True.
2. True or False: A guard condition allows the transition to be followed only if a specific condition is true.
   The answer is True.
3. Which of the following list is not a transition type?
   A. Automatic
   B. Expiration
   C. Timeout
   D. Operation

   The answer is C.
4. True or False: A condition can be used to start a service.
   The answer is True.

© Copyright IBM Corporation 2017

Figure 7-25. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Working with business state machines

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 7-26. Exercise 6

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

IBM Training

Exercise objectives

- Explore the building blocks of the SodaMachine with the business state machine editor
- Test the SodaMachine state machine with Business Process Choreographer Explorer

Business state machines

© Copyright IBM Corporation 2016

Figure 7-27. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

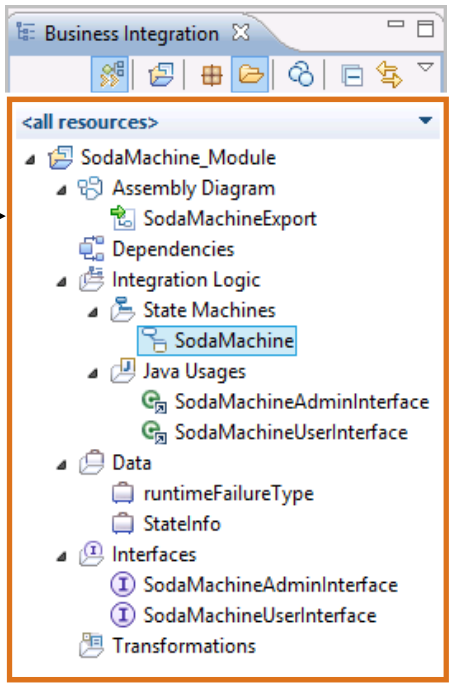
Components that are required for Exercise 6

Prebuilt components that are imported in the lab:

1. **SodaMachine BSM**

New components that you create in the lab:

None



© Copyright IBM Corporation 2017

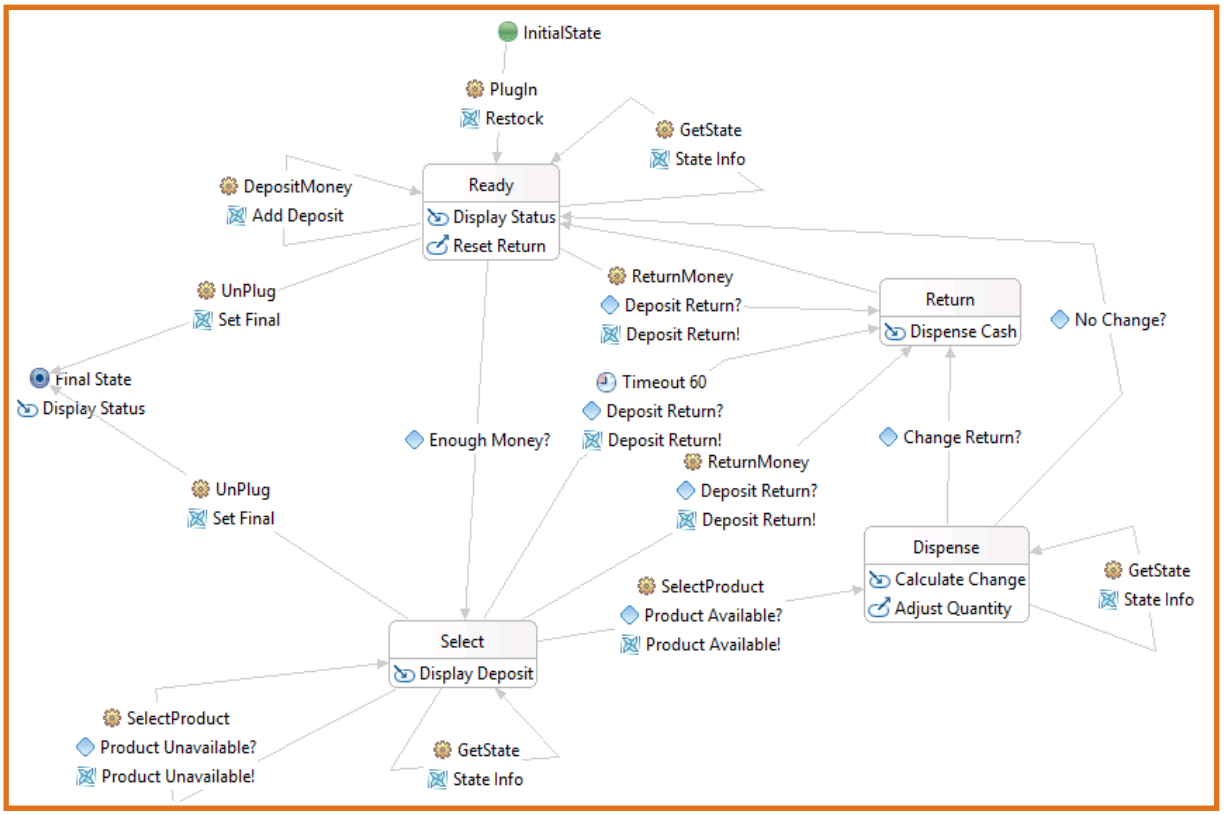
Figure 7-28. Components that are required for Exercise 6

© Copyright IBM Corp. 2017

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

IBM Training

Explore SodaMachine business state machine in Exercise 6



© Copyright IBM Corporation 2017

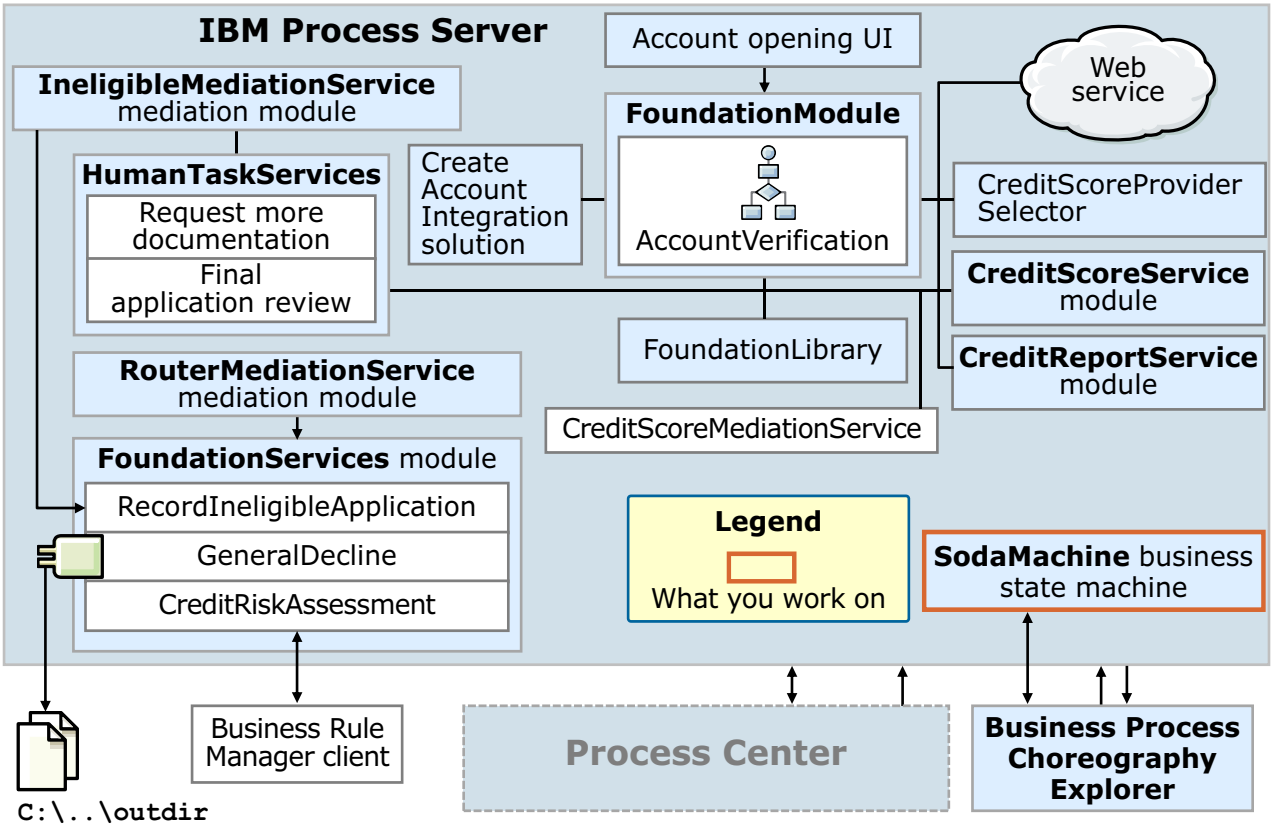
Figure 7-29. Explore SodaMachine business state machine in Exercise 6

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 6



Business state machines

© Copyright IBM Corporation 2017

Figure 7-30. Course business scenario: What you work with in Ex 6

In this lab, you work with IBM Integration Designer.

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

Unit 8. Transactional behavior of SCA applications

Estimated time

01:00

Overview

This unit explains the transactional behavior of SCA applications, including transaction boundaries, propagation, and transaction quality of service qualifiers.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe the general properties of transactions
- Explain the purpose of the transaction quality of service qualifiers
- Categorize activities according to their effect on transactions
- Describe the default transactional behavior of various types of activities
- Describe the implied behavior of activities that choose flexible transactional properties
- Describe the transactional behavior of a business process during a failure

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Transactional behavior concepts
- Transactional behavior during failure

© Copyright IBM Corporation 2017

Figure 8-2. Topics

© Copyright IBM Corp. 2017

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

8.1. Transactional behavior concepts

© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior concepts

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 8-3. Transactional behavior concepts

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Transaction concepts review

A transaction is an indivisible unit of work that updates multiple resources such that all or none of the updates are made permanent

- The atomicity, consistency, isolation, and durability (ACID) properties describe the key characteristics of a transaction
  - **A**tomicity promises that the system executes either all of the updates or none of the updates
  - **C**onsistency maintains that the system remains in a legal state when the transaction begins or ends
  - **I**solation ensures that changes that are made within the transaction are not visible to operations outside the transaction until it is committed
  - **D**urability promises that after a transaction is committed, the updates that the transaction makes will not be undone, even after system failure
- A long-running process consists of a set of **stratified transactions**
  - Each navigation step, including one or more activities, is executed in its own transaction

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-4. Transaction concepts review

A microflow runs under one transaction. This presentation focuses on managing transaction boundaries in long-running processes.

A short-running business process instance runs in a single thread and a single transaction; therefore, it has no crash recovery when failure occurs. In contrast, long-running business processes are multithreaded and have transaction boundaries to persist each state. Therefore, this unit applies to long-running business processes.

The characteristics of transactions, ACID, are straightforward. As an example, think of online shopping. If customers change their minds in the middle of check-out and clear the shopping cart, or close out of a web browser, they are not charged for the product (atomicity). While customers are completing the information about shipment options, the particular transaction remains in the “in progress” state instead of “order complete” or any other state (consistency). Until a customer commits or submits the order, the seller cannot see the customer’s credit card information (isolation). Even if the system fails, when customers log back in to the site, the shopping cart retains what products they placed into it (durability).

© Copyright IBM Corp. 2017

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

IBM Training

Local versus global transactions

- Resource manager local transaction
  - Transaction is contained within one resource manager
- Resource managers include:
  - Relational databases (JDBC): Most databases, including DB2
  - Messaging systems (JMS): SIBus, WebSphere MQ
  - Enterprise information system (EIS) through JCA: CICS, SAP
- Global transactions
  - Multiple resource managers are involved, requiring an external transaction manager such as WebSphere Application Server
  - IBM Process Server transactional models use WebSphere Application Server as the foundation
  - WebSphere MQ and databases can also act as transaction managers

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-5. Local versus global transactions

© Copyright IBM Corp. 2017

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

IBM Training

Rationale for multiple transactions in a process

- In some cases, it is desirable or even necessary to use more than one transaction within a business process
  - Running activities in different, separate transactions ensures that the outcome of one activity does not affect the outcome of other activities
  - An activity that causes a rollback and reexecution does not necessarily require all other activities in the process to roll back and reexecute
  - Example: If an order cannot be billed to the customer on the first try, the order itself does not need to be canceled and re-created
  - Certain activities require more than one transaction to complete
  - Intermediate results become visible only after a commit
- However, running each activity within its own transaction affects the performance of a business process
  - Stretching the transaction boundary beyond one activity might increase the performance of a long-running process
  - This strategy allows long-running processes to approach the performance of a microflow

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-6. Rationale for multiple transactions in a process

On various occasions, you would want a particular block of code (activity) to run in its own transaction. Remember the ACID characteristics of transactions. Certain activities require more than one transaction to complete. For example, you might start a web service that your partner provides to verify your customer's credit history, which takes 5 hours to complete. After you receive the report, the following activity fails, which causes rollback. In such a case, you do not want to resubmit the request and spend another 5 hours to get the customer's credit history report. It is better that this particular activity runs in its own transaction so that you can see the benefit of the durability characteristic that a transaction has.

One thing to know is that ACID characteristics are not cheap. The benefits of transactions must outweigh the need for the performance and speed. In other words, use transactions for your benefit, and have good strategies that work best for your business needs.

© Copyright IBM Corp. 2017

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

IBM Training

Transaction behavior settings in activities

- By design, most activities have one of the following three effects on the current transaction:
  - Certain activities commit the current transaction and start a new transaction
  - Certain activities commit the current transaction
  - Certain activities have no effect on the current transaction
- Certain activities also allow user-defined transaction characteristics
  - This feature allows the process designer to provide the business process choreography engine with hints about transaction demarcations
  - The business process choreography engine reserves the right to browse the process with more transactions if necessary

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-7. Transaction behavior settings in activities

Examining transaction behavior of activities allows the process designer to control the process transaction behavior. The WS-BPEL specification or the business process engine design enforces certain transaction boundaries. An improper activity choice or transaction behavior setting has an impact on performance.

In BPEL, each activity behaves in a certain way to affect transaction boundaries: commit before, commit after, and participate. This unit covers all of these behaviors by looking at some examples.

© Copyright IBM Corp. 2017

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

IBM Training

Activity types and transaction boundaries

- The following chart divides activities according to their influence on transaction boundaries:
  - Activities that end the current transaction and start a new transaction

    Receive | ReceiveChoice | Wait

  - Activities that end the current transaction

    Terminate | Throw | Rethrow | Compensate | Reply

  - Activities that have no effect on transaction boundaries

    Assign | EmptyAction | Data Map | Choice | WhileLoop
    Sequence | Parallel Activities | ForEach | GeneralizedFlow | Scope
    RepeatUntilLoop | CollaborationScope

  - Activities that affect transaction boundaries that are based on user settings

    Invoke | Snippet | HumanTask

Transactional behavior of SCA applications © Copyright IBM Corporation 2017

Figure 8-8. Activity types and transaction boundaries

If you think of what each activity does, the effects on transactions make sense. For example, receive and receive choice activities accept incoming events. Therefore, it makes sense that the current transaction gets committed; then, you can start a new transaction for receive and receive choice activities. The wait activity is basically a sleep timer so that whatever activities were run are committed before the process instance goes to sleep for a certain duration of time.

Terminate, throw, rethrow, compensate, and reply activities all release control to another process so that it makes sense to commit the current transaction upon their execution.

Sequence, parallel, and scope activities organize the flow of basic activities so that they do not have a particular preference on transaction boundaries.

When it comes to invoke, snippet, and human task activities, the behavior must be evaluated case by case. In later slides, more examples are presented.

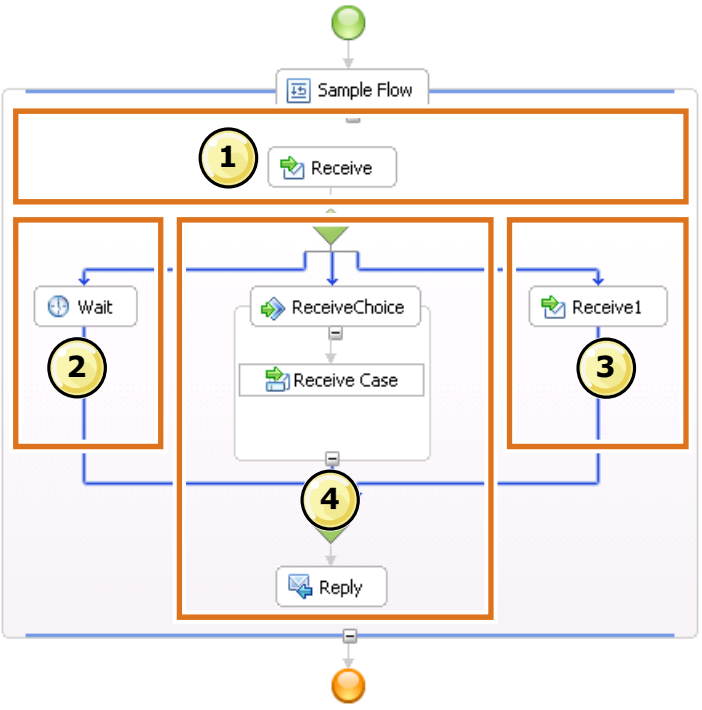
© Copyright IBM Corp. 2017

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

IBM Training

Activities that commit and start new transactions

1. The ***Receive*** activity starts a new transaction
2. The ***Wait***, ***ReceiveChoice***, and ***Receive*** activities each start a new transaction for their respective paths
3. The ***Receive*** and ***Wait*** activities complete and commit their respective transactions at the link to the ***Reply***
4. The ***ReceiveChoice*** activity finishes last and the transaction commits after the ***Reply*** activity



Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-9. Activities that commit and start new transactions

Before the process engine runs the *receive* and *receive choice* activities, it must commit the current transaction. The *isolation* property of transactions prevents activities within the transaction to interact with clients outside the transaction. Therefore, a commit is necessary before the process instance receives a message from the outside.

When several parallel, concurrent transactions synchronize at the *reply* activity, each of these transactions cannot see the changes that are made in the other transactions because of the *isolation* property. To join the three paths into one, two of the three transactions must commit their changes.

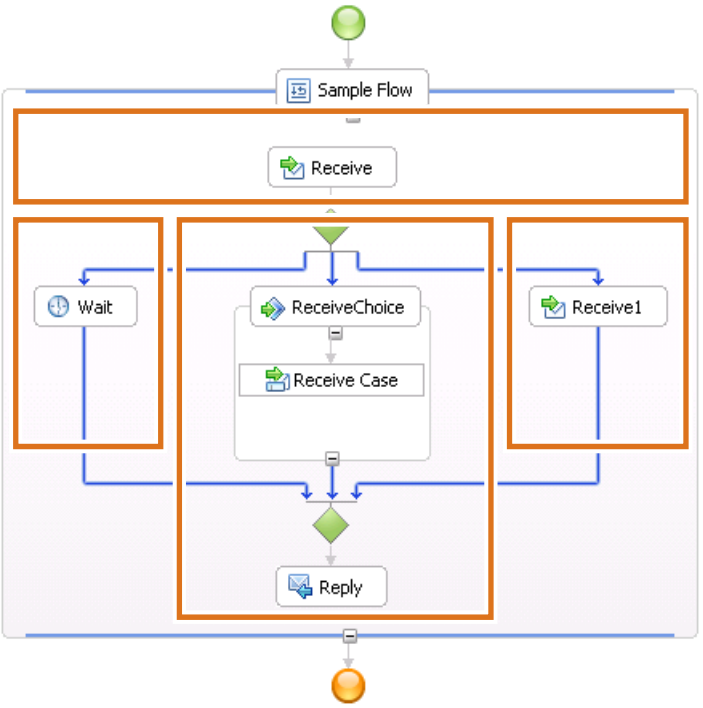
© Copyright IBM Corp. 2017

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

IBM Training

Shifting transaction boundaries

- Transaction boundaries might change every time when you run the process
  - For example, if the *Wait* activity finishes after the *ReceiveChoice* and *Receive*, that transaction includes the final *Reply* activity



© Copyright IBM Corporation 2017

Figure 8-10. Shifting transaction boundaries

As explained on the previous slide, the transaction for the first two concurrent execution paths must commit before the third. Otherwise, the *reply* activity cannot read the link conditions from the first two completed execution paths because of the *isolation* transaction property.

The process engine does not have to create another transaction to run the reply. After the first two transactions commit their respective changes, the last transaction can view the changes that are made in the first two paths.

However, since the three paths are running concurrently, you cannot assume that one path completes before another. Therefore, the transaction boundaries might change every time that the engine runs a process instance.

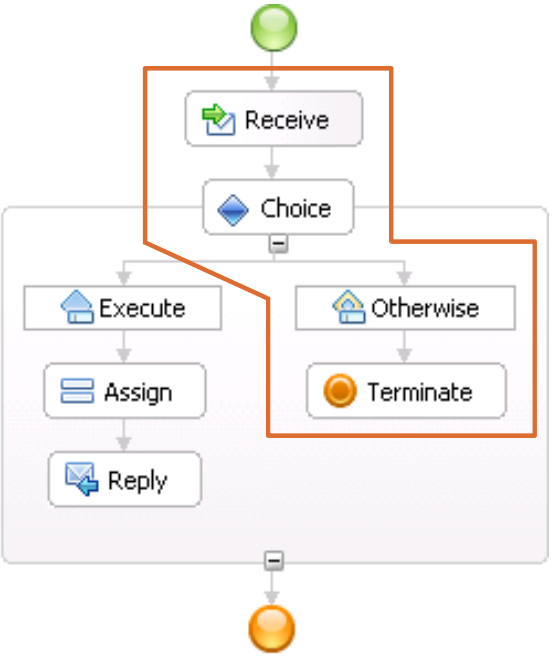
© Copyright IBM Corp. 2017

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

IBM Training

Activities that end (commit) transactions

- Terminate activities have the following effect on processes and transactions:
  - All activities that are currently active are forced to terminate, without fault handling or compensation
  - The business process engine commits the current transaction



Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-11. Activities that end (commit) transactions

A terminate activity commits the current transaction, and then terminates the particular process instance, without fault handling or compensation. Therefore, the typical use of a terminate activity is to start it within the fault handler.

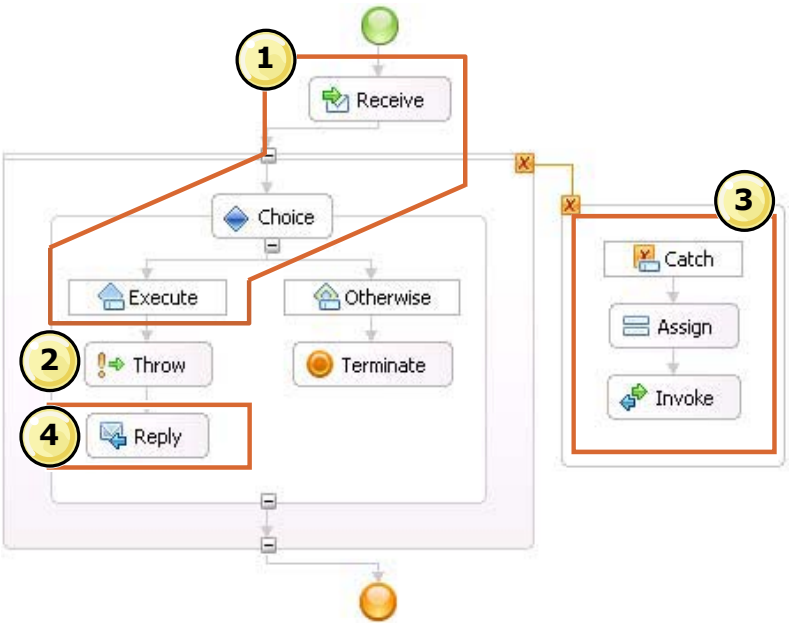
© Copyright IBM Corp. 2017

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

IBM Training

Fault handling and transactions

1. A transaction begins at the ***Receive*** activity
2. The ***Throw*** activity throws a fault, causing the current transaction to end
   - Since a fault handler is available, control passes to the handler and the current transaction is committed
3. The fault handler itself has its own transaction
4. With the fault successfully handled, execution resumes after the ***Invoke*** activity
   - The ***Reply*** activity has its own transaction



© Copyright IBM Corporation 2017

Figure 8-12. Fault handling and transactions

The flow of the process and the transaction boundaries is described in the slide, and is straightforward. A receive activity requires that you start a new transaction upon its execution. The choice activity has no effect on the current transaction at all; therefore, it runs in the same transaction as the one started for the receive activity. The throw activity ends the current transaction when it finishes.

The fault handler has its own transaction. In this scenario, the reply activity starts a new transaction since no transaction is available for it to join, and it requires that you commit the transaction after its execution.

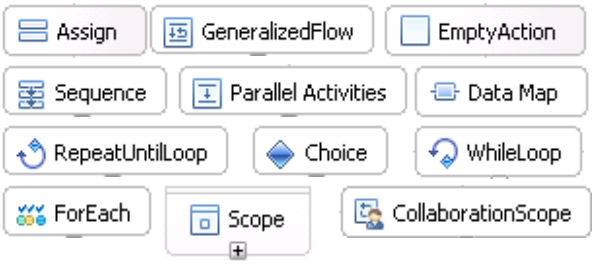
© Copyright IBM Corp. 2017

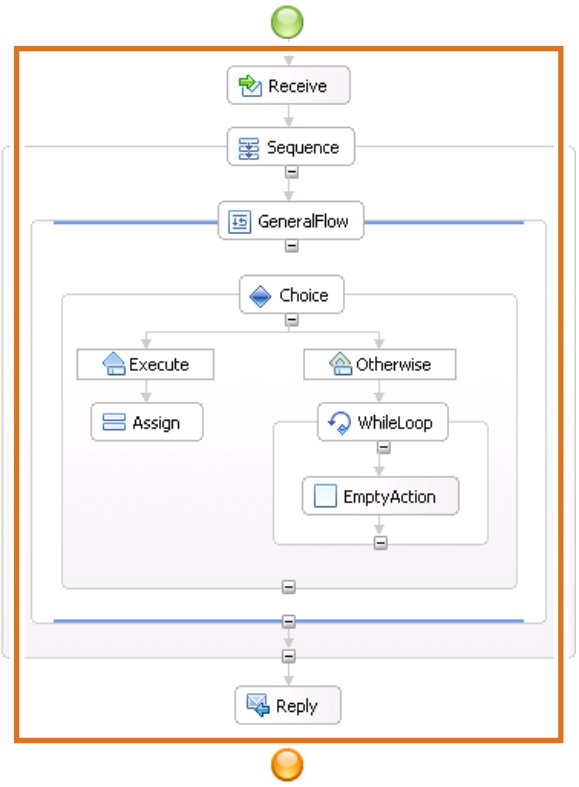
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Activities with no influence on transactions

- Since the *while*, *choice*, *sequence,* and *parallel* activities have no effect on the transaction boundaries, the entire process runs under one transaction
- With the process editor, you cannot add or change transaction attributes for these activities





© Copyright IBM Corporation 2017

Figure 8-13. Activities with no influence on transactions

The process that is shown does not do any work. It is only meant to show all of the activities that do not affect transaction boundaries.

© Copyright IBM Corp. 2017

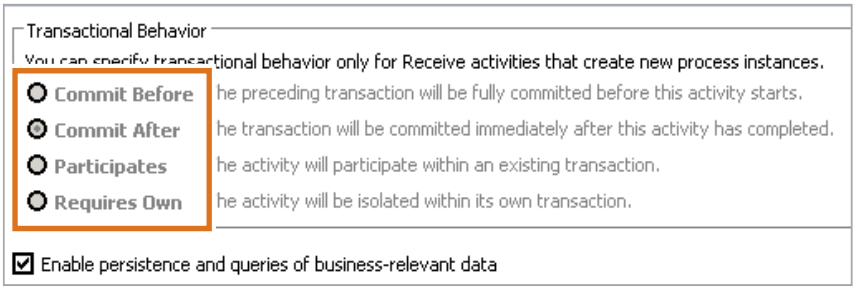
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Transactional behavior: Settings for activities

Certain activities have user-defined transaction attributes:

- **Commit before** promises that the current transaction ends, and a new transaction begins before starting the activity
- **Commit after** promises that the current transaction ends after the activity finishes
- **Participates** transaction behavior depends on the previous activities
- **Requires Own** is a combination of **commit before** and **commit after**
  - The container starts a new transaction, runs the activity, and then ends the transaction



© Copyright IBM Corporation 2017

Figure 8-14. Transactional behavior: Settings for activities

Transaction behavior settings are merely hints to the business process choreography engine. These hints can provide better performance in running business processes. The process engine might decide to use more transactions; however, it never uses fewer transactions than specified.

Remember that developers have control over how human task, invoke, and snippet activities affect the current transaction. The screen capture at the bottom of the slide shows the Properties view (Server tab) in IBM Integration Designer, where you specify the transaction behavior.

Use the *commit before* setting when you want the activity to affect the transaction just like receive, receive choice, and wait activities.

Use the *commit after* setting when you want the activity to affect the transaction just like terminate, throw, rethrow, compensate, and reply activities.

The *participates* option is more optimistic, and it joins the current transaction if any exists. If there is no transaction to join, it starts a new transaction. It does not commit the transaction after its execution unless an activity follows that requires you to commit the transaction before.

Requires own is the most expensive setting, which requires that you commit the running transaction before its execution, and commit it after it finishes.

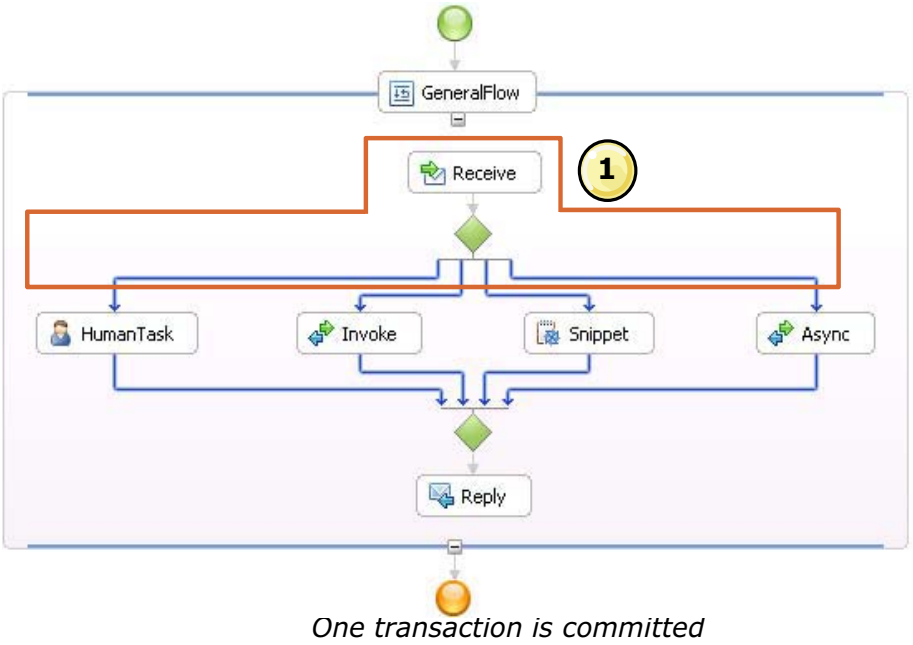
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit before (1 of 3)

1. The first transaction commits before any of the four activities begin



© Copyright IBM Corporation 2017

Figure 8-15. Transactional behavior: Commit before (1 of 3)

The next few slides demonstrate an example of the **commit before** option. When invoke, human task, and snippet activities are all set to commit the transaction before, the receive activity ends up having its own transaction.

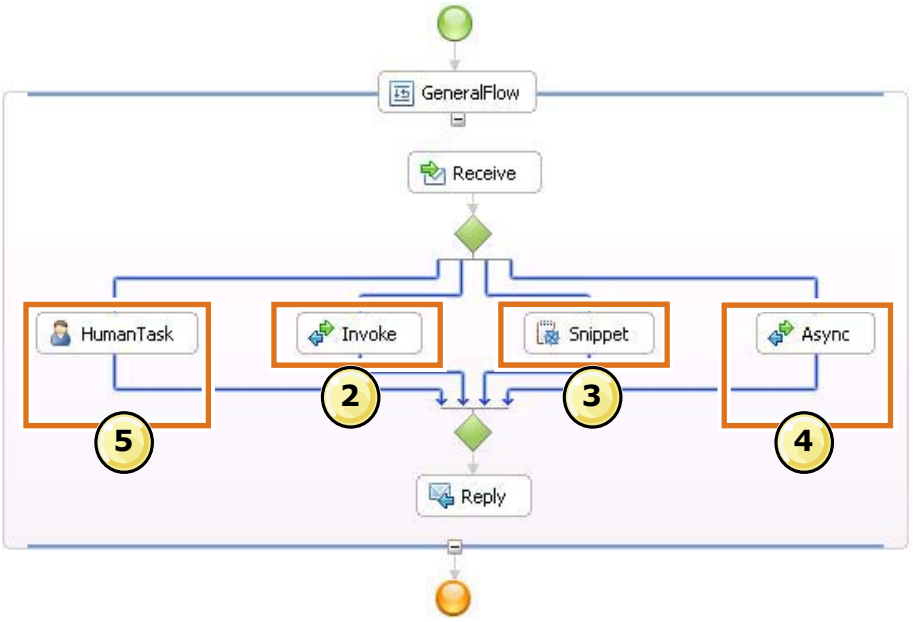
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit before (2 of 3)

2. The one-way ***Invoke*** activity completes and commits its transaction before the ***Reply***
3. The ***Snippet*** also commits its transaction
4. The request/response ***Invoke*** activity commits its transaction after sending the request message
5. The ***HumanTask*** also commits before the activity completes



Five transactions are committed

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-16. Transactional behavior: Commit before (2 of 3)

The asynchronous *invoke* activity sends and receives messages by using the Java Message Service. If the activity did not commit the transaction, the request message is not readable on the queue or topic. Therefore, the *invoke* activity must commit its transaction *after* sending the request but *before* receiving the response message. The *human task* has the same condition because the Human Task Manager service relies on JMS queues to send and receive messages.

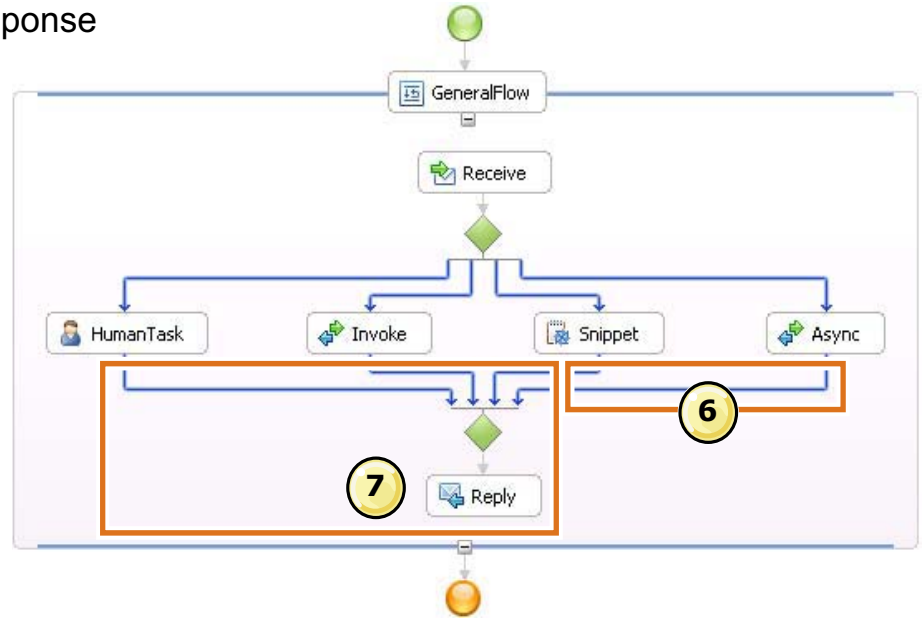
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit before (3 of 3)

6. The request/response ***Invoke*** finishes first and commits its transaction before the ***Reply***
7. The ***HumanTask*** completes
   - As the last inbound link to complete, the transaction continues until the ***Reply*** activity finishes



Seven transactions are committed

© Copyright IBM Corporation 2017

Figure 8-17. Transactional behavior: Commit before (3 of 3)

If the human task happens to be the last activity to finish, then the last reply activity joins its transaction since the commit before setting does *not* require that you commit the current transaction after it.

The reply activity can run the transaction of any of the four activities, depending on which one finishes last. Therefore, it is not possible to determine the exact transaction boundaries at design time.

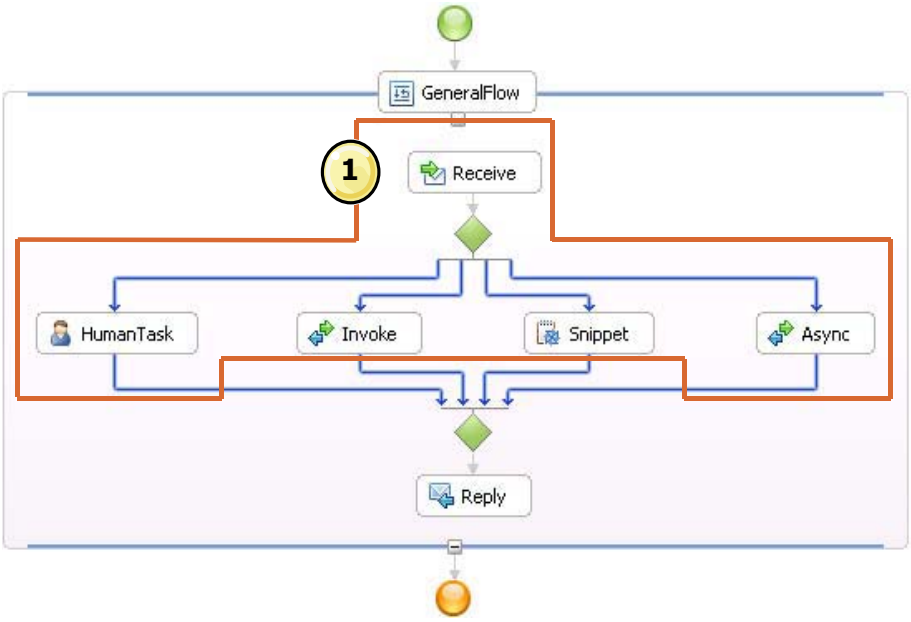
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit after (1 of 3)

1. When all four activities are set to *commit after*, most activities are included in the first transaction
   - The request/response ***Invoke*** and ***HumanTask*** activities mark the transaction boundary after sending their respective request messages



One transaction is committed

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-18. Transactional behavior: Commit after (1 of 3)

This diagram is an example where the business process choreography engine uses more transactions than specified in the process design. Although the second *invoke* activity labeled *async* is set to *commit after* behavior, the transaction must be committed after it sends a request and before it receives a response message. Otherwise, the service that is called does not see the request message.

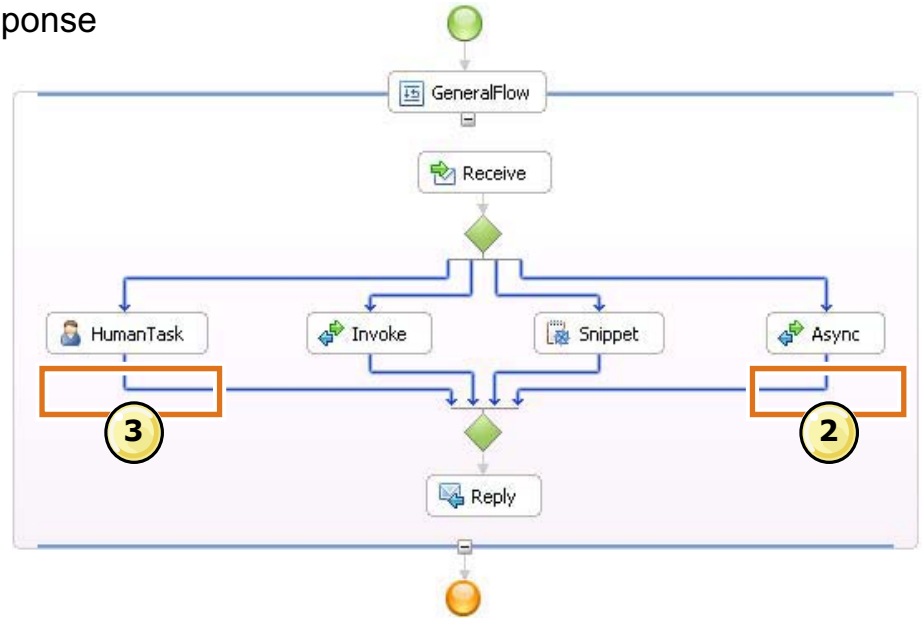
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit after (2 of 3)

2. The request/response ***Invoke*** activity commits the transaction after it sends the request message
   - The ***Invoke*** waits until it receives a reply
3. The ***HumanTask*** activity also commits the transaction after sending the request



Three transactions are committed

© Copyright IBM Corporation 2017

Figure 8-19. Transactional behavior: Commit after (2 of 3)

In this case, the reply portion of the asynchronous invoke and human task activities ends up having its own transaction.

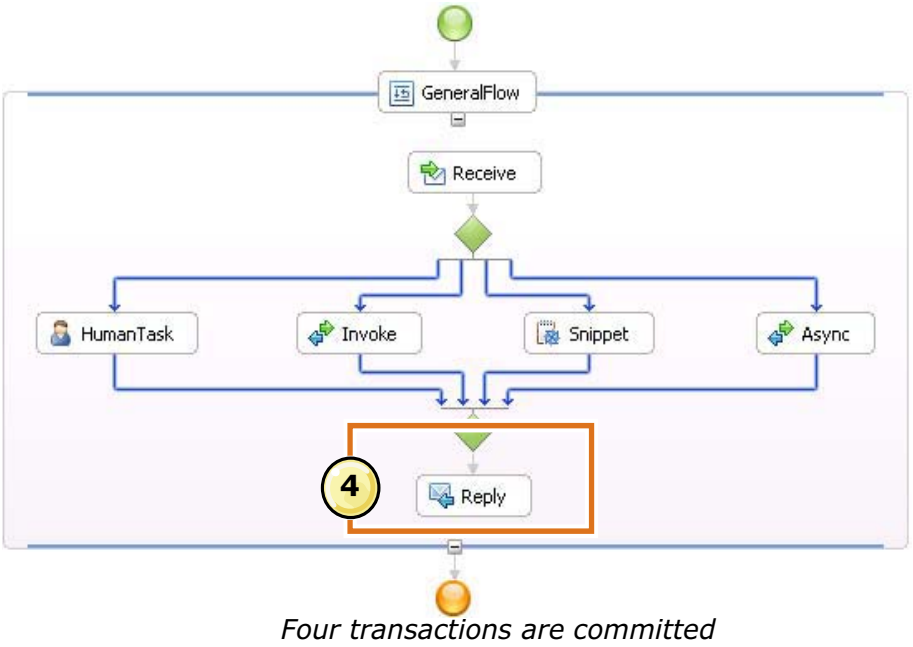
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Commit after (3 of 3)

4. Each of the five links after the activities participates in the same transaction
   - This final transaction includes the final ***Reply*** activity



© Copyright IBM Corporation 2017

Figure 8-20. Transactional behavior: Commit after (3 of 3)

The reply activity starts a new transaction and commits the transaction upon its execution since commit after is the transaction behavior of a reply activity.

Regardless of the activity that finishes last, the *reply* activity always completes in its own transaction.

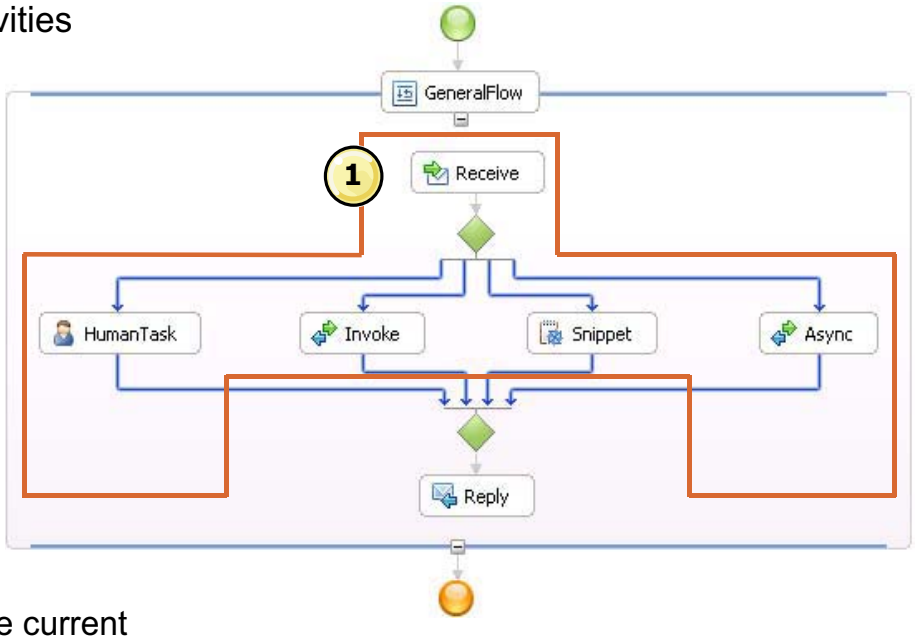
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Participates (1 of 2)

1. With all five activities set to *participates*, the one-way ***Invoke*** and ***Snippet*** activities all complete within the same transaction
   - The asynchronous ***Invoke*** and ***HumanTask*** activities end the current transaction after they send their respective request messages



One transaction is committed

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-21. Transactional behavior: Participates (1 of 2)

The *participates* option is more optimistic behavior in that it joins a running transaction if any exists. However, only the request message delivery of the asynchronous invoke and human task activities joins the current transaction (a human task activity is an asynchronous request/response execution).

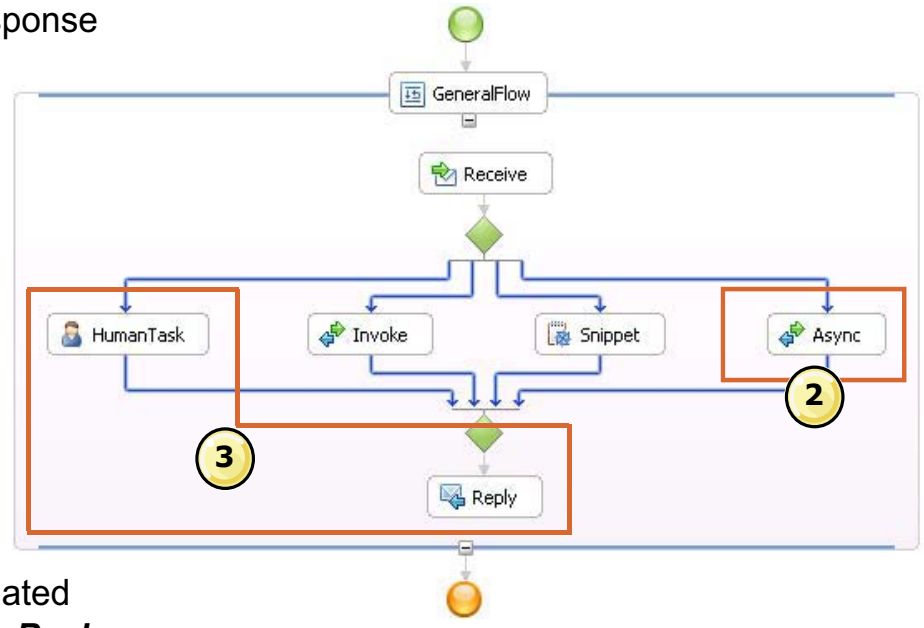
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Participates (2 of 2)

2. The request/response ***Invoke*** activity finishes first, committing its transaction after calculating the link transition condition
3. The reply activity starts only when the last link is evaluated
   - In this case, the ***Reply*** becomes part of the transaction started at the ***HumanTask*** activity



Three transactions are committed

© Copyright IBM Corporation 2017

Figure 8-22. Transactional behavior: Participates (2 of 2)

Although the *participates* transaction behavior attempts to include all activities in the same transaction, it is not guaranteed. In this business process, the asynchronous *invoke* and the *human task* require no less than two transactions to complete.

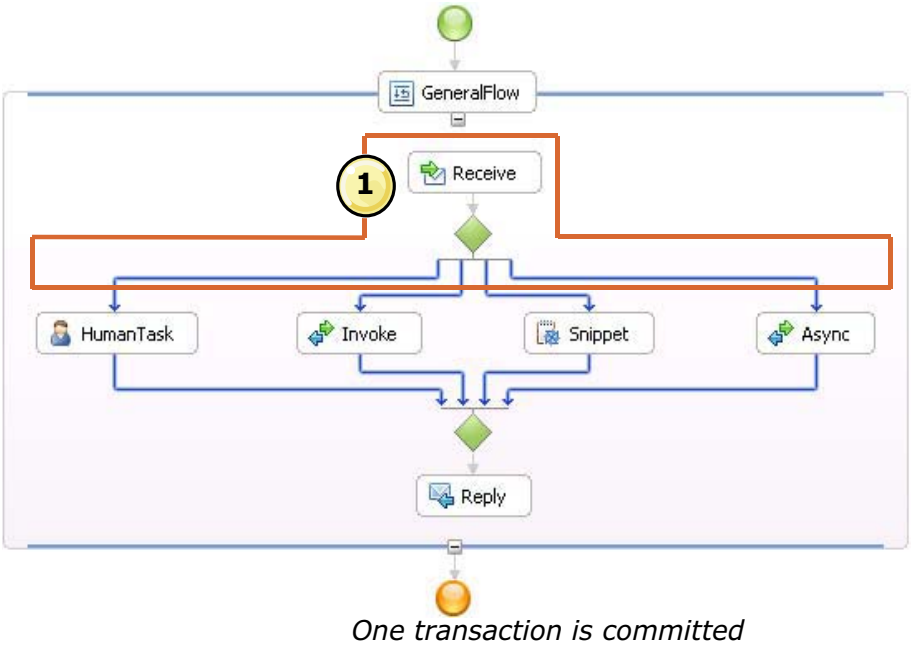
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Requires own (1 of 4)

1. In the beginning, the *requires own* behavior acts in a similar manner to the *commit before* behavior
   - Before any of the five activities begin, the first transaction ends and commits



© Copyright IBM Corporation 2017

Figure 8-23. Transactional behavior: Requires own (1 of 4)

The *requires own* behavior forces the current transaction to commit before *and* after.

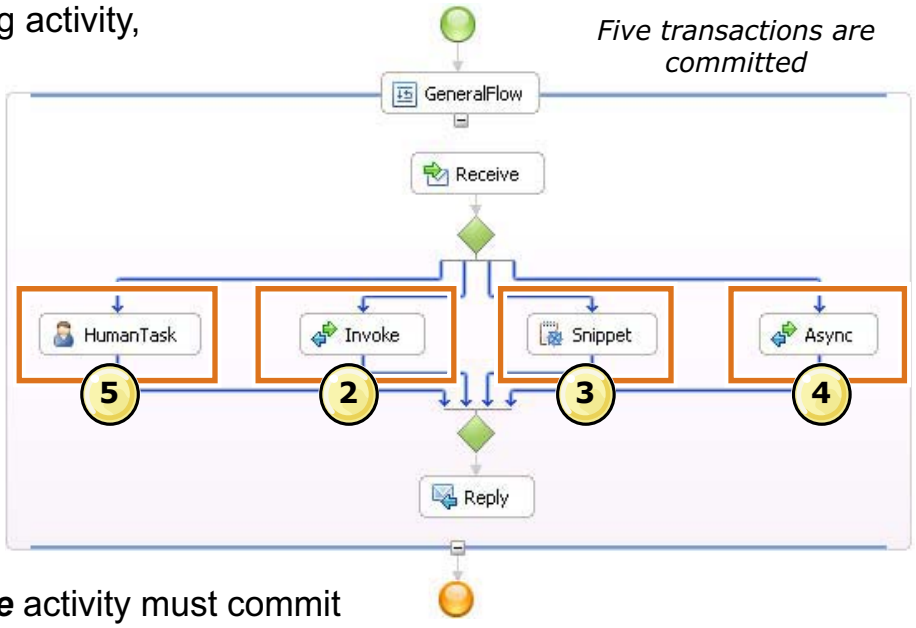
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Requires own (2 of 4)

2. As a nonblocking activity, the engine creates a transaction, runs the ***Invoke*** activity, and ends the transaction
3. The ***Snippet*** activity runs in the same manner
4. The request/response ***Invoke*** activity must commit its transaction after sending the request but before receiving a response
5. The ***HumanTask*** has the same transaction behavior as the request/response ***Invoke*** activity



© Copyright IBM Corporation 2017

Figure 8-24. Transactional behavior: Requires own (2 of 4)

The non-blocking invoke activity and the snippet activity run in their own transaction, while asynchronous request and human task creation commit after.

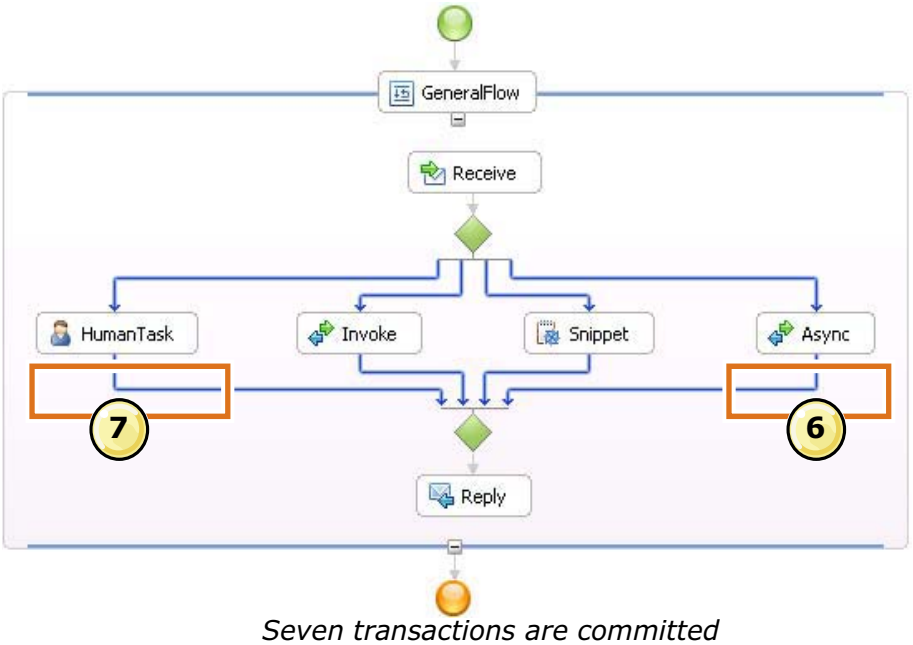
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Requires own (3 of 4)

6. Upon receiving a response message, the request/response ***Invoke*** activity commits its transactions
7. The ***HumanTask*** activity also commits its transaction after receiving a response message



Seven transactions are committed

© Copyright IBM Corporation 2017

Figure 8-25. Transactional behavior: Requires own (3 of 4)

After the asynchronous request of the invoke activity comes back, the transaction gets committed, and the same holds for human tasks.

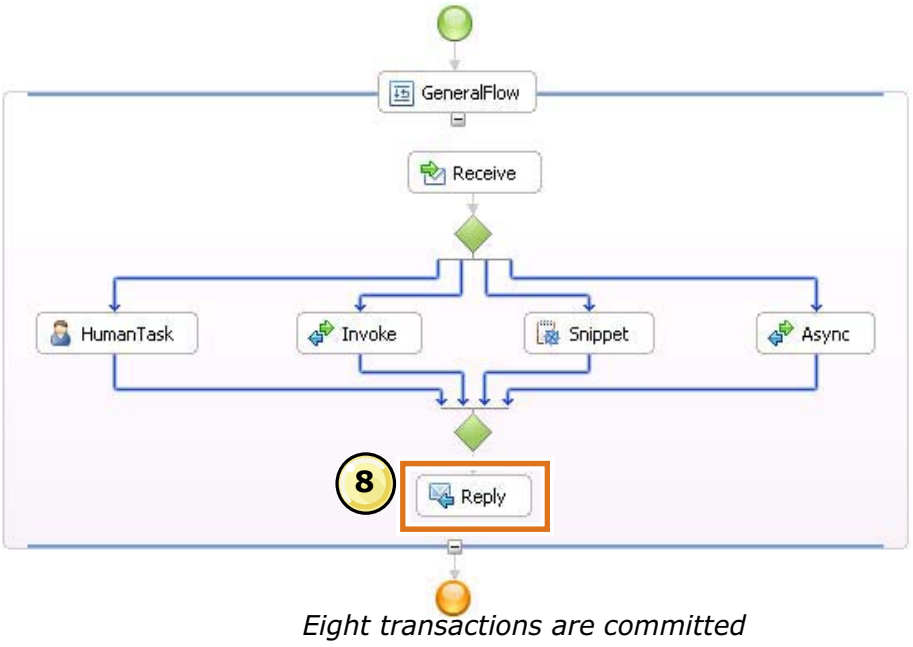
© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior: Requires own (4 of 4)

8. Evaluating each of the five links and running the final ***Reply*** activity occurs in one single transaction
   - In total, there are eight transactions, as opposed to three transactions with the *participates* transaction behavior



© Copyright IBM Corporation 2017

Figure 8-26. Transactional behavior: Requires own (4 of 4)

As a result, the reply activity also ends up running in its own transaction.

© Copyright IBM Corp. 2017

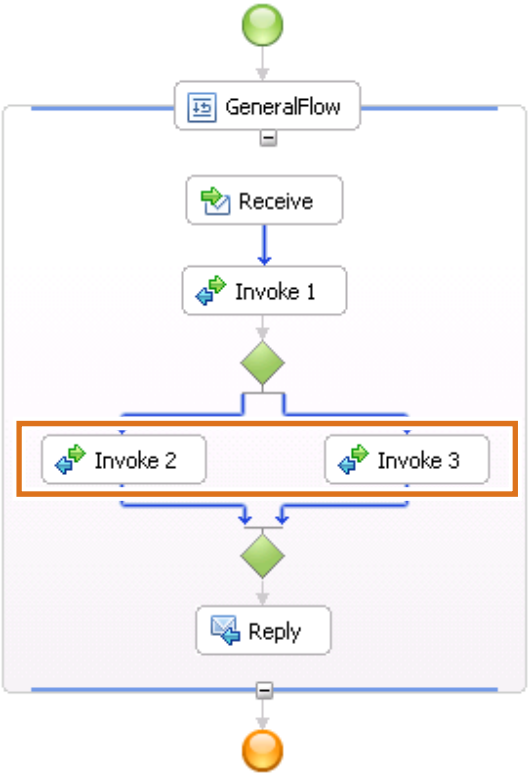
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Transactions affecting parallel execution

- How do transaction boundaries affect parallel execution and activity execution order?
- The point at which a transaction commits affects the subsequent activity execution order
  - The *commit before* and *requires own* behavior committed the transaction before the path split
  - The *commit after* behavior committed the transaction after the activity, before the paths merged

If the transaction behavior of Invoke 1 and Invoke 2 are both set to *participate*, and commit after Invoke 3, these activities do not get run in parallel



© Copyright IBM Corporation 2017

Figure 8-27. Transactions affecting parallel execution

By now, you know that a long-running business process can have parallel execution of activities. To force parallel executions of paths, you might have to set the transaction behavior in such a way that the current transaction commits first. The next few slides explain how to configure the transaction behavior so that more than one activity can be run in a different thread.

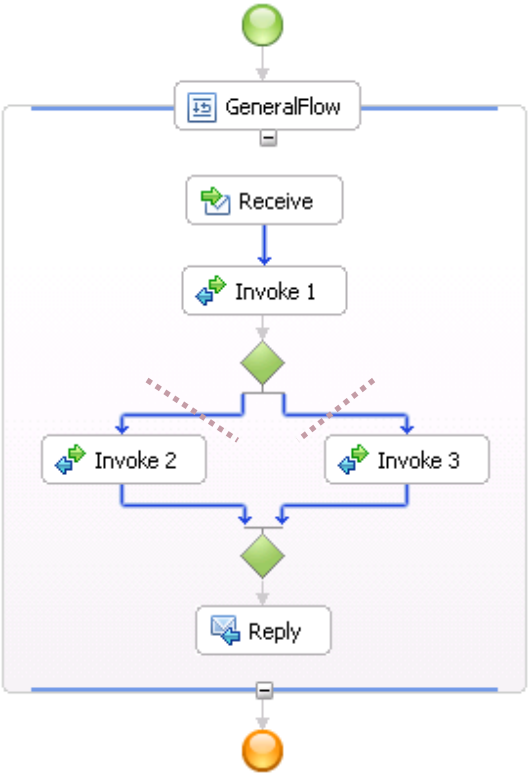
© Copyright IBM Corp. 2017

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

IBM Training

Long-running processes and parallelism

- To force separate threads for each execution path, a new transaction for n – 1 paths is required
  - A transaction commit must occur at the bars (see diagram)
- Four options are available:
  - Set Invoke 1 to *requires own* or *commit after*
  - Set Invoke 2 to *requires own* or *commit before*
  - Set Invoke 3 to *requires own* or *commit before*
  - Set both Invoke 2 and Invoke 3 to *requires own* or *commit before*



Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-28. Long-running processes and parallelism

There must be at least *n-1* transactions for *n* concurrent execution paths because one path can continue to use the transaction before the parallel split.

Remember that the invoke activity is one of the activities that allow developers to decide how they can affect the current transaction. To ensure that Invoke 2 and Invoke 3 run in a separate thread, you might choose to do one of the four options that are listed here.

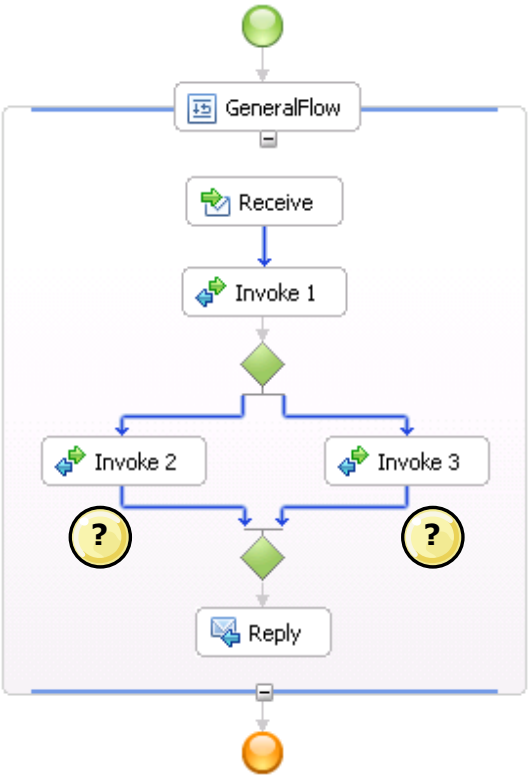
© Copyright IBM Corp. 2017

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

IBM Training

Microflows and parallelism

- Since all microflows run under one thread, parallel execution is not possible
  - After the split from *Invoke1*, *Invoke2,* and *Invoke3,* run one after another in an undetermined order
- Execution order of activities in parallel paths cannot be accurately predicted or guaranteed



© Copyright IBM Corporation 2017

Figure 8-29. Microflows and parallelism

The previous slide showed ways to split transaction boundaries to ensure parallelism. The execution order of parallel paths cannot be accurately predicted or guaranteed at development time.

Remember that the activities within parallel activities (flow) structure has no ordering constraint between two activities at the same level.

© Copyright IBM Corp. 2017

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

IBM Training

Uncertain transaction boundaries with participates transaction behavior

The following transaction behaviors tell when the transaction is going to commit:

- **Commit before** always commits the current transaction before starting the activity
- **Commit after** always commits the current transaction after finishing the activity
- **Requires own** always commits the current transaction, creates a new transaction, finishes an activity, and then commits the second transaction

However, the behavior for *participates* is not guaranteed

- *Participates* typically reduces parallel execution paths
- *Participates* does not commit the current transaction before starting the activity
- If a fault occurs in the business process, *participates* attempts to rerun the activities that are participating in the transaction
  - *Participates* might choose to commit the transaction before an activity

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-30. Uncertain transaction boundaries with participates transaction behavior

The business process engine might overwrite the transaction behavior that the developer sets.

Commit before, *commit after*, and *requires own* transaction behaviors guarantee when the transaction commits, if it completes successfully. However, the transaction behavior settings alone do not determine the total number of transactions within a long-running process. As shown in the previous slide, certain types of invocation styles cause an invoke operation to commit its transaction. Human tasks also cause a transaction to commit before the activity completes.

© Copyright IBM Corp. 2017

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

8.2. Transactional behavior during failure

© Copyright IBM Corp. 2017

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

IBM Training

Transactional behavior during failure

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

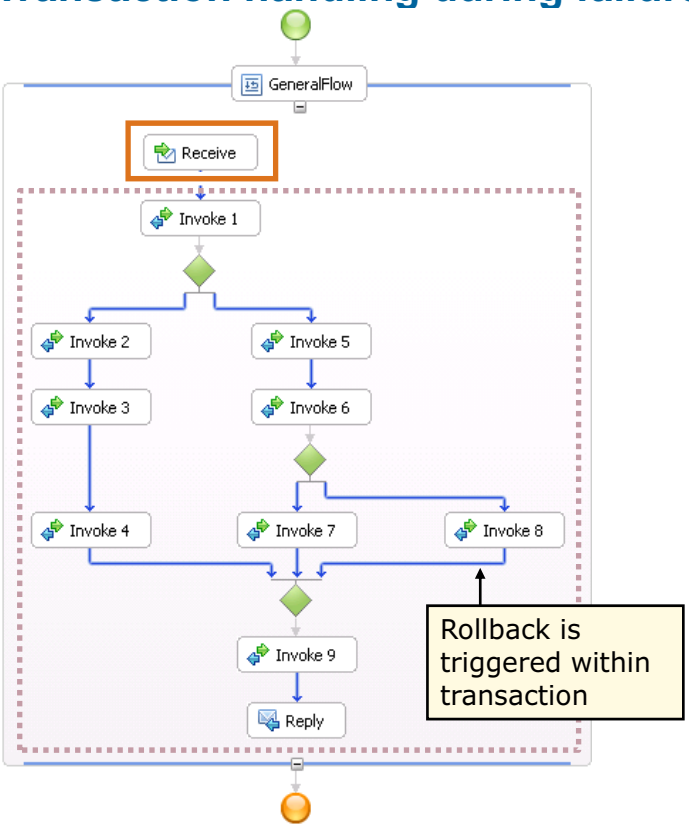
Figure 8-31. Transactional behavior during failure

© Copyright IBM Corp. 2017

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

IBM Training

Transaction handling during failure (1 of 5)



- All invoke activities (except the first) set to *participates*
- The first invoke activity is set to *commit before*
- Invoke activities extend transaction boundary to service
- Those services that are started by using a transaction-aware binding participate
- ***No*** internal message is sent

© Copyright IBM Corporation 2017

Figure 8-32. Transaction handling during failure (1 of 5)

Rectangles with solid edges represent committed transactions. Dotted edge rectangles represent the transaction that the business process engine is running.

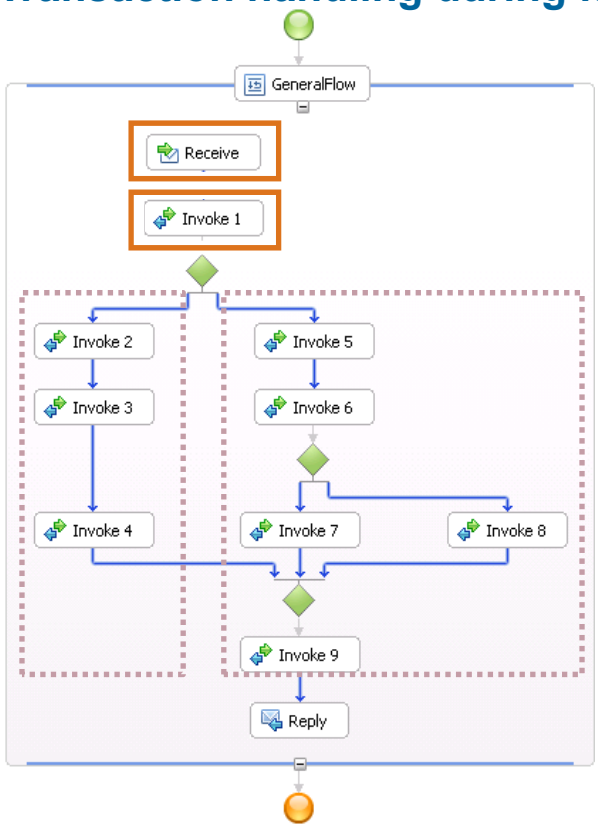
In this scenario, the first invoke activity (Invoke 1) is set to **commit before** as its transaction behavior, and the rest of the invoke activities are set to **participates**. These settings cause the receive activity to run in its own transaction, and since the reply activity **commits after**, it joins the existing transaction. The prediction during the development time is to have two transactions in such processes. Notice that there are no parallel paths.

© Copyright IBM Corp. 2017

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

IBM Training

Transaction handling during failure (2 of 5)



- All invoke activities (except the first) set to *participates*
- The first invoke activity is set to *commit before*
- Invoke activities extend transaction boundary to service
- Those services that are started by using a transaction-aware binding participate
- ***Three*** internal messages sent

© Copyright IBM Corporation 2017

Figure 8-33. Transaction handling during failure (2 of 5)

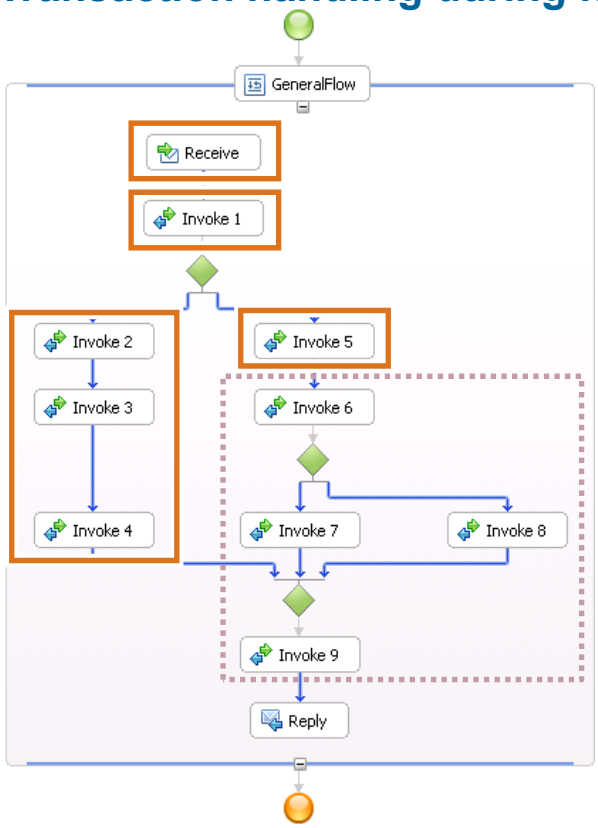
Because all of the invoke activities are running in the same transaction, it fails as a whole (atomicity) and rolls back. The Invoke 1 activity is run successfully, so the engine commits the transaction after Invoke 1, which results in splits of the transaction.

© Copyright IBM Corp. 2017

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

IBM Training

Transaction handling during failure (3 of 5)



- All invoke activities (except the first) set to *participates*
- The first invoke activity is set to *commit before*
- Invoke activities extend transaction boundary to service
- Those services that are started by using a transaction-aware binding participate
- ***Four*** internal messages sent

© Copyright IBM Corporation 2017

Figure 8-34. Transaction handling during failure (3 of 5)

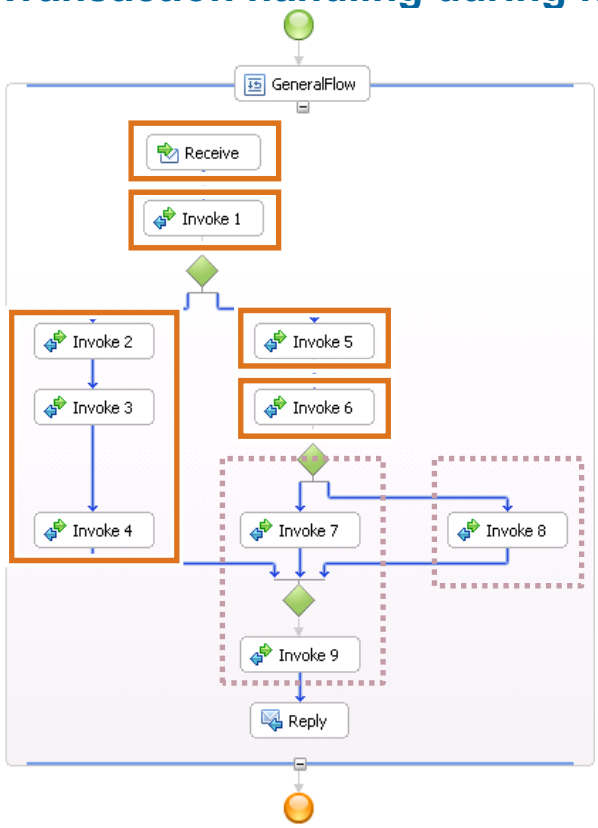
Invoke 6, Invoke 7, Invoke 8, or Invoke 9 did not cause the failure. Since the transaction fails as a whole, it rolls back as a whole. Similar to what you saw previously, the transaction gets committed after the successful execution of Invoke 5.

© Copyright IBM Corp. 2017

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

IBM Training

Transaction handling during failure (4 of 5)



- All invoke activities (except the first) set to *participates*
- The first invoke activity is set to *commit before*
- Invoke activities extend transaction boundary to service
- Those services that are started by using a transaction-aware binding participate
- ***Six*** internal messages sent

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-35. Transaction handling during failure (4 of 5)

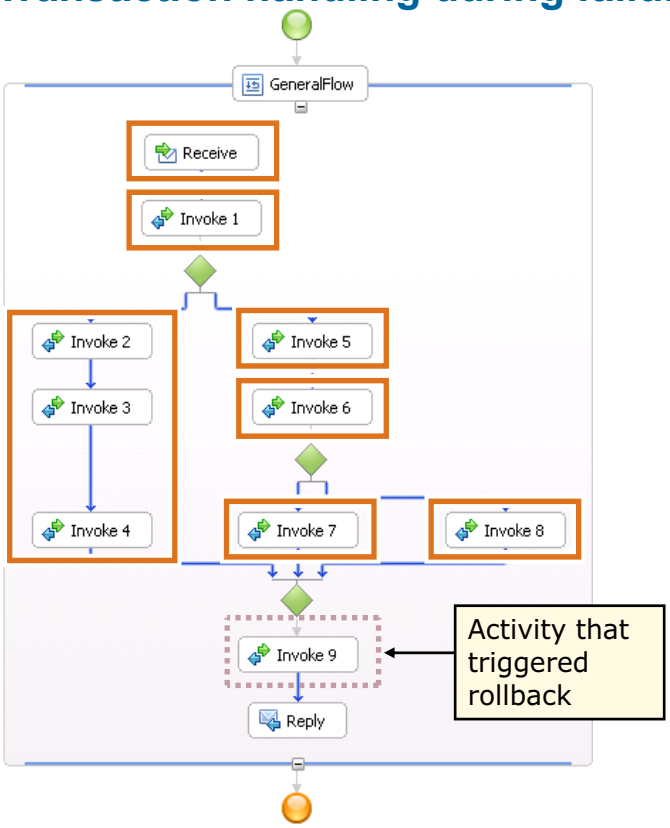
The transaction commits after the successful execution of Invoke 6, and you have parallel execution of Invoke 7 and Invoke 8.

© Copyright IBM Corp. 2017

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

IBM Training

Transaction handling during failure (5 of 5)



- All invoke activities (except the first) set to *participates*
- The first invoke activity is set to *commit before*
- Invoke activities extend transaction boundary to service
- Those services that are started by using a transaction-aware binding participate
- ***Seven*** internal messages sent

© Copyright IBM Corporation 2017

Figure 8-36. Transaction handling during failure (5 of 5)

After sending seven internal messages, the business process engine isolated the activity in which Invoke 9 was responsible for the rollback. This activity is now run three times. After that, a human being must do manual interventions to complete this activity.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe the general properties of transactions
- Explain the purpose of the transaction quality of service qualifiers
- Categorize activities according to their effect on transactions
- Describe the default transactional behavior of various types of activities
- Describe the implied behavior of activities that choose flexible transactional properties
- Describe the transactional behavior of a business process during a failure

© Copyright IBM Corporation 2017

Figure 8-37. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: Transaction boundaries can affect the runtime performance of a business process.
2. True or False: Service bindings can affect transaction behavior.
3. True or False: The “participates” transaction setting on flexible activities uses the least amount of processing for a business process.
4. True or False: Reducing the number of transactions within a long-running process always increases performance.

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-38. Review questions

Write your answers here:

1.

2.

3.

4.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: Transaction boundaries can affect the runtime performance of a business process.
   The answer is True.
2. True or False: Service bindings can affect transaction behavior.
   The answer is True.
3. True or False: The “participates” transaction setting on flexible activities uses the least amount of processing for a business process.
   The answer is True.
4. True or False: Reducing the number of transactions within a long-running process always increase performance.
   The answer is False. The transaction behavior settings alone do not determine the total number of transactions within a long-running process and its performance. Certain types of invocation styles also play a role.

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-39. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Defining transactional behavior in SCA applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 8-40. Exercise 7

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

IBM Training

Exercise objectives

- View transaction propagation settings for a composite application by using an integration solution diagram
- View transaction propagation settings for an individual module by using the assembly diagram
- Examine the transaction boundaries for activities in long-running business processes
- Use a scope in a long-running business process to manage persistence

Transactional behavior of SCA applications

© Copyright IBM Corporation 2016

Figure 8-41. Exercise objectives

© Copyright IBM Corp. 2017

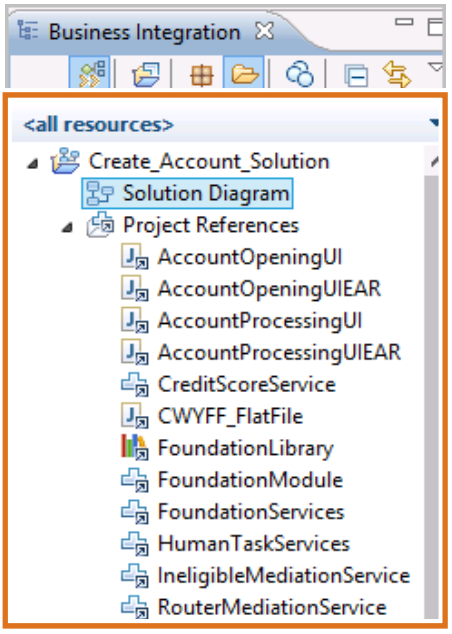
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 7

Prebuilt components that are imported in the lab:

1. **Create_Account_Solution** solution diagram



New components that you create in the lab:

None

Transactional behavior of SCA applications

© Copyright IBM Corporation 2017

Figure 8-42. Components that are required for Exercise 7

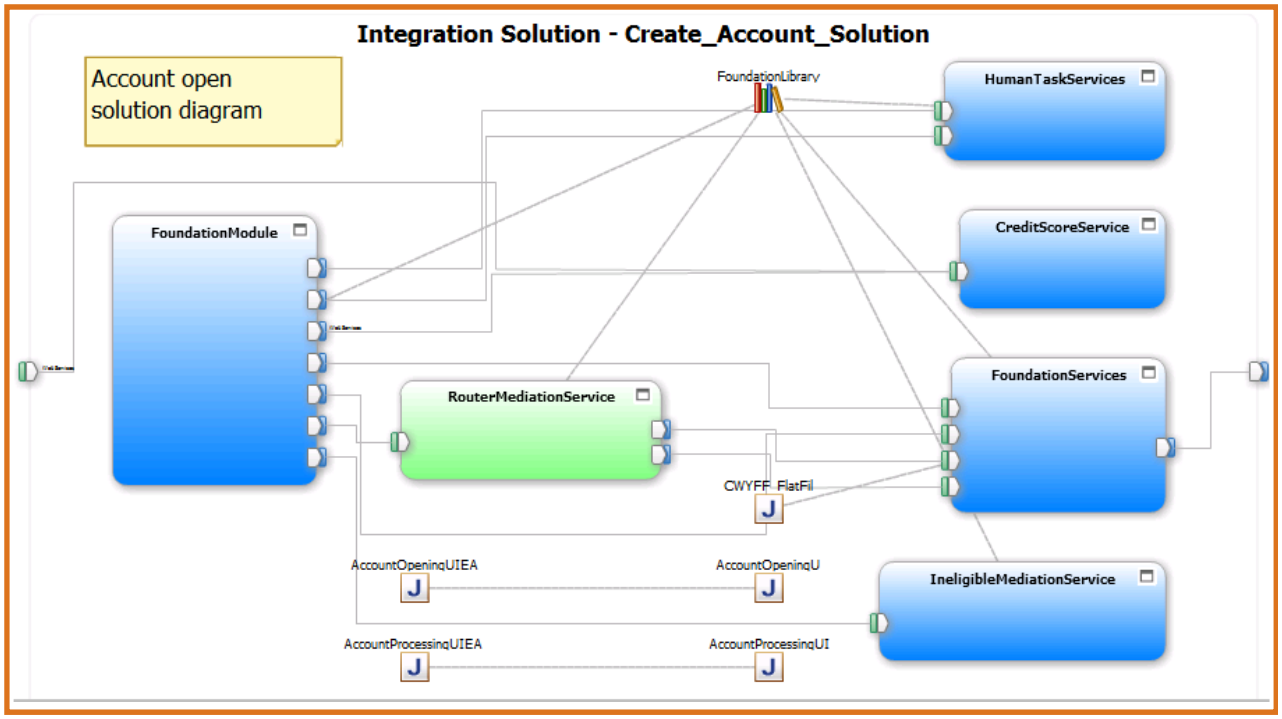
You view transaction settings in applications developed in IBM Integration Designer in several ways. You can view transaction propagation settings in the assembly editor and in integration solution diagrams. In this portion of the exercise, you examine transaction settings in the account verification lab scenario.

© Copyright IBM Corp. 2017

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

IBM Training

Explore Create Account Solution diagram in Exercise 7



© Copyright IBM Corporation 2017

Figure 8-43. Explore Create Account Solution diagram in Exercise 7

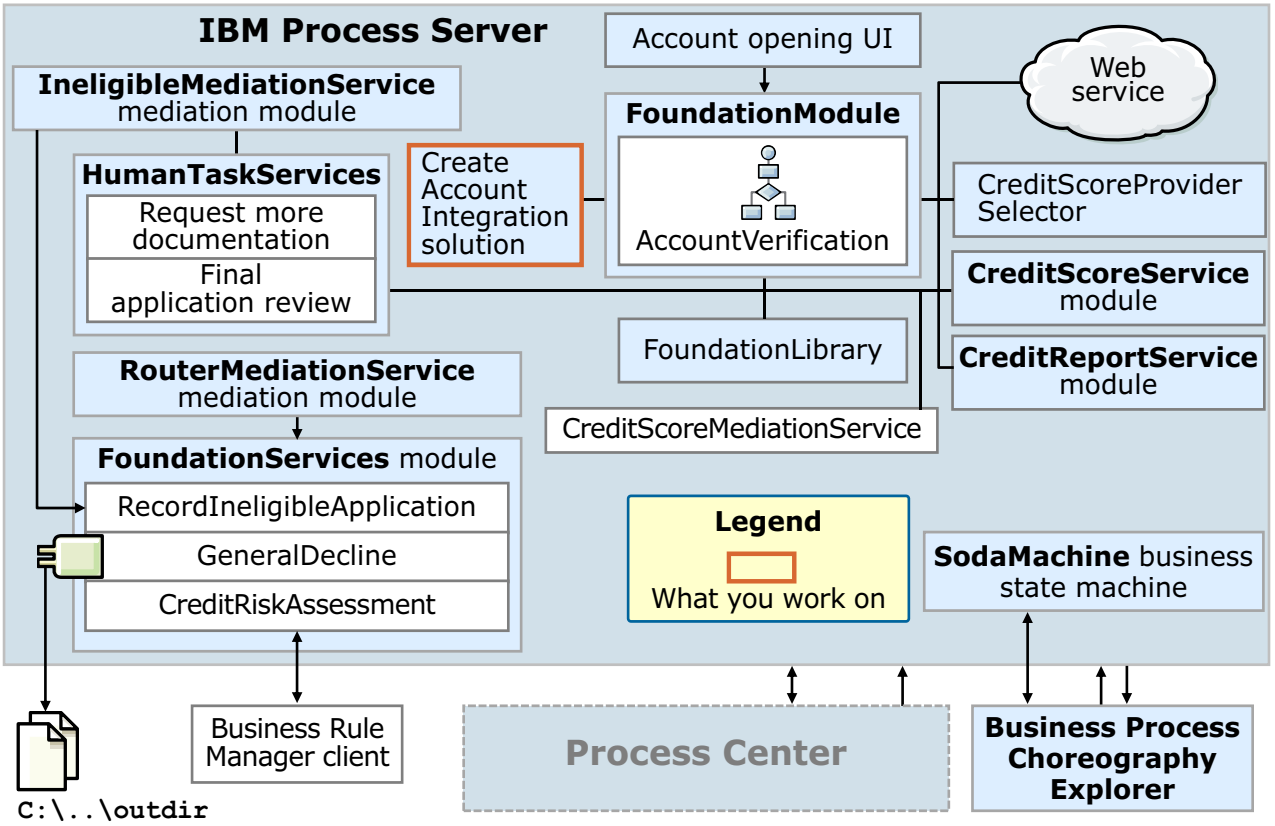
In the integration solution editor, you view and highlight the transaction path between components in a single module or between the components in all of the modules that make up your integration solution. You also show and highlight the transaction path from a specific component to the other components with which it interacts.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 7



Transactional behavior of SCA applications © Copyright IBM Corporation 2017

Figure 8-44. Course business scenario: What you work with in Ex 7

In this lab, you work with IBM Integration Designer.

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

Unit 9. Business process flexibility

Estimated time

01:00

Overview

This unit describes many of the IBM Process Server features that provide for flexible business processes, including selector components, dynamic routing, and process repair.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Identify the business process components that provide application flexibility and dynamicity
- Explain the purpose and business value of selectors
- Describe the runtime behavior of selectors
- Provide a high-level description of the runtime architecture of content-based routing
- Identify the BPEL components that allow dynamic process navigation at run time
- Describe the runtime components that support the dynamic repair of process instances

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Selectors
- Dynamic routing
- Repair process navigation

© Copyright IBM Corporation 2017

Figure 9-2. Topics

© Copyright IBM Corp. 2017

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

9.1. Selectors

© Copyright IBM Corp. 2017

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

IBM Training

Selectors

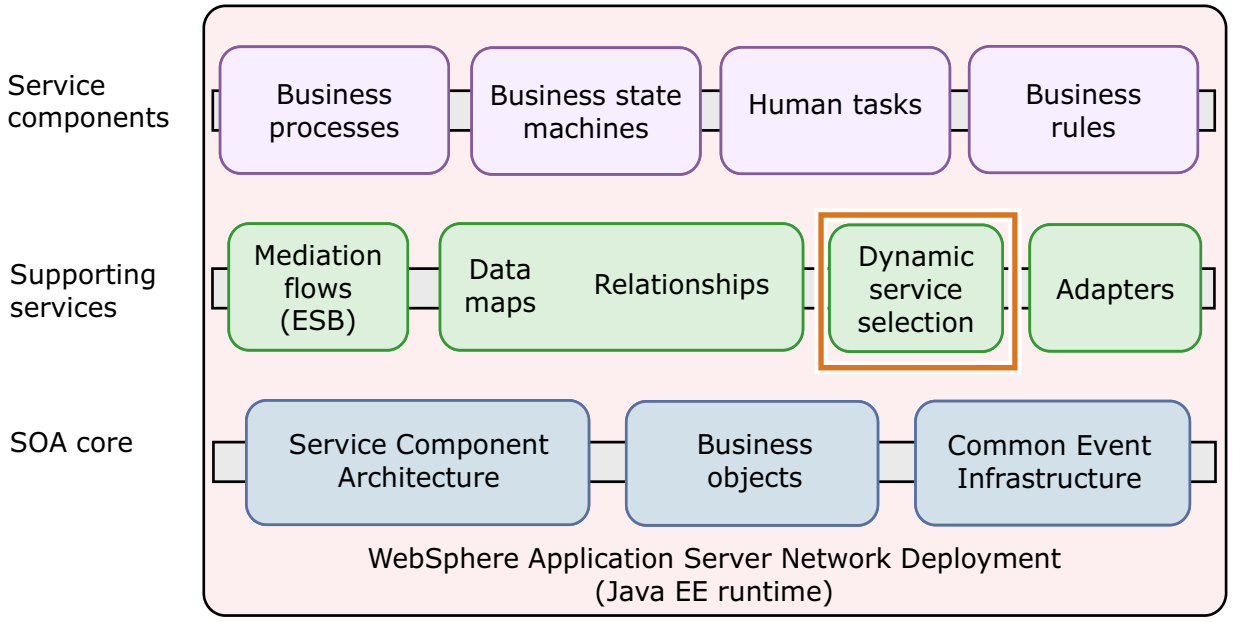
© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-3. Selectors

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

IBM Training

Selectors are a supporting service component



Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-4. Selectors are a supporting service component

Dynamic service selection is a supporting service component. This category includes all facilities that allow a service to be determined and started at run time. Included in this service category are selectors, which dynamically select services that are based on a runtime condition, such as holidays versus regular business days.

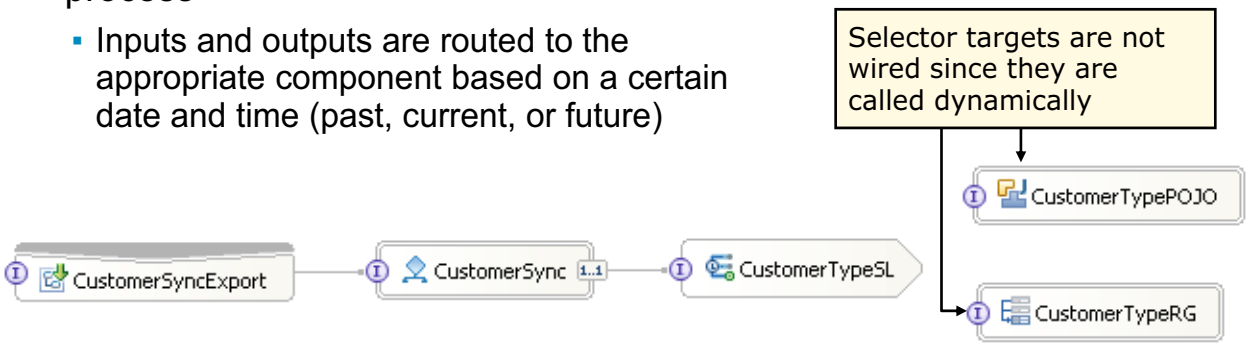
© Copyright IBM Corp. 2017

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

IBM Training

Overview of selectors

- A selector is an SCA component that is used to dynamically invoke one of several other SCA components by a set criteria
  - The target components all have matching interfaces, but might have different implementations
  - The criteria that is used to choose the target is a date and time range
  - Criteria can also be based on the content of the business object input, but this type of choice must be created programmatically
- Selectors behave like routers by changing the path of a business process
  - Inputs and outputs are routed to the appropriate component based on a certain date and time (past, current, or future)



Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-5. Overview of selectors

The main job of the selector is to dynamically determine which implementation to start based on some defined set of criteria (currently, based on a date and time). Think of it as a router, routing inputs and outputs to the appropriate target component based on a certain date (past, current, or future).

© Copyright IBM Corp. 2017

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

IBM Training

Selector examples

- A selector can be used to choose between rule groups:
  - One applies rental car rates for off-season dates
  - One applies rental car rates for peak dates
- A selector can be used to silently update a user interface when the rollout date occurs
  - User interface remains consistent while the hidden, back-end functionality changes
- A selector can be used to replace one web service invocation with another
  - You decide to switch to a different credit score provider
  - You replace one credit card processor with another
  - The service provider you were using is no longer in business

© Copyright IBM Corporation 2017

Figure 9-6. Selector examples

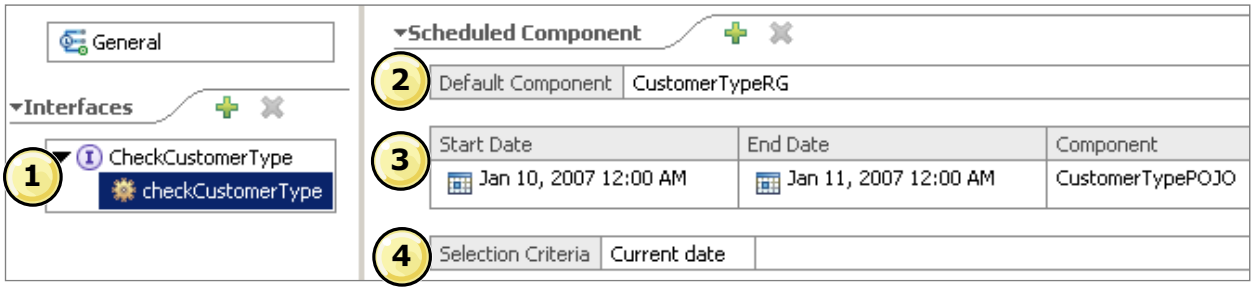
© Copyright IBM Corp. 2017

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

IBM Training

Selector components

- Selectors are composed of:
  1. An interface
  2. A default SCA component (guarantees a target)
  3. One or more target destinations corresponding to start and end date/time ranges (dates cannot overlap)
  4. A date selection criteria (value is compared to the target start-end dates)



Business process flexibility © Copyright IBM Corporation 2017

Figure 9-7. Selector components

It is important to note that the interface of the selector and the target implementations must match. If the interfaces of the selector targets differ, you can use an interface map and make the map the selector target.

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

IBM Training

Business value of selectors

- Selectors fit neatly into SOA
  - A selector can use any SCA component as its target, including an export component in another module
  - You can use this feature to decouple your client application from the specific implementation of the target
  - If the implementation of the target changes, your client application is not affected
- The choice of SCA component targets, and the date and time criteria, can be modified at run time
  - If a target service is down, changes its interface, or is rendered obsolete, you can apply a new SCA target implementation without restarting your application or rebooting the server
  - You do not need to redeploy the module

Business process flexibility

© Copyright IBM Corporation 2017

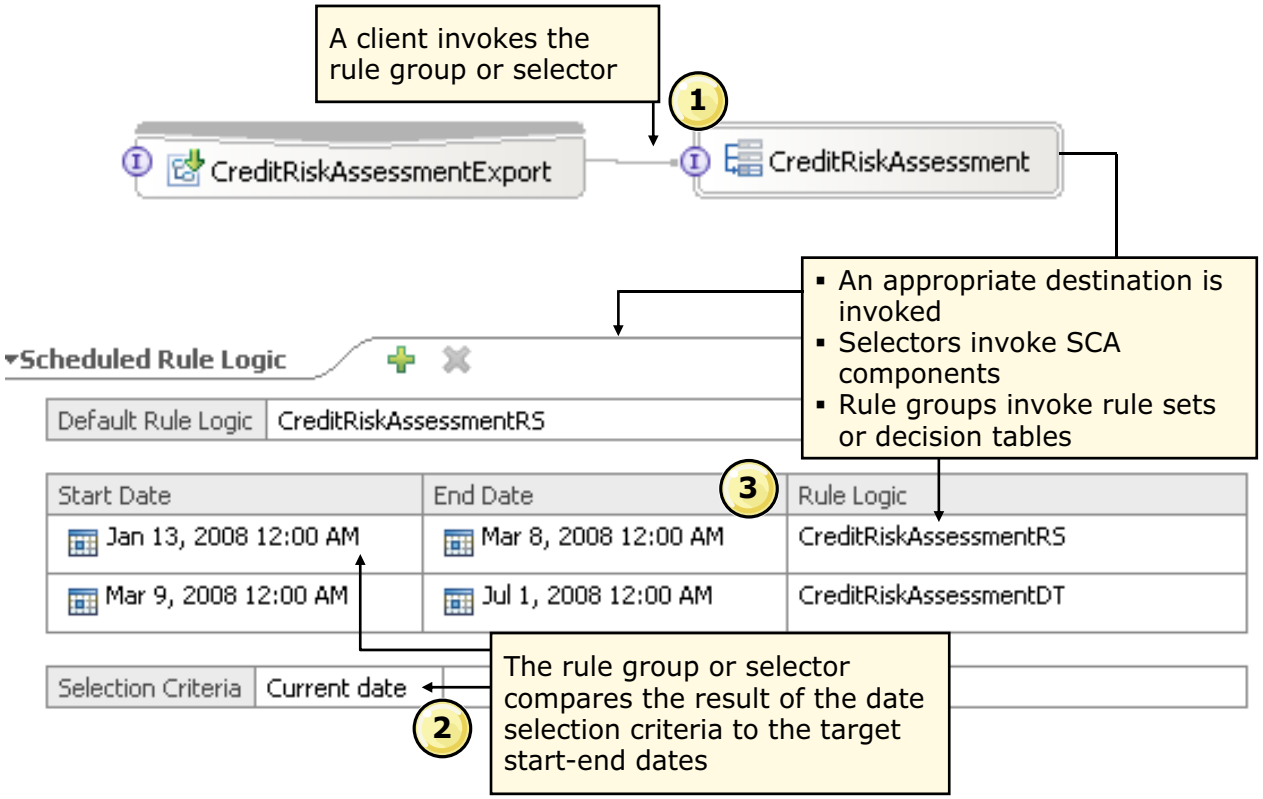
Figure 9-8. Business value of selectors

© Copyright IBM Corp. 2017

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

IBM Training

Selectors at run time



Business process flexibility © Copyright IBM Corporation 2017

Figure 9-9. Selectors at run time

Rule groups and selectors function at run time as follows:

1. A client (JSP, POJO, SCA component) calls the rule group or selector component interface.
2. The date that is returned from the date selection criteria is compared to the start and end date criteria of the targets.
   - The date can be “mined” from an attribute in the incoming business object by using XPath or a Java snippet.
   - You can choose to use the current system date.
   - You can use a date that is returned from a Java expression.
3. Based on a comparison of the date selection criteria and the target start and end dates, an appropriate target is started.
   - Rule group targets are rule sets and decision tables.
   - Selector targets are SCA components (in the same module or in other SCA modules).
   - Selectors follow the SCA programming model so that any SCA client can be the client of a selector.

© Copyright IBM Corp. 2017

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

- If the date criterion falls outside the start and end dates for the rule sets and decision tables, the default destination is used.

Every selector needs an interface to operate. You cannot create a selector in the tool without an interface. What does the interface really do? The interface provides the inputs and outputs that the selector passes along to the target implementation. When a client calls the selector, it already knows what interface the selector is going to use and passes the appropriate inputs and outputs.

On the right side of the slide, you see target implementations. With selectors, you have the freedom to select any SCA component you want, whether it is a POJO, a business rule, BPEL, or even another selector.

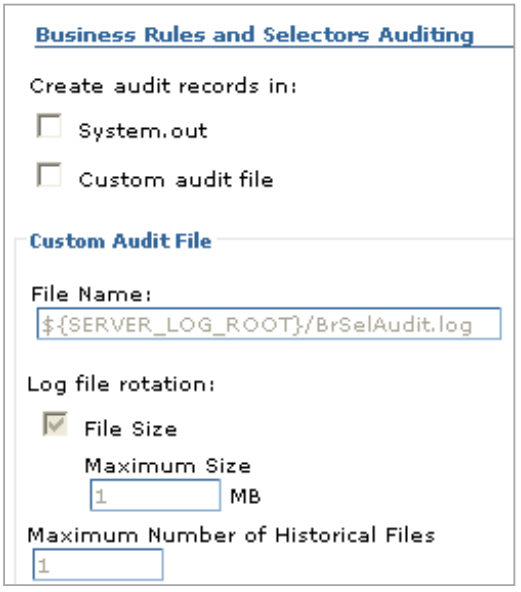
© Copyright IBM Corp. 2017

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

IBM Training

Selector auditing

- When changes occur to selectors at run time, the changes can be audited
- You can write changes to the **`SystemOut.log`** file or a custom audit log
- Auditing records:
  - Name of the person who is modifying
  - Location from where the change request originated
  - Old selector object
  - New selector object
- Audit logging for selectors can also be configured by using the command line



© Copyright IBM Corporation 2017

Figure 9-10. Selector auditing

The business rule objects are the complete business rule set, decision table, or business rule group for the business rule that is replaced and the new version that replaced it. You see the logs (the audit output cannot be directed to the Common Event Infrastructure database) to determine the changes that were made, by comparing the old and new business rules.

You can automatically log any changes to business rules and selectors. You can configure your server to automatically detect when changes are made to business rules and selectors, and create an entry in a log file with the details of the changes done. You can choose for the log entries to be written to either the standard JVM **`SystemOut.log`** file, or to a custom audit log file of your choice. Depending on how the changes are made, the server where each business rule or selector change is made logs the:

- Name of the person who is modifying
- Location from where the change request originated
- Old business rule or selector object
- New business rule or new selector object

© Copyright IBM Corp. 2017

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

9.2. Dynamic routing

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic routing

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-11. Dynamic routing

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Dynamic invocation overview (1 of 2)

- Dynamic invocation capabilities cover all binding types and used through:
  - Java component
  - Mediation flow component
- You can use dynamic invocation to:
  - Route the message to a local endpoint within a mediation module
  - Route the message to a remote endpoint through a wired reference
  - Route the message to a remote endpoint through an unwired reference
  - Route the message to a remote endpoint through a named import within the mediation module
- Dynamic invocation requires a combination of:
  - A reference on the invoking component
  - An endpoint reference (EPR)

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-12. Dynamic invocation overview (1 of 2)

When a mediation module is developed and deployed, the flow of messages through the module uses static information. You can enter fixed values that the import, bindings, and targets describe by using IBM Integration Designer. Messages that pass through the mediation module use these values.

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic invocation overview (2 of 2)

- The invocation can take the following forms:
  - Uses an import that contains a static endpoint URI
  - Uses an import and dynamically specified URI
  - No import: uses a dynamically specified URI
  - No import is only for legacy-type web services (JAX-RPC and SOAP/JMS) and SCA bindings
  - No import cannot be done with HTTP, JMS, or IBM MQ binding types
  - No import cannot be done to invoke a web service through JAX-WS
- The reference
  - A reference on the component is always required
  - The reference might not be wired to an import
- The endpoint reference (EPR)
  - Contains some combination of URI, binding type, and import name

© Copyright IBM Corporation 2017

Figure 9-13. Dynamic invocation overview (2 of 2)

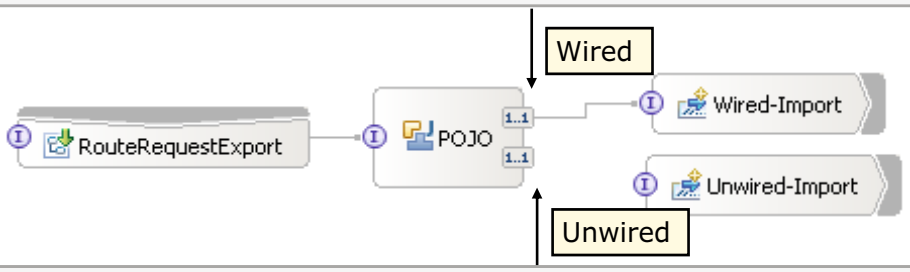
© Copyright IBM Corp. 2017

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

IBM Training

Example: Dynamic invocation that uses the EndpointReference APIs with HTTP binding

```
"EndpointReference epr =
     EndpointReferenceFactory.INSTANCE.createEndpointReference();
epr.setAddress(uri);
epr.setBindingType(EndpointReference.BINDING_TYPE_HTTP);
Service dynamicService = (Service)
ServiceManager.INSTANCE.getService(refname, epr);
DataObject customer = createCustomer(refname, "twoway", uri);"
```




```
"EndpointReference epr =
     EndpointReferenceFactory.INSTANCE.createEndpointReference();
epr.setAddress(uri);
epr.setImport(importName);
epr.setBindingType(EndpointReference.BINDING_TYPE_HTTP);
Service dynamicService = (Service)
ServiceManager.INSTANCE.getService(refname, epr);
DataObject customer = createCustomer(refname, "twoway", uri);"
```


© Copyright IBM Corporation 2017

Figure 9-14. Example: Dynamic invocation that uses the EndpointReference APIs with HTTP binding

This example demonstrates programmatic override of endpoint addresses.

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic endpoint selection in mediations (1 of 2)

- The runtime can route messages to an endpoint address that the message header element identifies
  - The mediation primitives can update the message header element, in a mediation flow
- Several different means are available to update an endpoint address:
  - **Dynamic routing by using registry lookup**: Choose which service is used at run time, by using a WebSphere Service Registry and Repository or Universal Description Discovery and Integration (UDDI) registry to look up service endpoints
  - **Dynamic routing by using message filters**: Choose which service is used at run time, by filtering a message
  - **Dynamic routing by using a database**: Choose which service is used at run time, by using a database to look up service endpoints

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-15. Dynamic endpoint selection in mediations (1 of 2)

You can determine the endpoint address in a number of ways at run time:

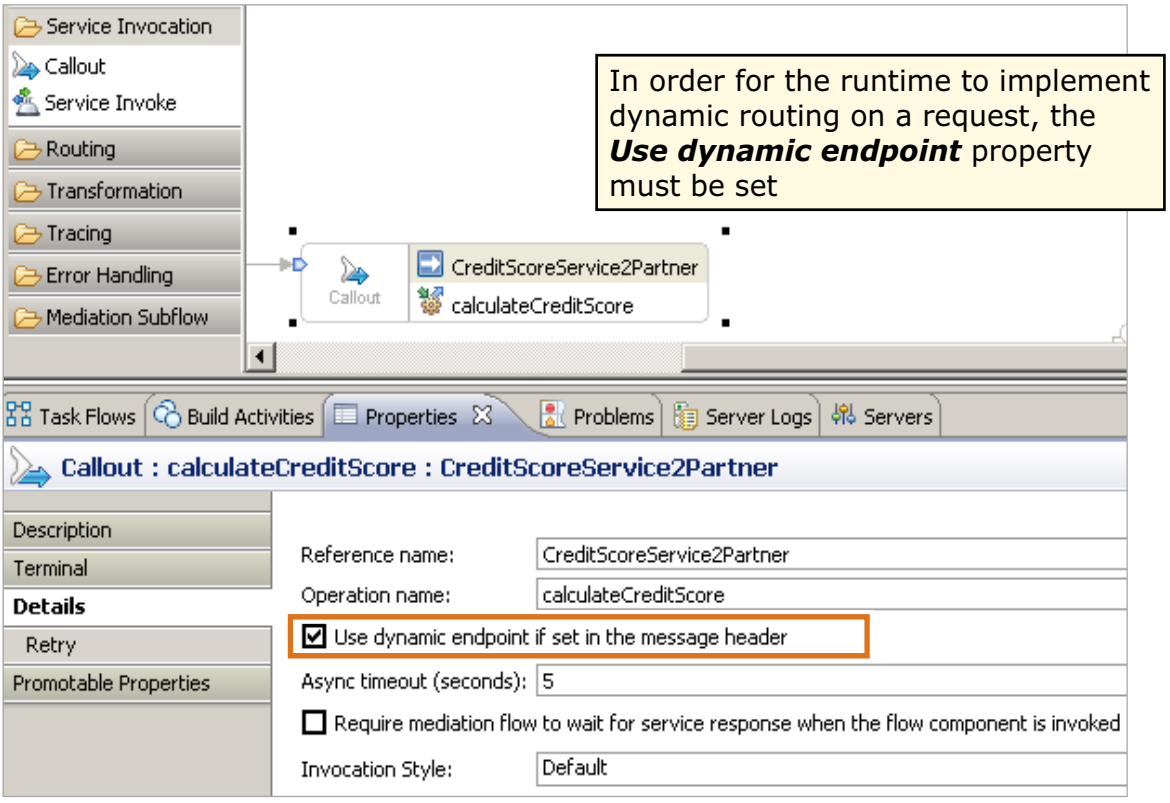
- You can do a “lookup” operation, by using a repository such as WebSphere Service Registry and Repository or a UDDI repository. Mediation primitives provide access to both of these repositories.
- You can retrieve endpoint values from a database table by using a database lookup mediation primitive. The values that are used to select the appropriate row from the table can be set dynamically elsewhere in the mediation.
- You can use a message filter mediation primitive to route a message through other paths in a mediation flow, which are based on the content of the message itself. For example, a message to a preferred customer can be routed through a different path of a mediation than a message for a regular customer, by checking a customer type field in the message.

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic endpoint selection in mediations (2 of 2)



Business process flexibility © Copyright IBM Corporation 2017

Figure 9-16. Dynamic endpoint selection in mediations (2 of 2)

The **Use dynamic endpoint** property in the callout node enables the dynamic routing that sets the callout to be dynamic. When the dynamic callout is started, it checks the `/headers/SMOHeader/Target/address` element in the message header for a service endpoint address, and routes the message to the address of that endpoint. If no address exists in the target address element, and an import is wired to the associated reference of the callout in the mediation flow component, a service is selected statically. By default, **Use dynamic endpoint** is enabled.

© Copyright IBM Corp. 2017

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

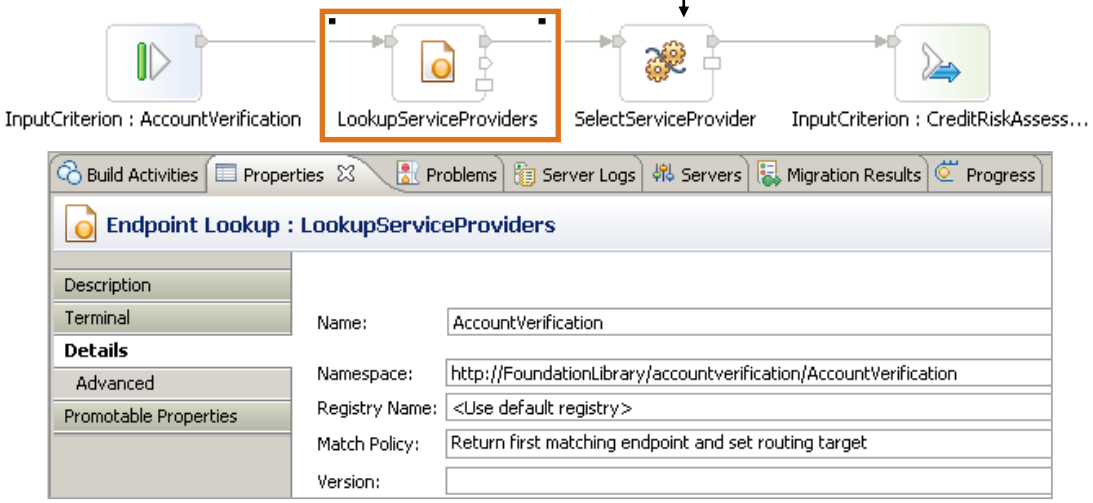
IBM Training

Endpoint lookup in mediations

Select endpoint by domain

- Endpoint lookup finds all endpoints for a service
- Custom code then:
  - Examines the returned endpoints
  - Selection favors providers in the same domain

Select a single service endpoint from the **/endpointReference/Address** element of **EndpointLookupContext**, and place it in the target address element of the **SMOHeader** (/headers/SMOHeader/Target/**address**)



© Copyright IBM Corporation 2017

Figure 9-17. Endpoint lookup in mediations

You can use the endpoint lookup mediation primitive to retrieve service endpoint information from a local or remote WebSphere Service Registry and Repository registry. The service endpoint information can relate directly to web services, or to Service Component Architecture (SCA) module exports.

(Note: mediation modules are types of SCA modules.)

To use the endpoint lookup mediation primitive, you might have to add service endpoint information to your WebSphere Service Registry and Repository registry. If you want to extract service endpoint information about web services, your WebSphere Service Registry and Repository registry must contain either the appropriate WSDL documents, or SCA modules that contain exports that use web service bindings. If you want to extract service endpoint information about exports that use the default SCA binding, your WebSphere Service Registry and Repository registry must contain the appropriate SCA modules.

You can use the endpoint lookup mediation primitive to retrieve service endpoint information that relates to the following bindings:

- Web services that use SOAP/HTTP
- Web services that use SOAP/JMS
- SCA module exports with web service bindings that use SOAP/HTTP

© Copyright IBM Corp. 2017

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

- SCA module exports with web service bindings that use SOAP/JMS
- SCA module exports with the default SCA binding

When the endpoint lookup mediation primitive receives a message, it sends a search query to the registry. The search query is constructed by using the endpoint lookup properties that you specify; and the query might return nothing, or it might return one or more service endpoints. You can select to be informed of all endpoints that match the query, or just one endpoint that matches the query.

The endpoint lookup mediation primitive has one input terminal (in), two output terminals (out and noMatch), and a fail terminal (fail). The in terminal is wired to accept a message, and the other terminals are wired to propagate a message. If an exception occurs during the processing of the input message, the fail terminal propagates the input message, together with any exception information. If service endpoints are retrieved from the registry, the out terminal propagates the original service message object (SMO) which the service endpoint information modifies. If no services are retrieved from the registry, the noMatch terminal propagates an unmodified message.

In the example that is shown on the visual, the custom mediation primitive copies the service endpoint from the `/endpointReference/Address` element of the EndpointLookupContext into the `/headers/SMOHeader/Target/address` element of the SMO header. After copying endpoints, the callout node uses it to route the request message.

Note:

For the runtime engine to implement dynamic routing on a request, you must set the **Use** dynamic endpoint if set in the message header property in the callout node or the service start mediation primitive. You can specify a default endpoint that the runtime engine uses when it cannot find a dynamic endpoint. You specify a default endpoint by wiring an import for which the default service is selected.

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic invocation by using the SMO structure

- The SMO structure enables dynamic invocation by overriding the endpoint address or by using a target import:
  - **address**: The address field contains the dynamic invocation target service URI
  - **bindingType**: The bindingType provides more details about the URL, indicating the type of binding used during a dynamic invocation
  - **import**: The import field provides the name of a target import for dynamic invocation

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-18. Dynamic invocation by using the SMO structure

The SMO structure overrides the endpoint address, or uses a target import as follows:

```
/headers/SMOHeader/Target/address
/headers/SMOHeader/Target/@bindingType
/headers/SMOHeader/Target/@import
/headers/SMOHeader/AlternateTarget/address
/headers/SMOHeader/AlternateTarget/@bindingType
/headers/SMOHeader/AlternateTarget/@import
```

Where the following fields apply:

- Address: The `address` field contains the dynamic invocation target service URI for requests, or the dynamic reply address URI for responses if a web service (JAX-WS) binding is being used.
- @bindingType: When requests are being routed, the `bindingType` field provides more details about the URL, indicating the type of binding used during a dynamic invocation.
- @import: When requests are being routed, the `import` field provides the name of a target import that is used for dynamic invocation.

You can set each of these fields or leave them unchanged, depending on the dynamic invocation you required. The fields can be inspected and set by using appropriate mediation primitives. For example, you can dynamically override the endpoint address of a message with a service that the

© Copyright IBM Corp. 2017

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

JMS-based URI identifies, by setting values for the `Target/address` and `Target/@bindingType` fields.

© Copyright IBM Corp. 2017

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

9.3. Repair process navigation

© Copyright IBM Corp. 2017

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

IBM Training

Repair process navigation

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-19. Repair process navigation

© Copyright IBM Corp. 2017

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

IBM Training

Repair and dynamicity features

- Jumping and skipping
  - Provides greater flexibility for predefined processes
  - Allows adapting a business process to a changed situation easily
  - Makes it possible for business users and solution administrators to override the navigation of a process
  - Jump forward and back between activities of a running process
  - Skip specific activities within a running process
  - Incorporate process relevant data changes in-flight
  - Included support for dynamic modifications in graphical process widget by using Business Process Choreographer Explorer for solution administrators

© Copyright IBM Corporation 2017

Figure 9-20. Repair and dynamicity features

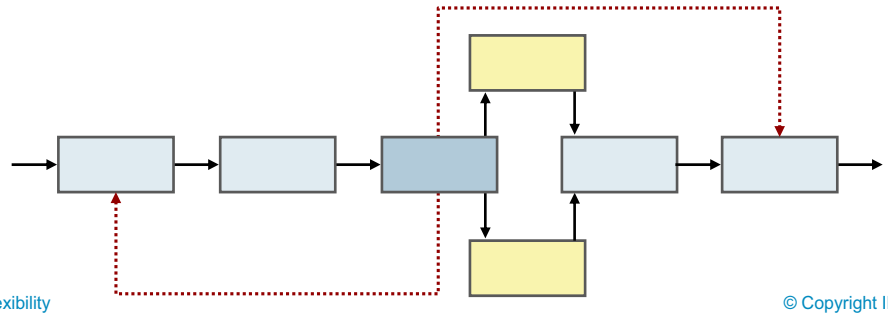
© Copyright IBM Corp. 2017

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

IBM Training

Jump between activities of a process

- To support dynamic modifications in in-flight processes, jumping forward and backward between activities within a running process is introduced
- You can jump forward or back in a business flow
  - Source of jump is a basic activity in running state, which includes ready, claimed, and running states
  - Target of jump can be a basic or structured activity
  - Jump into nested constructs is not supported
- Possible within single threads or branches of a parallel flow
- Possible to jump out of scopes with a single activity



Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-21. Jump between activities of a process

You can jump from one activity of a process instance to another activity of the process instance. You can select to complete the source activity before you jump to a target activity.

Jumping is needed to allow start activities that contain associated handlers (such as fault or compensation handlers) as jump targets, since specifying a handler for an invoke activity surrounds the activity in a scope.

Jumping is supported between activities in a cyclic flow or sequence. The source of the jump is a basic activity, such as an invoke or human task, which is in a “running” state (like “ready”, “claimed”, or “running”). The target of the jump can be both a basic and a structured activity. Jumping into nested constructs is not supported (such as jumping into a sequence).

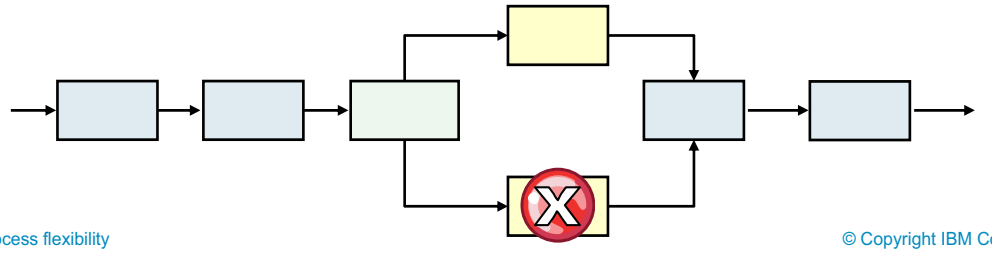
© Copyright IBM Corp. 2017

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

IBM Training

Skip activities within a process

- To support dynamic modifications for in-flight processes, you can skip an activity that is not needed for a particular process
  - Skip current active activity or future activity
  - Future activities can be marked as “skipReqested”
  - Only basic activities can be skipped
  - Skip can be combined with jump
- Skipping helps in dynamic human workflow and case handling scenarios
- Skipping activities can also be undone
  - Cancel the request to skip future activities
  - Use **`cancelSkipRequest()`** API



Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-22. Skip activities within a process

Activities that are currently active can be skipped immediately, such as those activities in a nonterminal state (for example, “running” or “ready”). Future activities can be marked “skipRequested” so that they are skipped if they are targeted. The request to skip a future activity can be canceled by using the appropriate APIs.

Only basic activities can be skipped. Structured activities, such as scopes and sequences, cannot be skipped.

The skips and jumps in the activities can be combined.

© Copyright IBM Corp. 2017

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

IBM Training

Update variables of a process

- Update or modify variables of a process
  - Set process variables while process instance is running
  - Useful in repair scenarios where inconsistent data needs to be changed
- Supported through APIs
  - **`getAllVariableNames()`**: Returns the names of all variables
  - **`getInitialVariableValue()`**: Retrieves the initial value of the specified variable
- Business Process Choreographer Explorer supports updating variables
  - Accomplished by using the **activity instance details** page
  - Variables that are visible from a specific activity can be updated

© Copyright IBM Corporation 2017

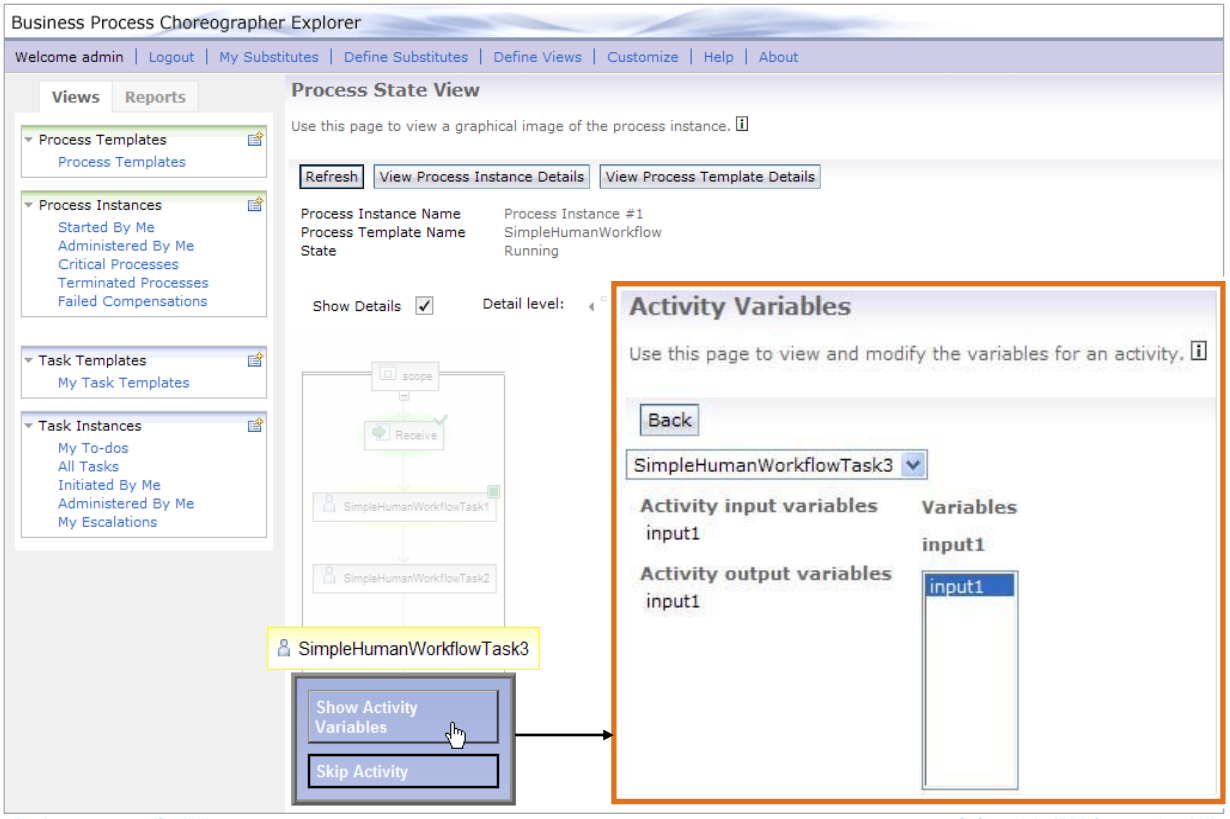
Figure 9-23. Update variables of a process

© Copyright IBM Corp. 2017

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

IBM Training

Repair and dynamicity features



© Copyright IBM Corporation 2017

Figure 9-24. Repair and dynamicity features

Get variables for activities not yet accessed; in Business Process Choreographer Explorer, variables of activities that are not yet accessed can be viewed and modified:

- When selecting an activity instance for skip, forceComplete, or completeAndJump from an activity instance list
- From a process instance details page by clicking New, triggering a new view to select an activity for which to show the list of variables
- From the graphical process widget when selecting a future activity and selecting “Show Activity Variables” in the menu

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Identify the business process components that provide application flexibility and dynamicity
- Explain the purpose and business value of selectors
- Describe the runtime behavior of selectors
- Provide a high-level description of the runtime architecture of content-based routing
- Identify the BPEL components that allow dynamic process navigation at run time
- Describe the runtime components that support the dynamic repair of process instances

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-25. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: You can jump to a structured activity but not into one.
2. True or False: To support dynamic modifications for in-flight processes, you can skip an activity that is not needed for a particular process.
3. True or False: Selector targets can be chosen based only on date and time criteria.

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-26. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: You can jump to a structured activity but not into one.
   The answer is True.
2. True or False: To support dynamic modifications for in-flight processes, you can skip an activity that is not needed for a particular process
   The answer is True.
3. True or False: Selector targets can be chosen based only on date and time criteria.
   The answer is True.

Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-27. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Creating flexible business processes

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 9-28. Exercise 8

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

IBM Training

Exercise objectives

- Use the Business Process Choreographer Explorer client to dynamically change the execution path of a running process instance
- Implement a selector component to choose between multiple service invocations
- Test a selector component in the IBM Integration Designer integrated test client

Business process flexibility

© Copyright IBM Corporation 2016

Figure 9-29. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

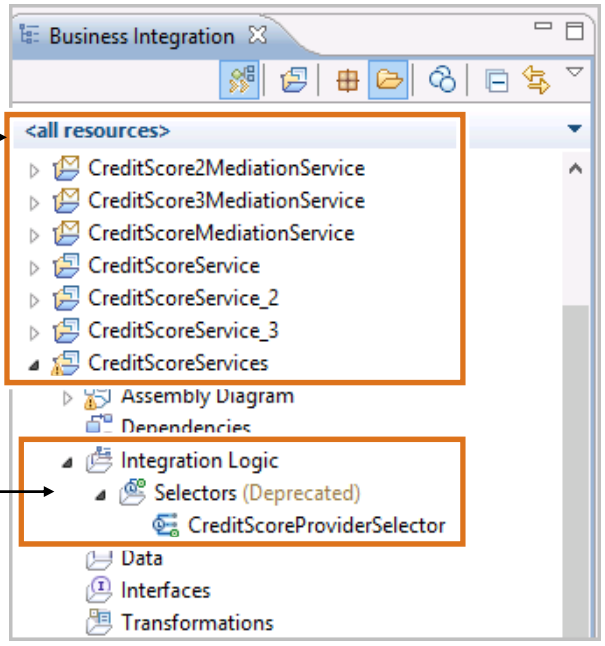
Components that are required for Exercise 8

Prebuilt components that are imported in the lab:

1. **CreditScore2mediationService**
2. **CreditScore3mediationService**
3. **CreditScoreService**
4. **CreditScoreService_2**
5. **CreditScoreService_3**
6. **CreditScoreServices**

New components that you create in the lab:

1. **CreditScoreMediationService** mediation subflow
2. **CreditScoreProviderSelector**



© Copyright IBM Corporation 2017

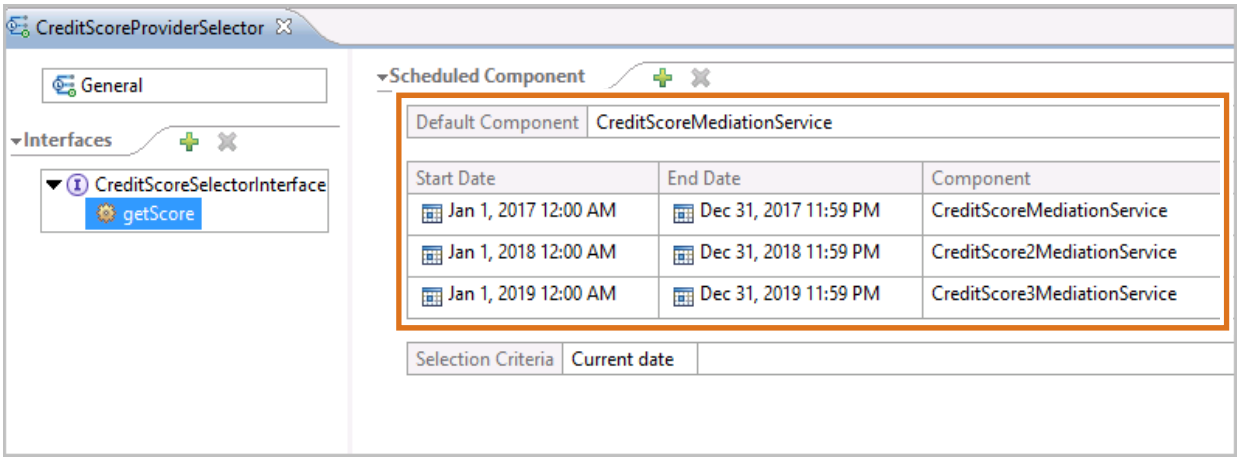
Figure 9-30. Components that are required for Exercise 8

© Copyright IBM Corp. 2017

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

IBM Training

Create the CreditScoreProviderSelector component in Exercise 8



Business process flexibility © Copyright IBM Corporation 2017

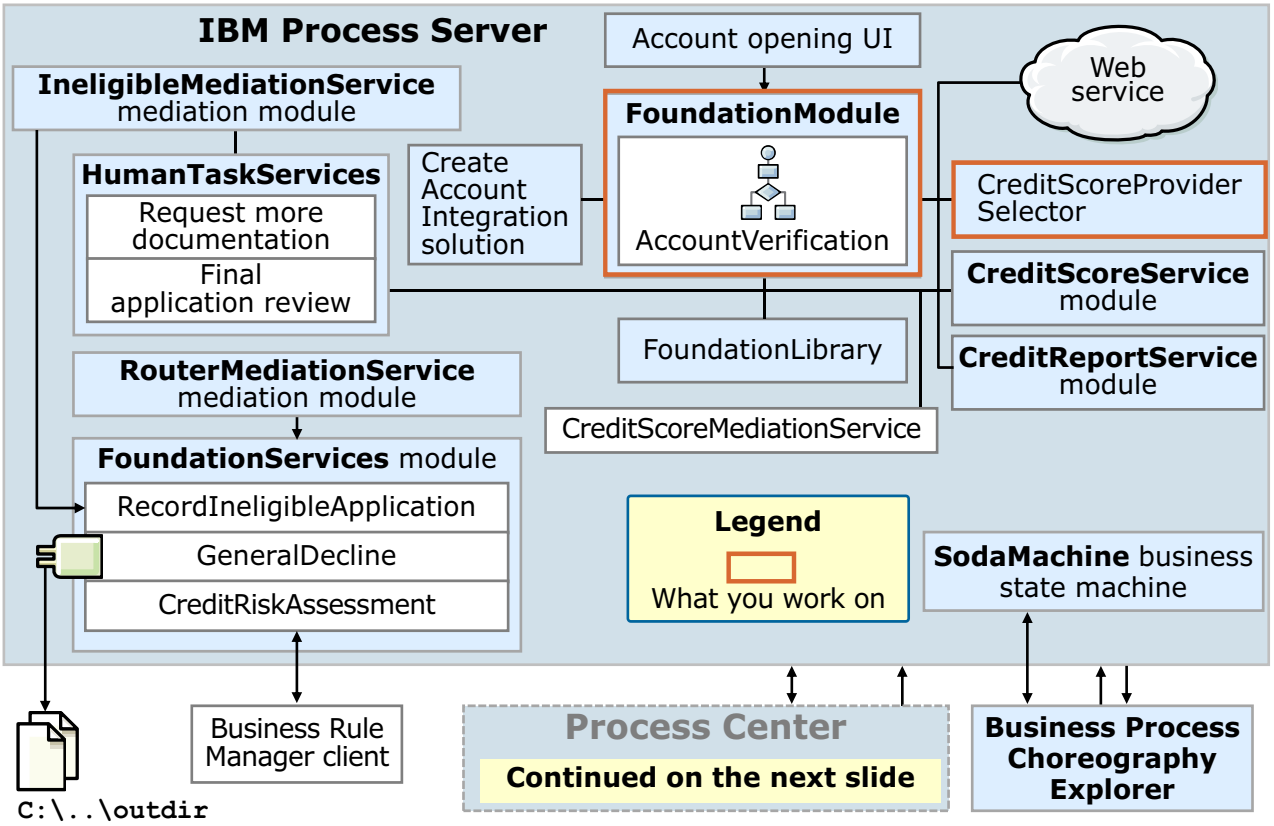
Figure 9-31. Create the CreditScoreProviderSelector component in Exercise 8

In the exercise, after creating the implementation, you test the application by using **companyName** IBM. Depending on when this test is run, the resulting credit score is either 9, 10, or 11. Recall from the selector component criteria that three possible mediation services can be invoked. **CreditScoreMediationService** is invoked between **1/1/17** and **12/31/17** and returns a **credit** score of 11. **CreditScore2MediationService** is invoked between **1/1/18** and **12/31/18** and returns a **credit score** of 10. **CreditScore3MediationService** is invoked between **1/1/19** and **12/31/19** and returns a **credit score** of 9. The result varies depending on the year in which the service is invoked. Any year other than 2017, 2018, or 2019 results in invoking the **Default Component**, which is **CreditScoreMediationService**, in which case the credit score is always 11.

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

IBM Training

Course business scenario: What you work with in Ex 8



Business process flexibility

© Copyright IBM Corporation 2017

Figure 9-32. Course business scenario: What you work with in Ex 8

In this lab, you work with IBM Integration Designer.

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

Unit 10. Relationships

Estimated time

01:00

Overview

This unit describes dynamic and static relationships in IBM Process Server.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe the purpose and business value of static and dynamic relationships
- Describe the runtime components of relationships
- Explain the deployment of SCA applications with relationship components
- Explain how to use the Relationship Manager tool to manage runtime relationship instances
- Describe the methods of predefining relationship tables with existing data
- Explain the use of composite keys in relationships
- Explain when foreign key cross-referencing or lookup is required

Relationships

© Copyright IBM Corporation 2017

Figure 10-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Introduction to relationships
- Dynamic relationships
- Static relationships
- Runtime aspects of relationships
- Managing relationships

© Copyright IBM Corporation 2017

Figure 10-2. Topics

© Copyright IBM Corp. 2017

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

10.1. Introduction to relationships

© Copyright IBM Corp. 2017

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

IBM Training

Introduction to relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

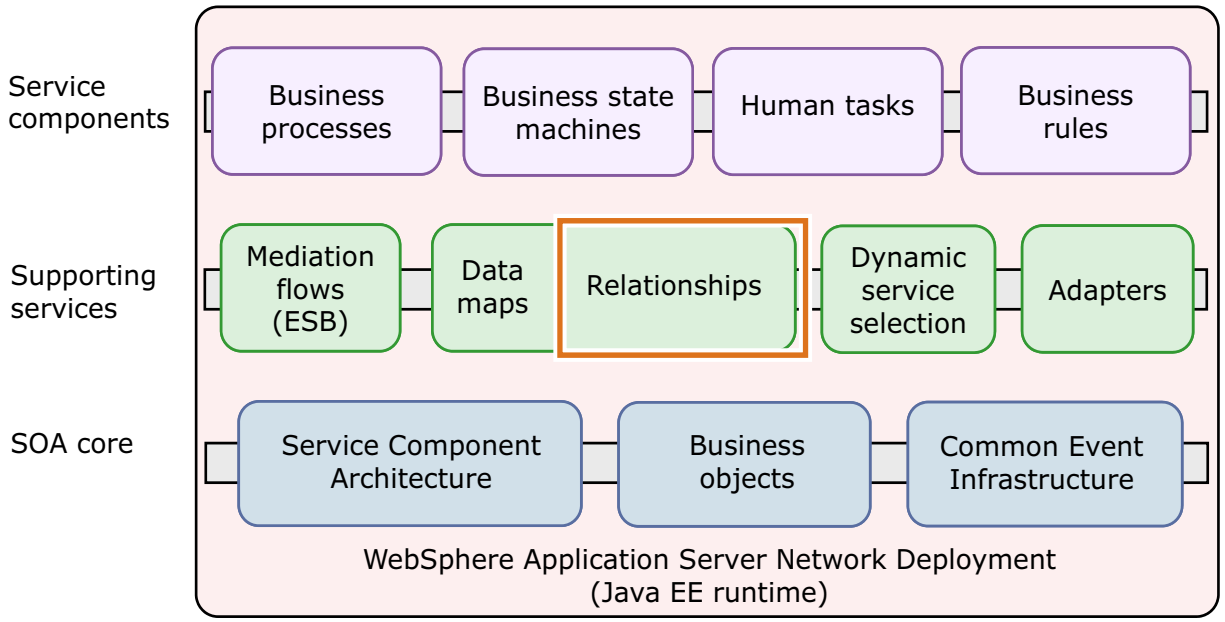
Figure 10-3. Introduction to relationships

© Copyright IBM Corp. 2017

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

IBM Training

Relationships are a supporting service



Relationships

© Copyright IBM Corporation 2017

Figure 10-4. Relationships are a supporting service

Above the SOA core, IBM built a set of more services that provide higher-level abstractions and ease-of-use capabilities. Although the SOA core is key to the operation of SOA solutions, they are not sufficient for the whole story. A set of higher-level services that use the SOA core is also available. These services are interface maps, business object maps, and relationships. The relationship service maintains relationships between business objects across disparate systems.

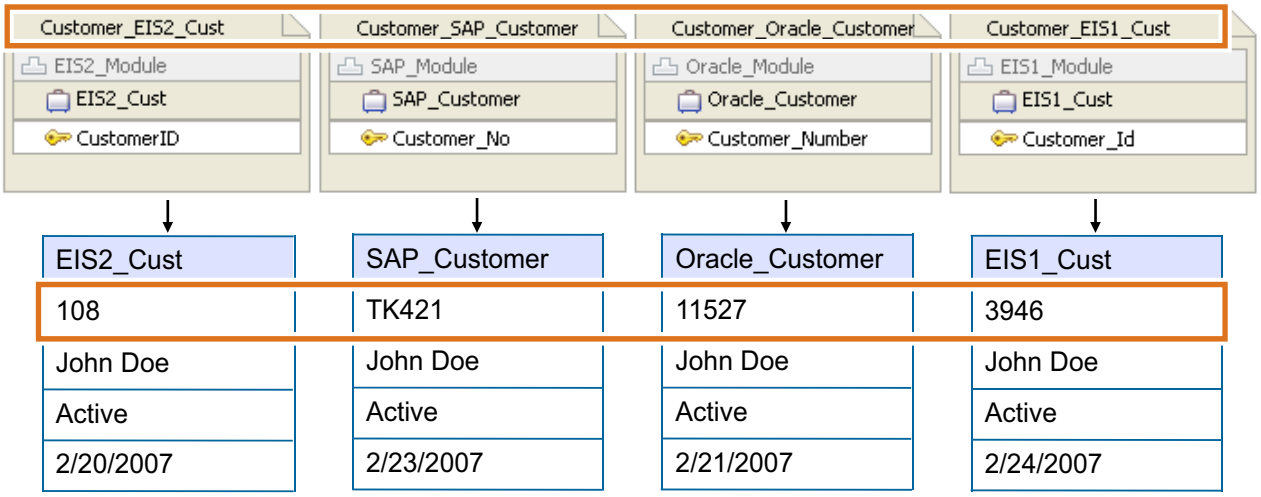
© Copyright IBM Corp. 2017

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

IBM Training

Overview: Relationships

- A relationship establishes associations between semantically equivalent business entities
  - For example, customer business objects in different EIS applications with different keys
  - Correlating key values allows synchronization of record creation and updates across systems



Relationships

© Copyright IBM Corporation 2017

Figure 10-5. Overview: Relationships

The ability to be able to map from one identification of an entity to another is a core component in a powerful SOA solution. IBM Process Server provides for the automated association of existing entities, or one service that is managing one business entities and the other service that is managing the same logical entity.

Different EIS systems might each have their own way of defining the same business data. Some systems store customer given name and surname as a separate attribute, and some do not. The date format can be varied across different systems. Some systems use a numeric value to define a customer status, and some use a string value.

However, they are all expressing the same data. Business object maps can handle such mismatches, but the unique customer ID cannot be mapped from one business object to another. As you can see, the customer ID for John Doe is different among the four EIS systems. Here the relationship definition helps on top of business object maps.

Process Server manages the relationship data, not the application, thus simplifying the task of the integration developer.

© Copyright IBM Corp. 2017

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

IBM Training

Relationship components

Relationships have a *definition*:

- The definition declares the structure of the relationship and its constraints
  - For example, the “Customer” relationship would be defined as a relationship between Customer object types, not Employees
- Each relationship definition has a set of role definitions that define how entities (EIS systems) participate in the relationship

Relationships have *roles*:

- The role definition captures structure and constraints of the **individual business objects**, including the properties and attributes that form the relationship
- For example, EIS1 participates in the “Customer” relationship through the EIS1 Customer object, which the key: Customer_ID identifies

Relationships

© Copyright IBM Corporation 2017

Figure 10-6. Relationship components

Again, “relationship” is one of the transformation types that you can set from a business object map. Therefore, the participants are at the business object level. Relationships are an SCA artifact that you must define in a shared library, and those modules that contain participating business objects must include the library as a dependency.

© Copyright IBM Corp. 2017

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

10.2. Dynamic relationships

© Copyright IBM Corp. 2017

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

IBM Training

Dynamic relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-7. Dynamic relationships

© Copyright IBM Corp. 2017

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

IBM Training

Relationship types: Dynamic relationships

- Dynamic relationships have two basic types: identity and non-identity
  - Dynamic relationships are established at run time (the business object and the business event create their unique identifier)
  - Dynamic relationships have at least two roles
- Identity relationships are one-to-one relationships
  - They represent the relationship between the unique keys in different systems that apply to the same record
  - For example, John Doe is CustomerID 1 in EIS1 and Customer_Number TX30 in EIS2
  - They require another role: a managed role
  - The managed role represents the generic business object and its key
  - The relationship service controls it at run time
  - Roles in identity relationships can have more than one key attribute except the managed role
- Non-identity relationships are one-to-many or many-to-many
  - They can contain roles with composite keys

Relationships © Copyright IBM Corporation 2017

Figure 10-8. Relationship types: Dynamic relationships

An identity relationship is also called a **dynamic relationship**. An identity relationship gets established at run time when you have the actual live data. When you are synchronizing customer data across four different EIS systems, you are using mapping components to resolve the interface and parameter mismatches. Business object mapping always involves a generic business object (GBO). For relationships, it is the same. If you have four different application-specific business objects (ASBO), only the GBO is in common across the four.

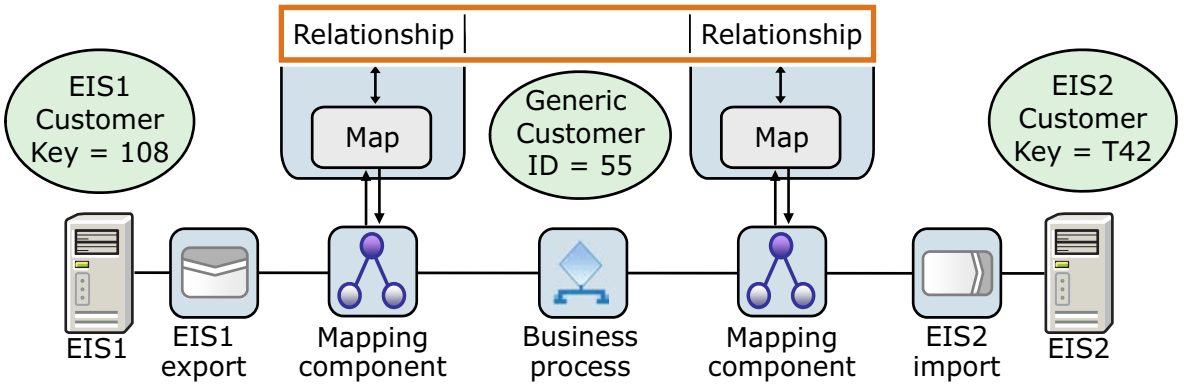
Other types of relationships, such as static relationships and composite relationships, are described later in this unit.

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

IBM Training

Identity relationships in IBM Integration Designer (1 of 2)

- When you synchronize business objects in different EIS applications, key values cannot use simple transformations
- To transform keys from different systems, a relationship transformation is added to the interface and business object map transformation components
- Relationships can also be called from a process component (by using Java code) or from a POJO component



© Copyright IBM Corporation 2017

Figure 10-9. Identity relationships in IBM Integration Designer (1 of 2)

Relationships are used to establish associations between attributes of multiple related business entities. These attributes cannot be easily mapped because:

- EIS applications maintain their own mechanisms for generating unique values for those attributes, so they cannot be copied across the integration environment.
- An attempt to implement transformations without relationships would be complicated, especially when a cross-reference is needed for a subsequent processing in the future.
- Generic business objects also require a key that correlates to the key values in the EIS applications.

This slide presents the issue of cross-referencing between two different and distributed EIS systems. Suppose that *MyCompany International* has 30,000 employees worldwide. Their payroll organization manages the employee data by using system EIS1, and the HR organization uses EIS2. If the role of an employee who is named John Smith is changed within a company, then the job title, department ID, and other information of John Smith must be updated in the database. The HR person updates the employee data of John in the HR system, and wants to send the updates to the payroll. How do you ensure that this update does not end up adding an entry into EIS1? A mechanism is necessary to cross-reference the data of John in two different systems. The employee database might not be centralized, especially now that many companies outsource HR jobs to an agency.

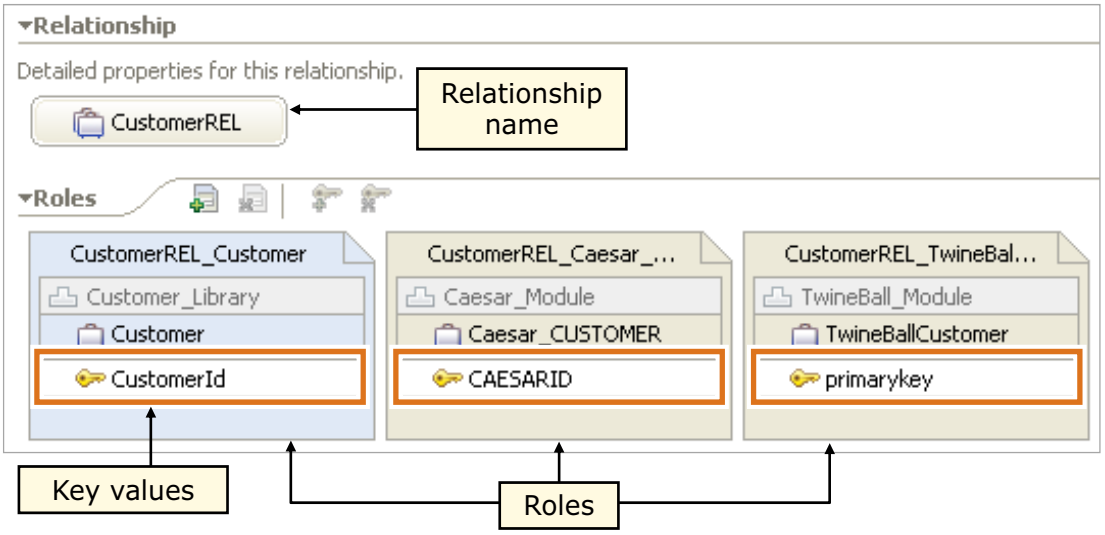
© Copyright IBM Corp. 2017

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

IBM Training

Identity relationships in IBM Integration Designer (2 of 2)

- An identity relationship in IBM Integration Designer has a name and a one-to-one cardinality
- The roles in the relationship represent the business objects of the participating business entities



© Copyright IBM Corporation 2017

Figure 10-10. Identity relationships in IBM Integration Designer (2 of 2)

The relationship hosts roles. Each relationship role is associated with a specific business entity and also with a specific business entity attribute. Ultimately, a relationship defines all roles with attributes that contain the unique IDs for all entities that are associated together, for example, all customer ID attributes.

Each role has a key that is used to uniquely identify the object for that participant. Identity relationships must also include a managed role.

In the example that is shown here, this “CustomerREL” relationship definition has two possible roles. The key attributes that are used for them are CAESARID and **primaryKey**. Those key attributes are what you use to cross-reference between the two systems. A managed role also exists (labeled CustomerREL_Customer, the first role that is shown on the visual) which is described shortly.

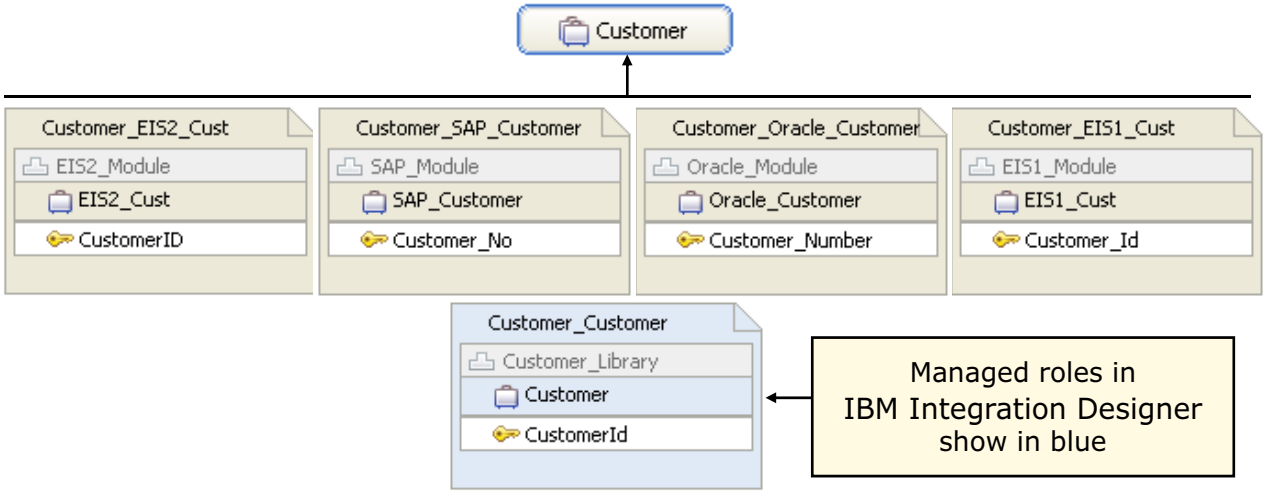
© Copyright IBM Corp. 2017

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

IBM Training

Managed roles in identity relationships

- At run time, the managed role coordinates records by using a system managed key: the instance ID
- The instance ID is used to populate the key value for the generic business object and to identify the relationship between the generic object and the application-specific objects in each EIS



Relationships

© Copyright IBM Corporation 2017

Figure 10-11. Managed roles in identity relationships

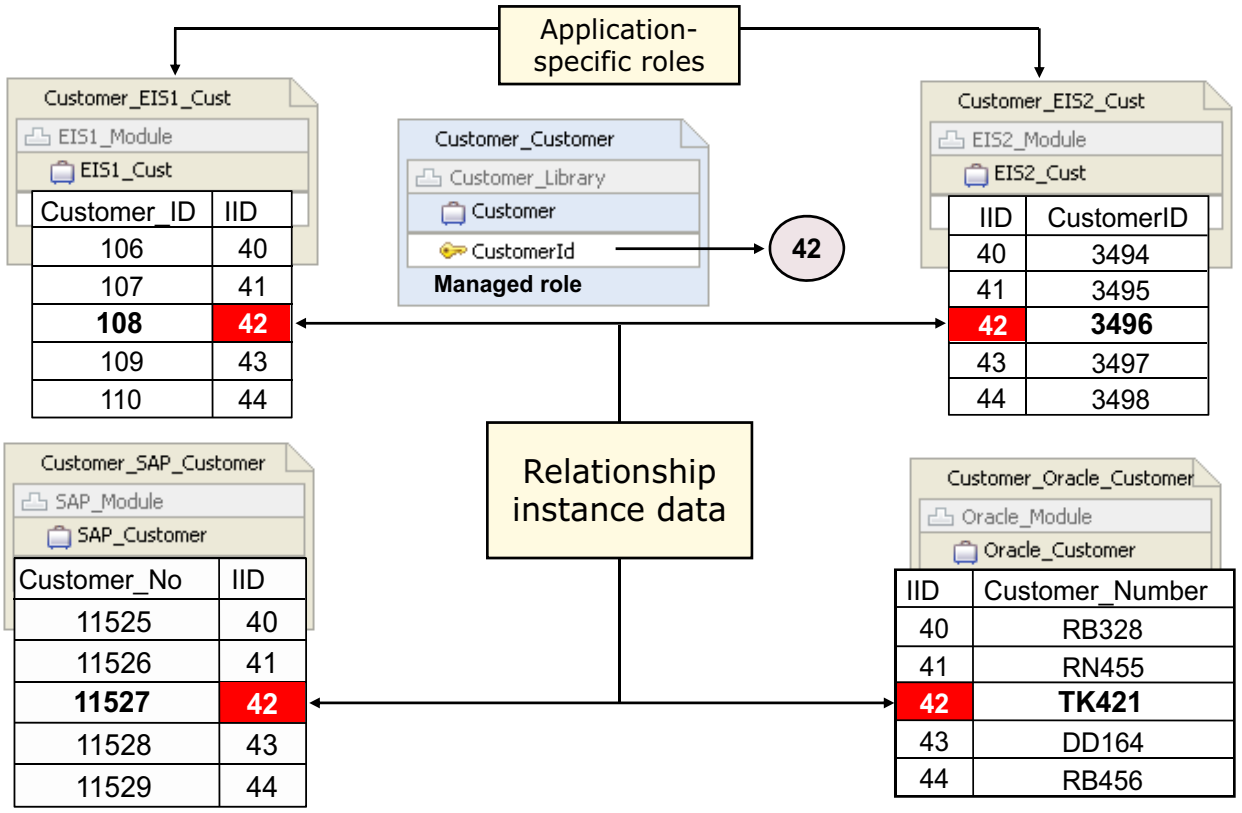
According to the relationship architecture, the business object maps are used to convert ASBO to GBO, and GBO to ASBO. Therefore, the relationship definition also includes a managed role for associating the GBO with the participants of the relationship.

© Copyright IBM Corp. 2017

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

IBM Training

Example: Identity relationship



Relationships

© Copyright IBM Corporation 2017

Figure 10-12. Example: Identity relationship

In an identity relationship, the relationship service generates a role table per participant. Notice the common IBM Integration Designer (instance ID) among all four tables. The relationship service manages this value. The managed role is used to associate the GBO with the other relationship participants by placing the IBM Integration Designer in the GBO key field. In the example:

1. Customer John Doe updates his address through EIS1, which must be sent to EIS2, SAP, and Oracle for synchronization.
2. In EIS1, the unique customer ID of John is 108. When the update gets sent out, the relationship service looks up the IBM Integration Designer, which is 42.
3. A business object map is started to convert EIS1 ASBO to GBO. For the GBO, the key attribute gets set to 42 instead of 108.
4. When preparing to send the data to EIS2, the business object map gets called to convert the GBO to an EIS2 ASBO.
5. The relationship service looks up the role table that is associated with EIS2 to find the corresponding **CustomerID** value, 3496.
6. The business object map assigns 3496 to the CustomerID attribute of EIS2 ASBO.

© Copyright IBM Corp. 2017

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

7. The target service (an adapter) then updates the entry with key attribute 3496 in the EIS2 system.

© Copyright IBM Corp. 2017

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

10.3. Static relationships

© Copyright IBM Corp. 2017

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

IBM Training

Static relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-13. Static relationships

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

IBM Training

Relationship types: Static relationship

- Static relationships are also called lookup relationships
- Static relationships do not require managed roles
  - Data can be predefined in the relationship editor
- Business data in static relationships rarely (if ever) changes
  - CA can represent a state in one application but show as 05 in another
  - USA can represent a country in one application but show as 105 in another
- In the business object map, you can have more than one relationship; you can have both static and dynamic relationships
  - For example, in an identity relationship, when you create a Customer business object with a unique key in EIS1, that must be mapped to a corresponding Customer business object (also with a unique key) in EIS2
  - You can also map the state attribute in EIS1 to the state attribute in EIS2

Relationships

© Copyright IBM Corporation 2017

Figure 10-14. Relationship types: Static relationship

The second type of relationship that is described in this unit is the static relationship.

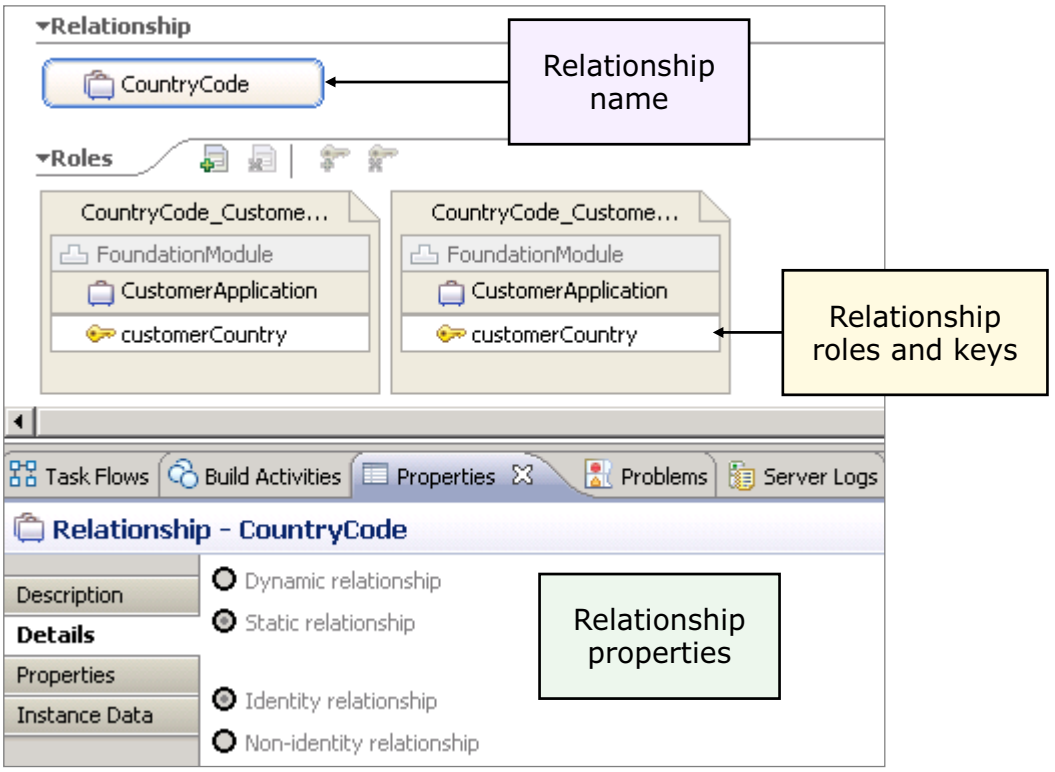
A static relationship is used to map data from one application to the other and to define the values for the mapping during the definition time of the relationship (for example, for state codes). In one application the state code might be “California”, while in another application it might be “CA”. A static relationship is created between these two pieces of data, which indicates that they are equivalent. The participant data that these types of relationships contain have values that do not change frequently. These relationships exist for both identity relationships and non-identity relationships.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: Defining the components



© Copyright IBM Corporation 2017

Figure 10-15. Static relationship: Defining the components

IBM WebSphere Integration Developer provides a relationship wizard to help you define relationships. You specify:

- The relationship name
- The type of relationship (static or dynamic, identity, or non-identity)
- The participating roles and their key fields

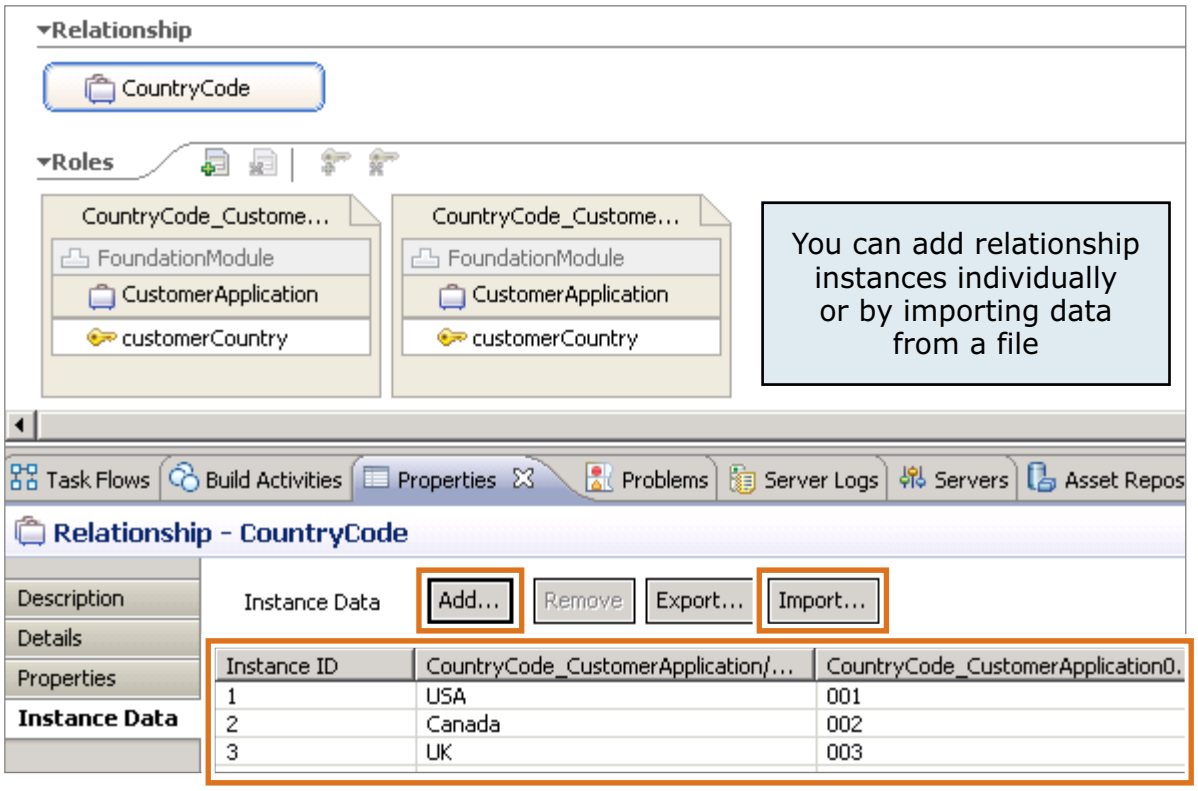
The relationship information is then visible in the Properties tab.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: Defining the instance data



© Copyright IBM Corporation 2017

Figure 10-16. Static relationship: Defining the instance data

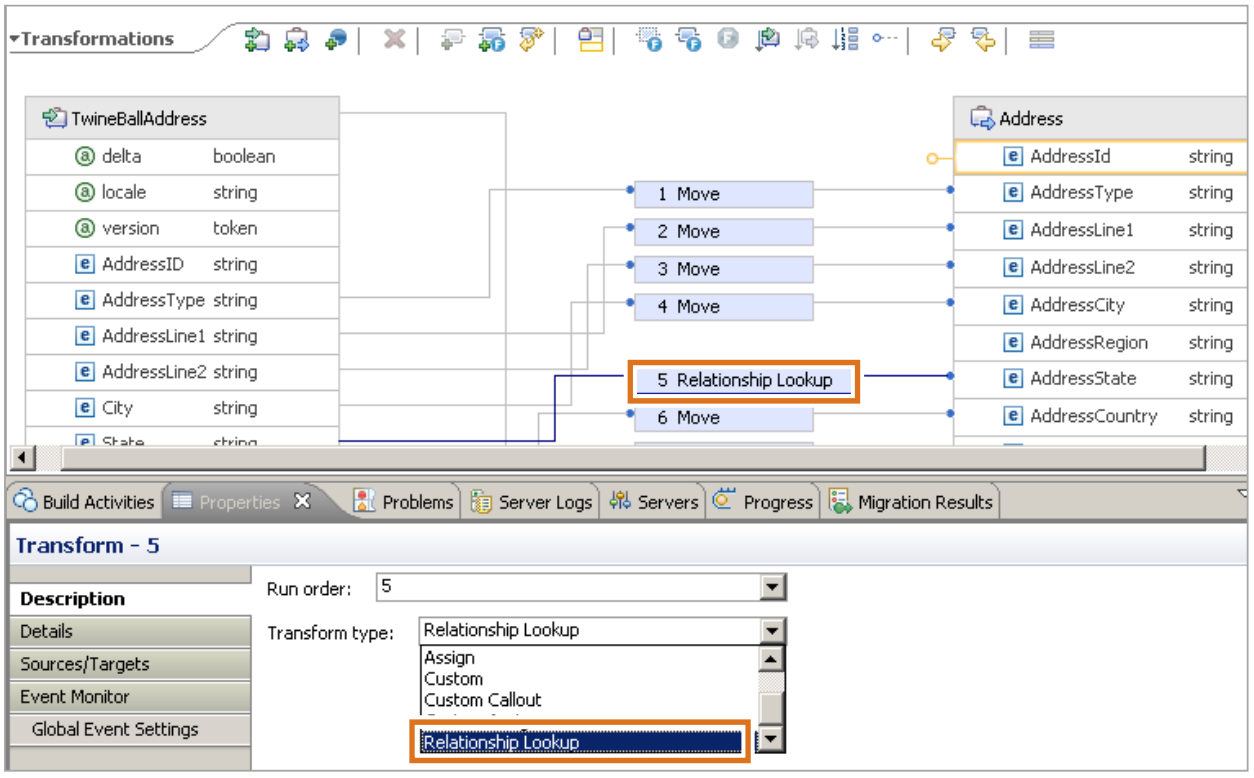
With a static relationship, after you define the participating business objects and keys, you can then enter the static lookup data in the Instance Data tab of the Properties view. You can also populate the instance data by importing a file that contains comma-separated values. These same operations can be done from the Relationship Manager in the administration console.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: Using the relationship lookup (1 of 2)



© Copyright IBM Corporation 2017

Figure 10-17. Static relationship: Using the relationship lookup (1 of 2)

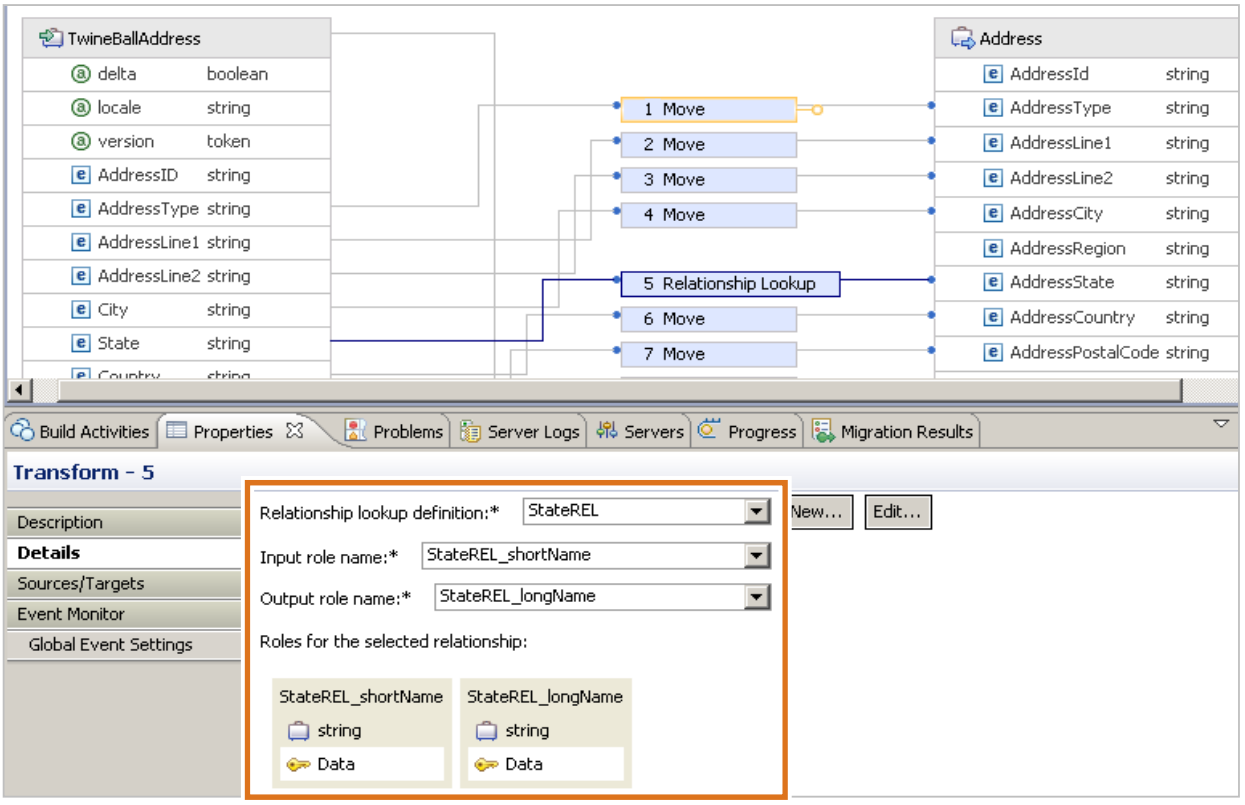
After you define a relationship, you can use it in a map. First, specify a transformation type of **Relationship Lookup**.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: Using the relationship lookup (2 of 2)



Relationships

© Copyright IBM Corporation 2017

Figure 10-18. Static relationship: Using the relationship lookup (2 of 2)

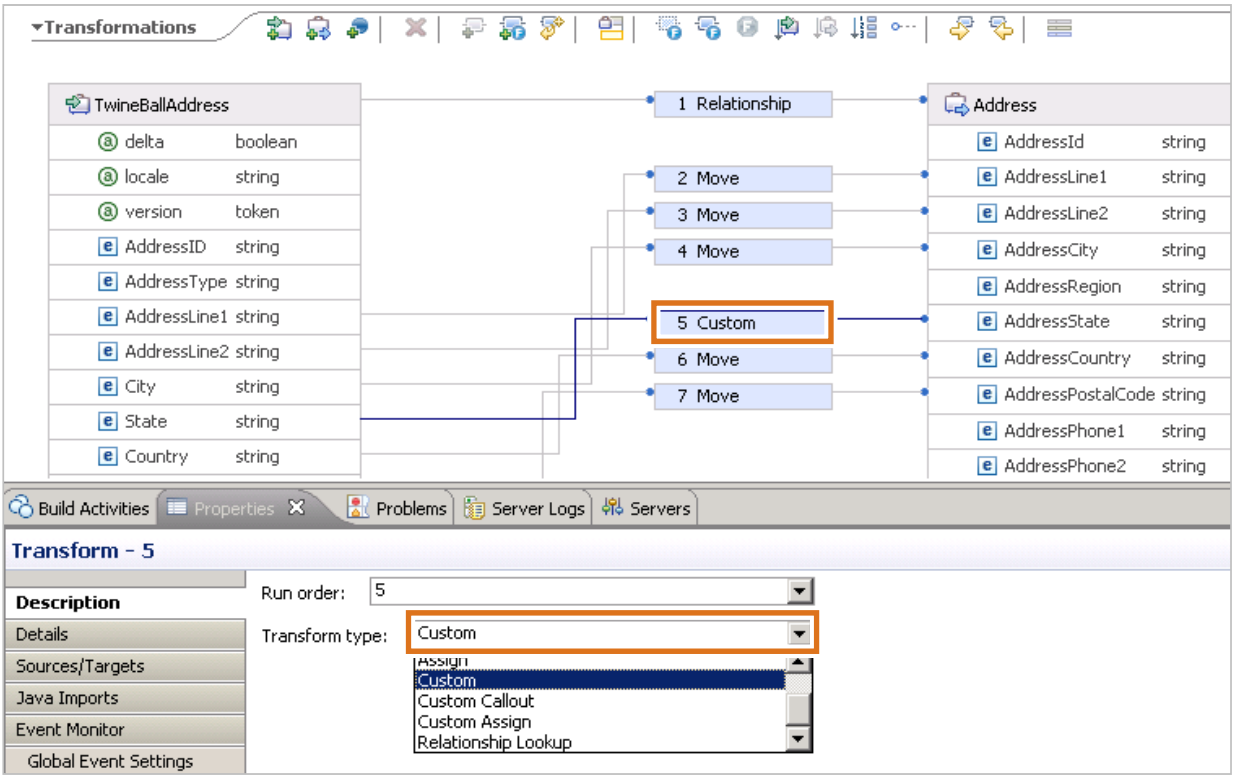
Then, define the role names that are used in the relationship. At run time, the prepopulated instance data is used to do the lookup.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: BO map implementation (1 of 2)



© Copyright IBM Corporation 2017

Figure 10-19. Static relationship: BO map implementation (1 of 2)

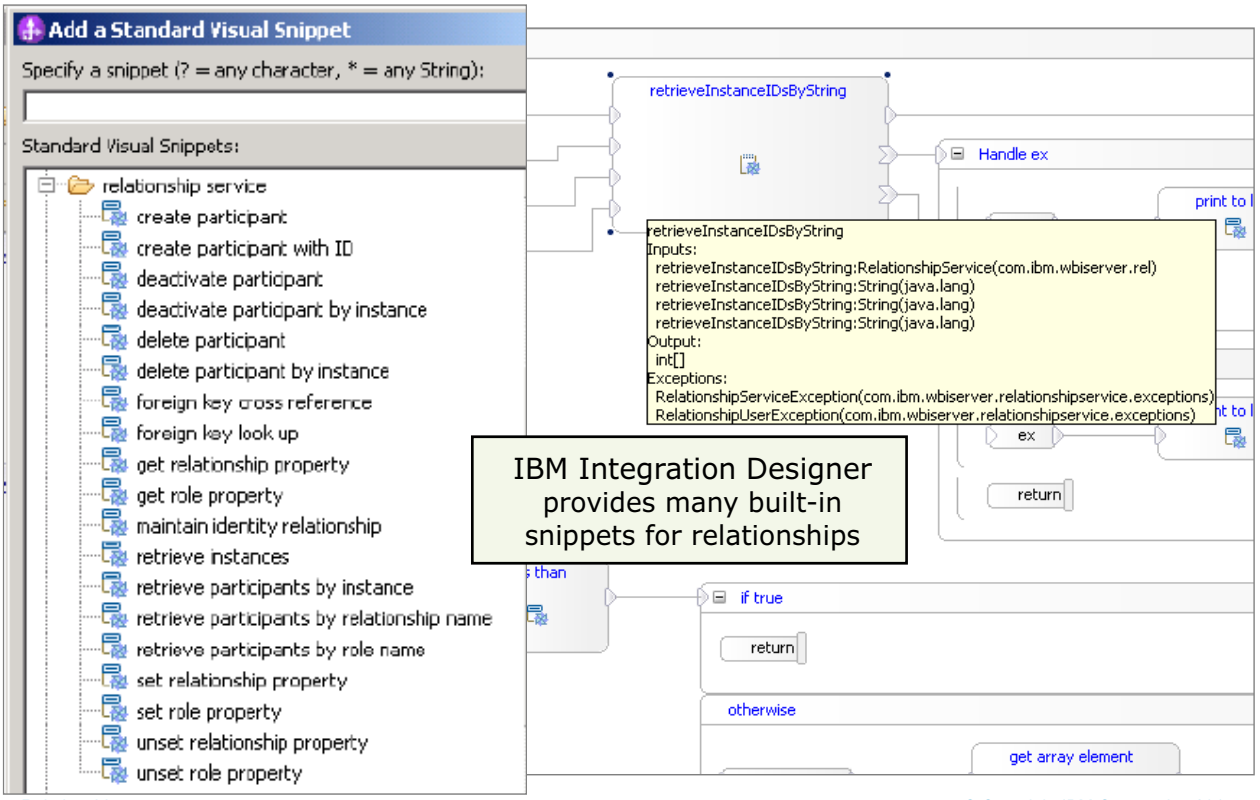
A second way of starting a relationship is by using a custom transformation. Specify a transform type of **Custom**.

© Copyright IBM Corp. 2017

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

IBM Training

Static relationship: BO map implementation (2 of 2)



Relationships © Copyright IBM Corporation 2017

Figure 10-20. Static relationship: BO map implementation (2 of 2)

Then, use the appropriate predefined visual snippets from the RelationshipService category of the Java visual Snippet editor. A number of snippets are provided to:

- Create, retrieve, and delete relationship participants
- Create, retrieve, and modify role and relationship properties
- Retrieve instances

It also runs other relationship-related actions.

© Copyright IBM Corp. 2017

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

10.4. Runtime aspects of relationships

© Copyright IBM Corp. 2017

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

IBM Training

Runtime aspects of relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

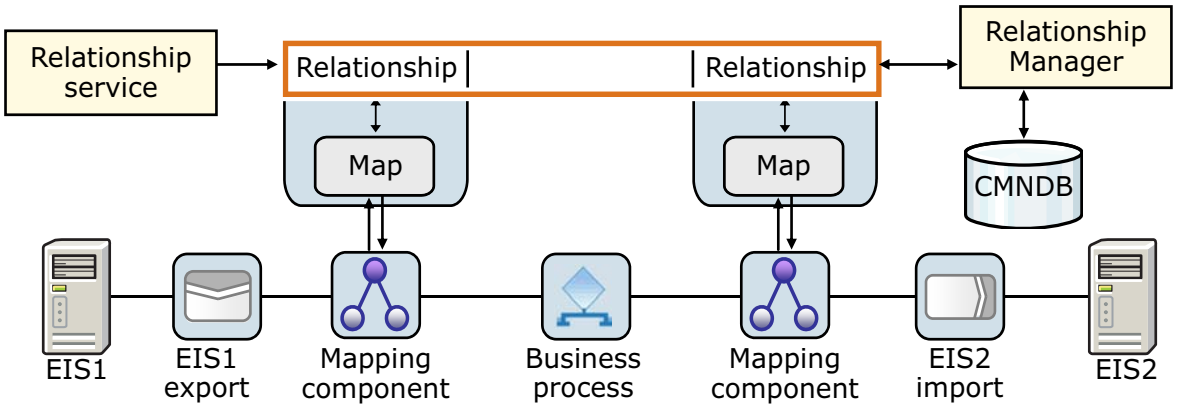
Figure 10-21. Runtime aspects of relationships

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Runtime components of relationships

- ***Relationship instance*** is the instantiation of a relationship definition at run time
- ***Role instance*** is the instantiation of a role definition at run time
- ***Relationship service*** allows relationships and roles to be explicitly represented and persisted
- ***Relationship Manager*** is the web tool that is used to create, manipulate, and view relationship runtime instances



Relationships

© Copyright IBM Corporation 2017

Figure 10-22. Runtime components of relationships

The Relationship Manager is used to view relationships and roles that are established for your data. As part of the relationship service that is responsible for running relationships, it stores this information in an internal database (CMNDB). The Relationship Manager then provides an interface for looking at that relationship information.

© Copyright IBM Corp. 2017

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

IBM Training

Relationship data constituents (1 of 2)

- At run time, dynamic relationships are stored in a series of role tables in the IBM Process Server database
  - Created for each (non-managed) role in a relationship definition
  - Store the unique (key) data from the particular business entity
  - Store the relationship instance ID, state information, and time stamps
- Static relationship data is also stored in the IBM Process Server database
- The Relationship Manager is used to view the data in the tables
- The relationship schema also contains stored procedures
  - Interact with the role tables based on the calling context
  - Perform RETRIEVE, INSERT, UPDATE, and DELETE statements
- Counters or sequences are also stored
  - Generate unique values for the relationship instances (the generic or managed object ID)

Relationships

© Copyright IBM Corporation 2017

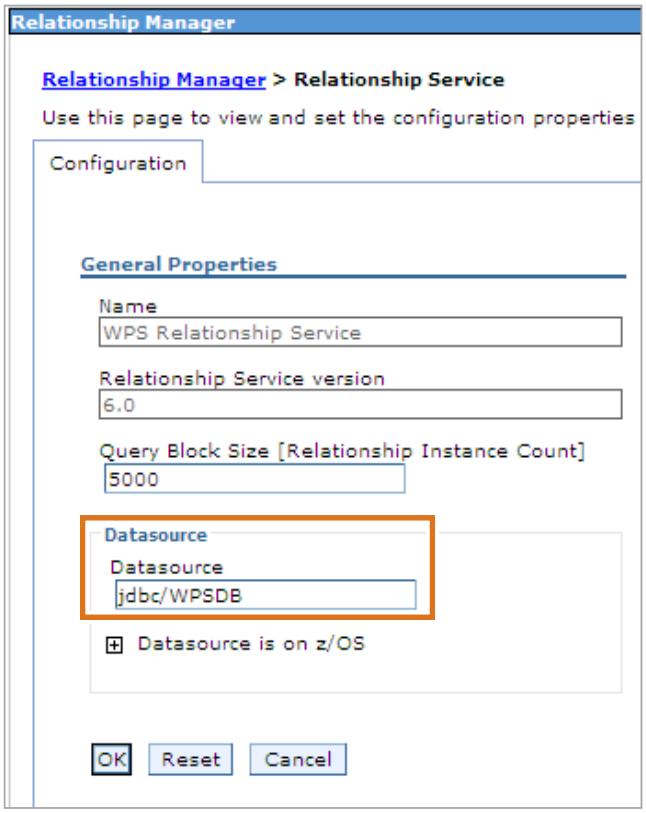
Figure 10-23. Relationship data constituents (1 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Relationship data constituents (2 of 2)



- It is possible to dedicate a database to relationship data
  - Allows for separate backup and recovery strategies
- Database location is specified in the Process Server administration console
  - Define a new data source in the JDBC Providers section
  - Use the Relationship Manager to update the database location

Relationships

© Copyright IBM Corporation 2017

Figure 10-24. Relationship data constituents (2 of 2)

You can designate a specific database that can be used exclusively for storing relationship data, which is done through the Relationship Manager. Doing so allows more control of the characteristics of the database, and flexibility for data maintenance (backup, restore, and other maintenance operations).

© Copyright IBM Corp. 2017

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

IBM Training

Relationships: Calling context and verbs

- The relationship service manages the relationships that are based on the calling context, the mode in which the relationship is executing (passed to the relationship by the mapping service)
  - It specifies whether the relationship is being invoked during a service request, or during a service response
- There are four possible calling contexts:
  - **EVENT_DELIVERY**
  - **SERVICE_CALL_REQUEST**
  - **SERVICE_CALL_RESPONSE**
  - **SERVICE_CALL_FAILURE**
- Relationship maintenance relies heavily on the verb in the business graph or change summary
  - For example, if a business object is sent with a Delete verb, the corresponding relationship instance must also be deleted
  - The verb indicates the operation on a specific business entity
  - Relationships support Create, Update, Delete, and UpdateWithDelete verbs

Relationships © Copyright IBM Corporation 2017

Figure 10-25. Relationships: Calling context and verbs

The relationship service maintains the relationships that are based on the calling context:

- EVENT_DELIVERY is a one-way invocation; that is, the map (and the relationship) is started on the request path with no response expected.
- SERVICE_CALL_REQUEST is a request invocation where the map (and the relationship) is started on the request path and a response might be expected.
- SERVICE_CALL_RESPONSE is a response invocation where the map (and the relationship) is started on the response path with a completion status.
- SERVICE_CALL_FAILURE is a response invocation where the map (and the relationship) is started on the response path with a failure status.

The *verb* indicates the operation (being) run on a specific business entity and is:

- Retrieved from the verb field of the business graph (after-image business objects)
- Obtained from the change summary of a particular (delta) business object

The relationship service gets the verb from the business graph first, or from the change summary record. The change summary must exist on a business object (top-level or child) whose identity relationship is being maintained.

© Copyright IBM Corp. 2017

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

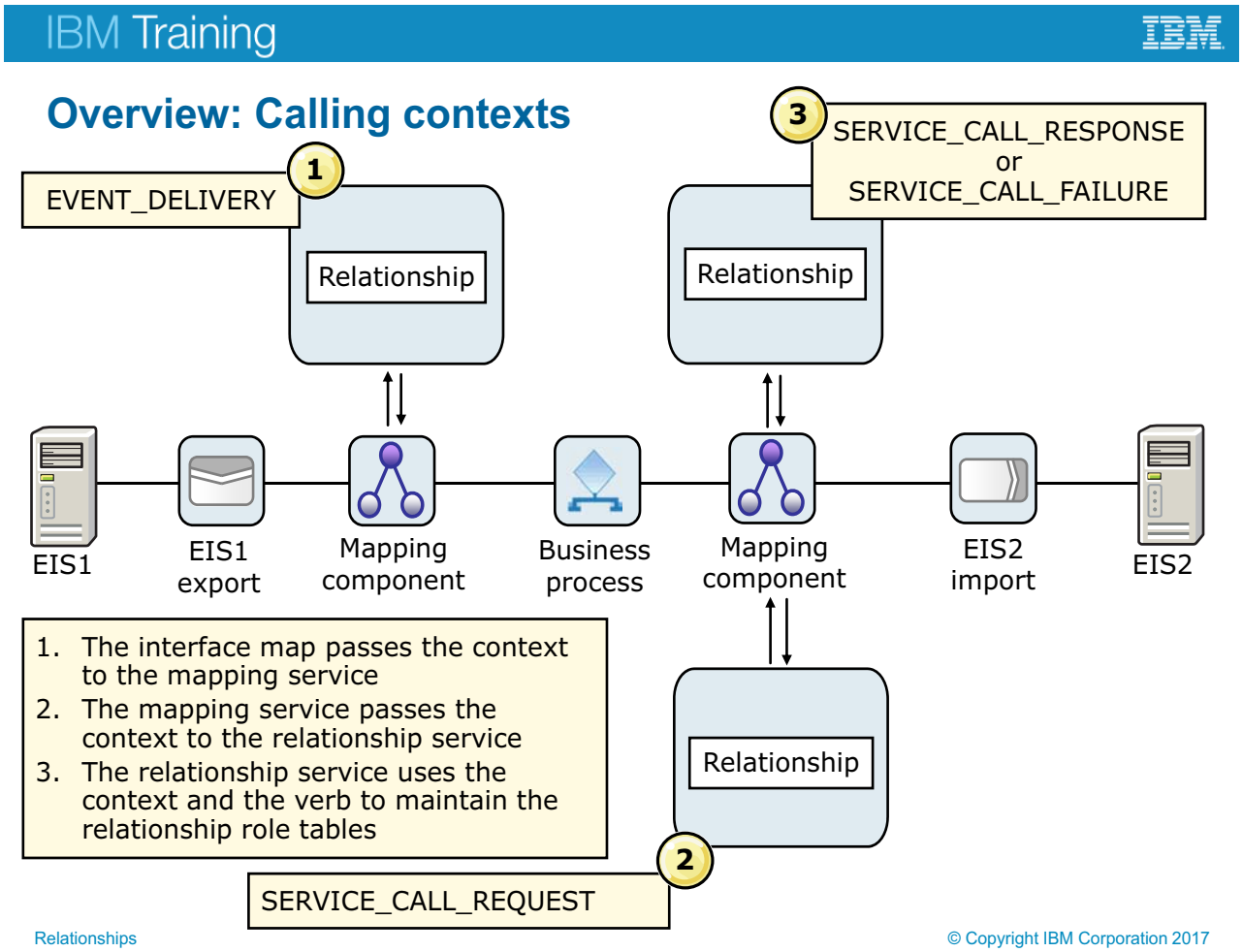


Figure 10-26. Overview: Calling contexts

This slide is an overview of the calling contexts that were described in the previous visual.

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

10.5. Managing relationships

© Copyright IBM Corp. 2017

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

IBM Training

Managing relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

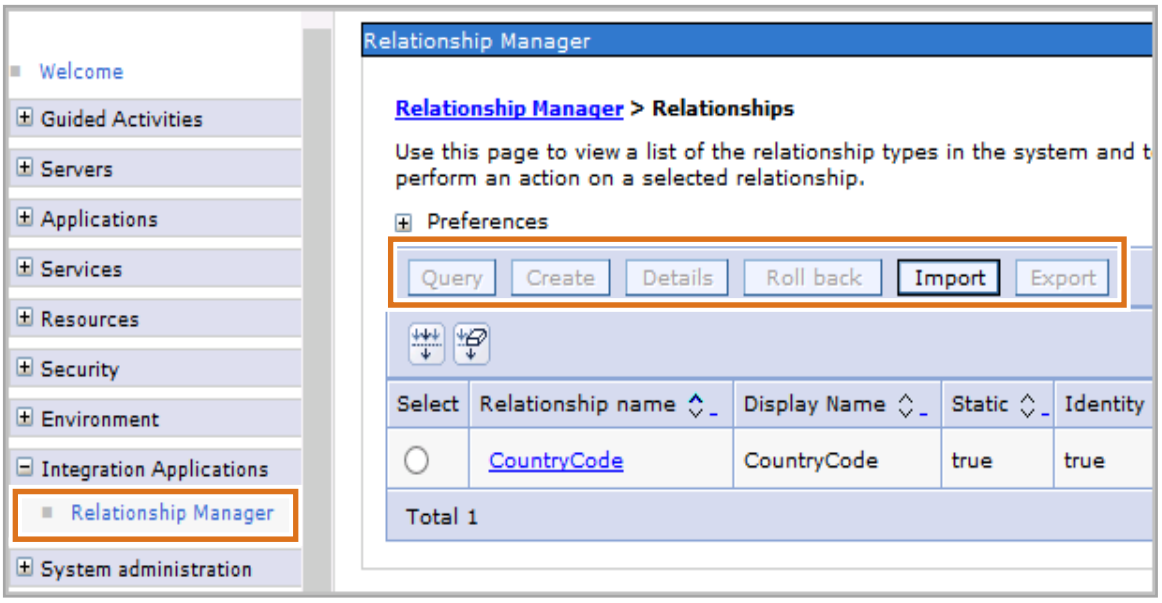
Figure 10-27. Managing relationships

© Copyright IBM Corp. 2017

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

IBM Training

Relationship Manager



You can query, create, view details, roll back relationships, import, and export role and relationship instance data

Relationships

© Copyright IBM Corporation 2017

Figure 10-28. Relationship Manager

The Relationship Manager is accessed from the administrative console as an integration application. You can use the Relationship Manager to:

- Browse and inspect the values for existing relationships
- Create and delete relationship instances
- For static relationships, import relationship instance data from or export it to a file of comma-separated values (CSV)
- Modify the contents of a relationship instance, such as adding and deleting role instances
- Edit the data of a relationship role instance, for example, role properties and logical state
- Activate and deactivate role instances
- Query role instances, providing the key attribute, start and end dates, and property value
- Salvage a situation when problems arise to roll back corrupted relationship data to a point in time when you know that the data is reliable.

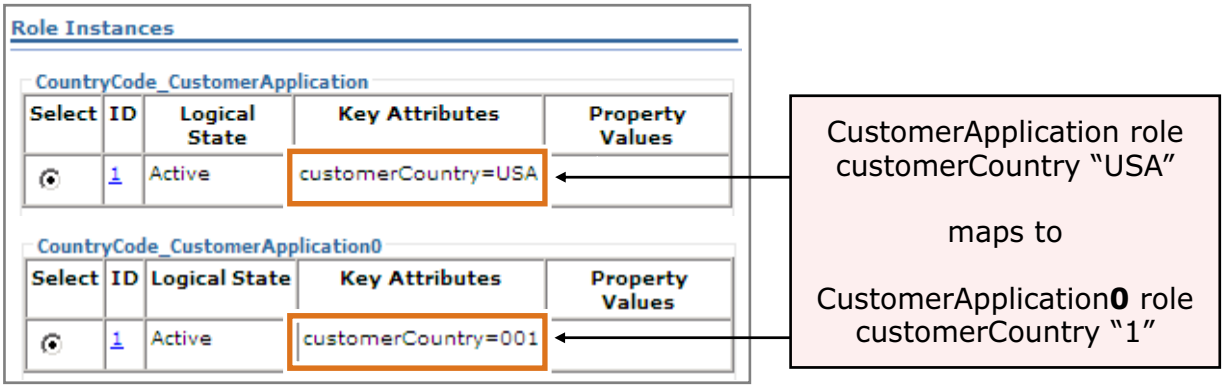
For example, you might have to move a large amount of data from one system to another in the evening with established relationships; and the next night, those relationships are no longer applicable. In this case, you can roll back those relationships, undo the capture type relationships, and capture new ones.

© Copyright IBM Corp. 2017

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

IBM Training

Instance creation through Relationship Manager



Relationships

© Copyright IBM Corporation 2017

Figure 10-29. Instance creation through Relationship Manager

To create instance data for a relationship, select the relationship that you want, and click **Create** (for static relationships, you can also do this task in the **Properties** tab of the relationship).

In this example, relationship CountryCode contains two roles: countryCode_CustomerApplication and countryCode_CustomerApplication**0**.

The example shows that the customerCountry value for the countryCode_CustomerApplication role is **US**, while the corresponding customerCountry value for the countryCode_CustomerApplication**0** role is **1**.

© Copyright IBM Corp. 2017

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

IBM Training

Populating the relationship instance table by using database views

- As part of application installation that contains relationships, database views are created that are associated with the relationship runtime tables
  - Each relationship instance table has an SQL view that is created
  - One view is created for each relationship role in the Process Server common database
  - You can query relationship data details with the view, or run the SQL statement to manipulate relationship instance data directly
  - The mapping between the view name and its corresponding relationship role is captured in a table that is called **RELN_VIEW_META_T** in the Process Server common database
  - Each view name is prefixed with letter **V**
  - Then, the partial relationship name follows **V**, partial role name, plus 3 digits of the unique identifier

    Example: **`V_ID_CLARIFYCUSTOMER_098`**

Note: The Derby database does not support updatable views

Relationships

© Copyright IBM Corporation 2017

Figure 10-30. Populating the relationship instance table by using database views

Another way to query and update relationship instance data is through database views. When you create a relationship database table, a corresponding SQL view is automatically created. These views are essentially encapsulations of the relationship data that is stored in database tables. You can use these views to populate, query relationship data, or both by using:

- SQL statements with a DB client
- JDBC to run SQL statements with a Java program

In either case, you can use the SQL views in the same manner as you would for tables. You can use this technique as another method to the Relationship Manager application to directly populate large sets of application-specific data by using SQL statements in your relationship databases. You can also use this technique to import data from a flat-text file into a database table.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe the purpose and business value of static and dynamic relationships
- Describe the runtime components of relationships
- Explain the deployment of SCA applications with relationship components
- Explain how to use the Relationship Manager tool to manage runtime relationship instances
- Describe the methods of predefining relationship tables with existing data
- Explain the use of composite keys in relationships
- Explain when foreign key cross-referencing or lookup is required

Relationships

© Copyright IBM Corporation 2017

Figure 10-31. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. Which of the following list is not a calling context?
   A. EVENT_DELIVERY
   B. SERVICE_CALL_REQUEST
   C. SERVICE_CALL_REPONSE
   D. SERVICE_CALL_FAULT
2. True or False: A managed role in an identity relationship has no key value.
3. True or False: You cannot implement any type of relationship by using an XML/XSLT data map.

Relationships

© Copyright IBM Corporation 2017

Figure 10-32. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. Which of the following list is not a calling context?
   A. EVENT_DELIVERY
   B. SERVICE_CALL_REQUEST
   C. SERVICE_CALL_REPONSE
   D. SERVICE_CALL_FAULT

   The answer is D.

2. True or False: A managed role in an identity relationship has no key value.
   The answer is False. The managed role represents the generic business object and its key.

3. True or False: You cannot implement any type of relationship by using an XML/XSLT data map.
   The answer is True.

Relationships

© Copyright IBM Corporation 2017

Figure 10-33. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Working with static relationships

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 10-34. Exercise 9

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

IBM Training

Exercise objectives

- Implement a static relationship to map corresponding values between business object fields
- Import static relationship data from a comma-separated value file
- Test a relationship in the IBM Integration Designer integrated test environment

Relationships

© Copyright IBM Corporation 2016

Figure 10-35. Exercise objectives

© Copyright IBM Corp. 2017

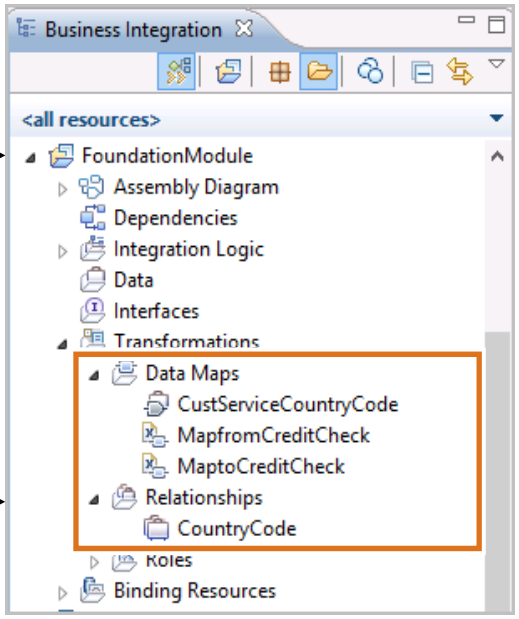
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 9

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CustomerApplication** business object
3. **AccountVerification** BPEL



New components that you create in the lab:

1. **CountryCode** static relationship
2. **CustServiceCountryCode** BO map

Relationships

© Copyright IBM Corporation 2017

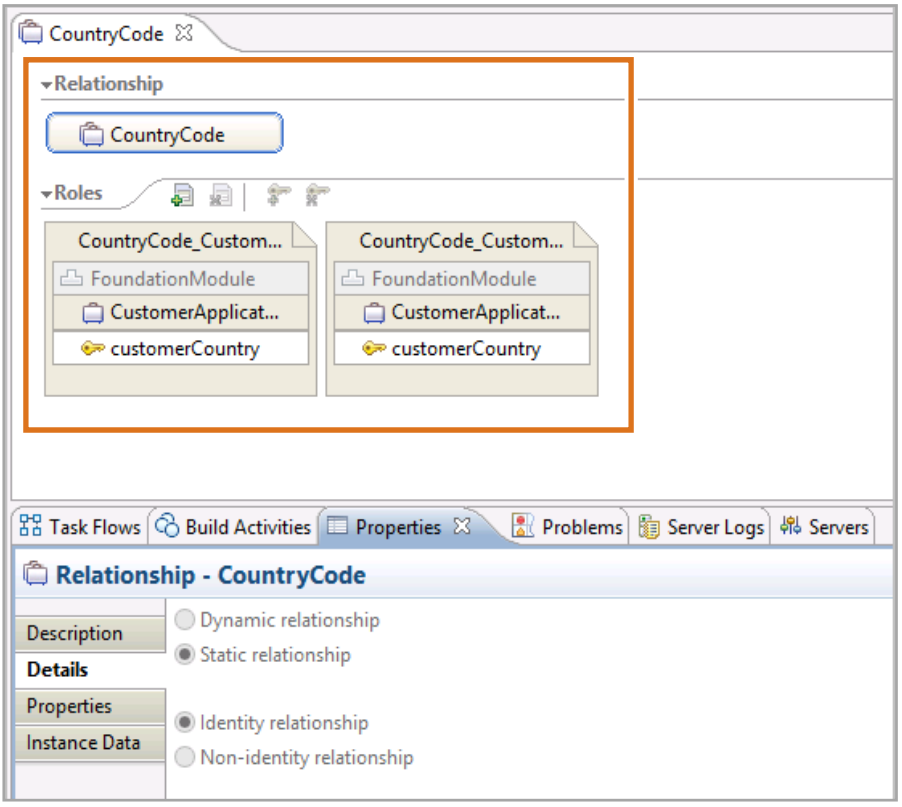
Figure 10-36. Components that are required for Exercise 9

© Copyright IBM Corp. 2017

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

IBM Training

Create the CountryCode static relationship in Exercise 9



© Copyright IBM Corporation 2017

Figure 10-37. Create the CountryCode static relationship in Exercise 9

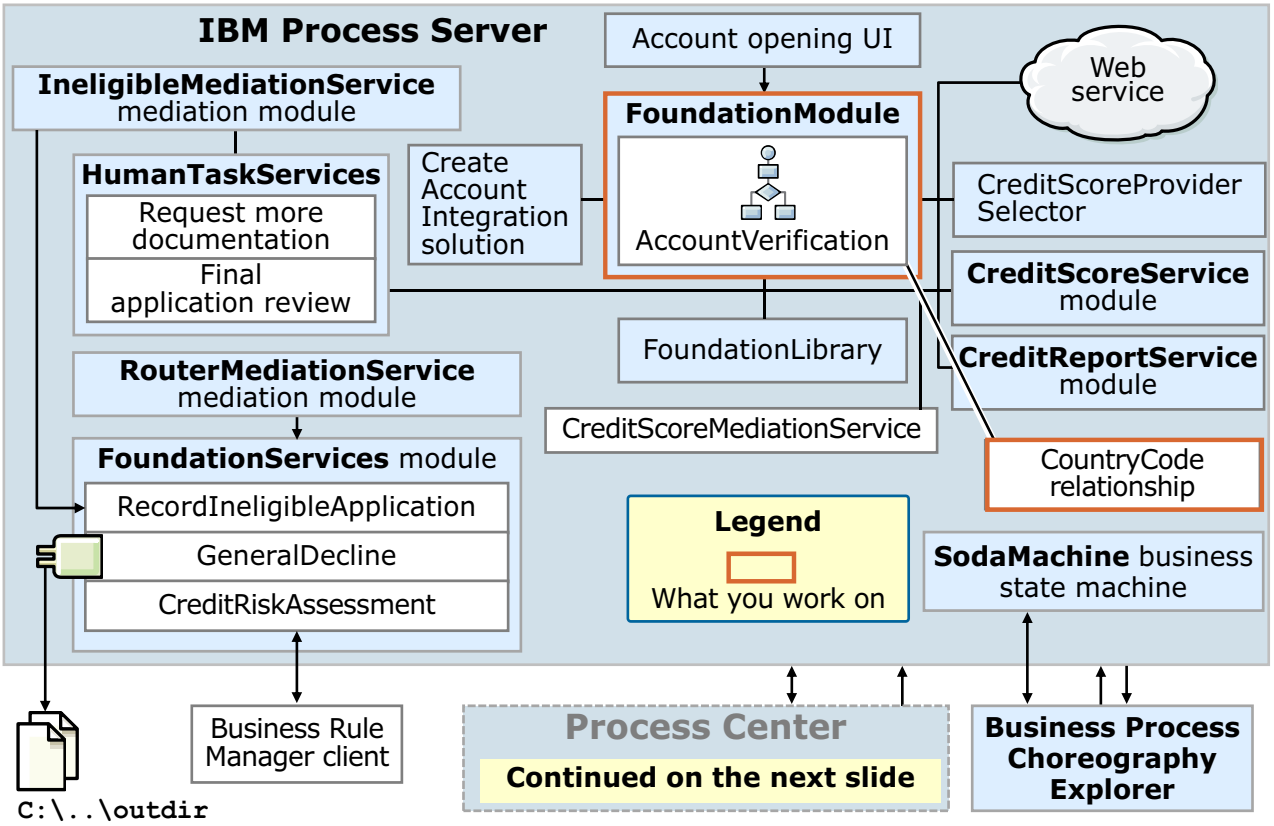
In this exercise, you test the module by using a customer that has a high credit risk. Because of this risk, the application is declined. The process flow takes the “Application is declined” path, which invokes the Generate Decline and Record Declined Application activities. The CountryCodeMap activity is then invoked to translate the country code that the DeclinedApplicationToMQ activity follows. The DeclinedApplicationToMQ activity sends the message to a WebSphere MQ queue.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 9 (1 of 2)



Relationships

© Copyright IBM Corporation 2017

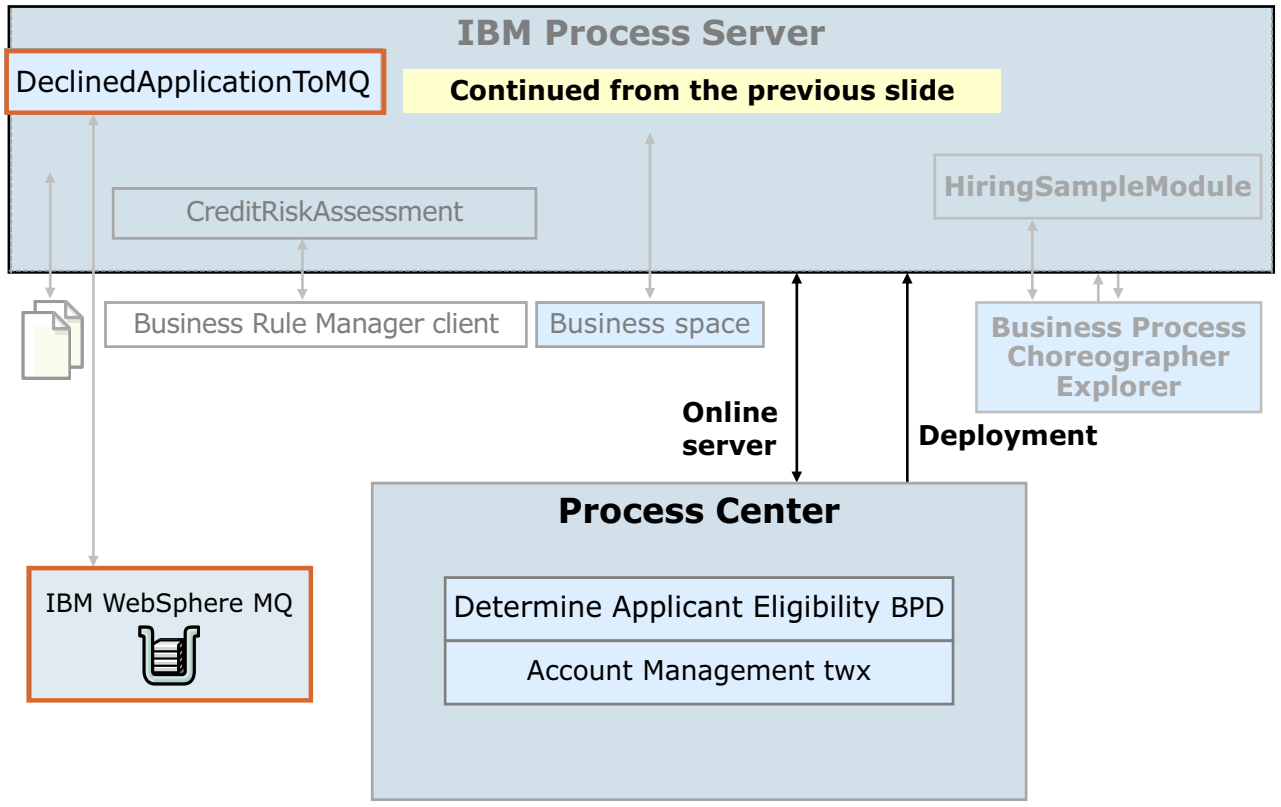
Figure 10-38. Course business scenario: What you work with in Ex 9 (1 of 2)

In this lab, you work with IBM Integration Designer.

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

IBM Training

Course business scenario: What you work with in Ex 9 (2 of 2)



Relationships

© Copyright IBM Corporation 2017

Figure 10-39. Course business scenario: What you work with in Ex 9 (2 of 2)

In this lab, you work with both the IBM Integration Designer and IBM WebSphere MQ. You do not work with Process Center.

© Copyright IBM Corp. 2017

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

Unit 11. Mediation primitives

Estimated time

01:30

Overview

This unit describes the mediation primitives in IBM Process Server that are not covered in the prerequisite courses.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe the role of mediation services in applications
- Describe the prebuilt mediation primitives that are available in IBM Integration Designer

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Miscellaneous mediation primitives
- Message validation and logging primitives
- Starting services statically and aggregating messages

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-2. Topics

© Copyright IBM Corp. 2017

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

11.1. Miscellaneous mediation primitives

© Copyright IBM Corp. 2017

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

IBM Training

Miscellaneous mediation primitives

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 11-3. Miscellaneous mediation primitives

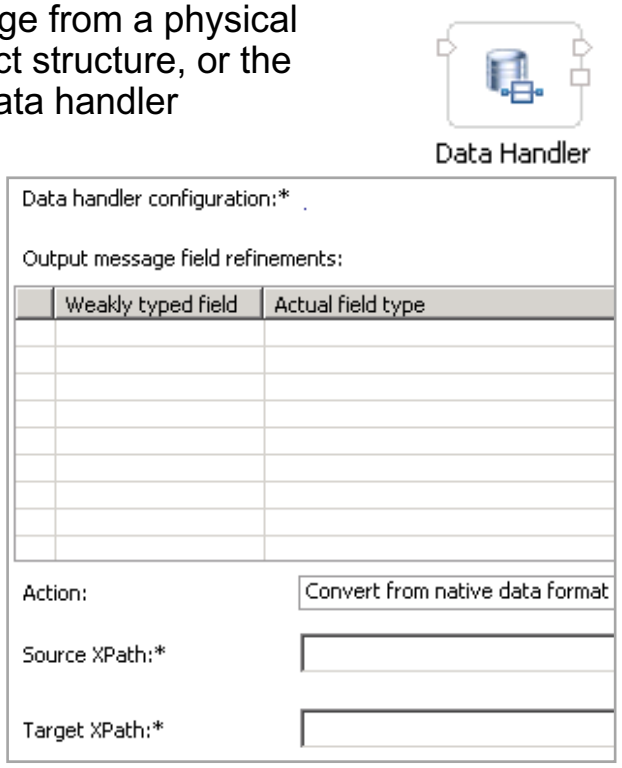
© Copyright IBM Corp. 2017

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

IBM Training

Data handler primitive

- Converts an element of a message from a physical format to a logical business object structure, or the reverse, by using a predefined data handler
- The data handler allows the transformation of the physical format into the specific business object in the mediation flow component, instead of the export and import
- The data handler primitive can change any part of the service message object



Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-4. Data handler primitive

Data handlers are used when you need a specific data format transformation that is not in the standard set of prepackaged data format transformations. In effect, you can use data handlers to customize a data format transformation for your particular needs. Data handlers are reusable transformation logic independent of a specific transport protocol, which means you can create a data handler and use it with JMS, WebSphere MQ, HTTP, EIS, and other bindings. Data handlers can be started from data bindings and Java components. When you are constructing a mediation flow for IBM Process Server, you can start a data handler by using the data handler mediation primitive.

Predefined data handlers that are provided with the product include:

- Comma-delimited
- Fixed width
- JAX-WS based EJB
- JSON (UTF8JSON)
- SOAP
- XML
- XML (UTF8XML)

© Copyright IBM Corp. 2017

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

- Atom feed

In addition to the predefined data handlers, you can write your own data handler in Java.

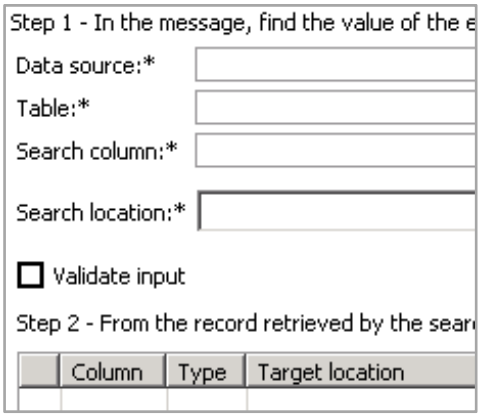
© Copyright IBM Corp. 2017

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

IBM Training

Database lookup primitive

- Alters the SMO with data that is retrieved from a relational database table
- If no matching record is found in database, the message is propagated through the `keyNotFound` terminal
- Only one database table can be read from, and only one column value can be used in the retrieval
- The database, data source, and any server authentication settings must be set up through the administration console



© Copyright IBM Corporation 2017

Figure 11-5. Database lookup primitive

You can use the database lookup primitive to enrich a message by using content that is retrieved from a database table.

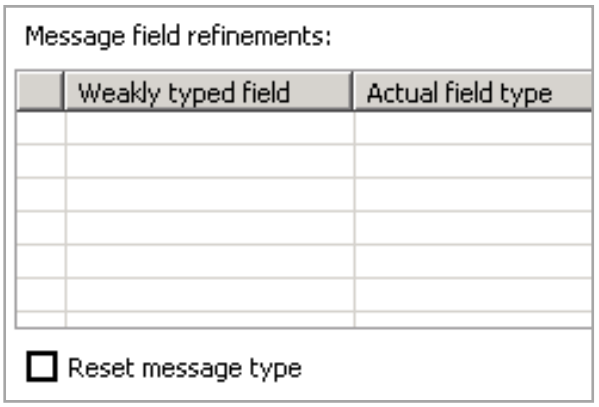
© Copyright IBM Corp. 2017

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

IBM Training

Set message type primitive

- You can treat weakly typed message fields as though they are strongly typed
- At run time, the set message type primitive does not affect the real structure or content of a message



© Copyright IBM Corporation 2017

Figure 11-6. Set message type primitive

You can use the set message type primitive to overlay message fields to act as an “alias.” This alias provides a way of logically redefining message fields, by overlaying message fields with more detailed structures, and then uses those more detailed structures in other mediation primitives. It does not affect the content or structure of the message itself.

You specify:

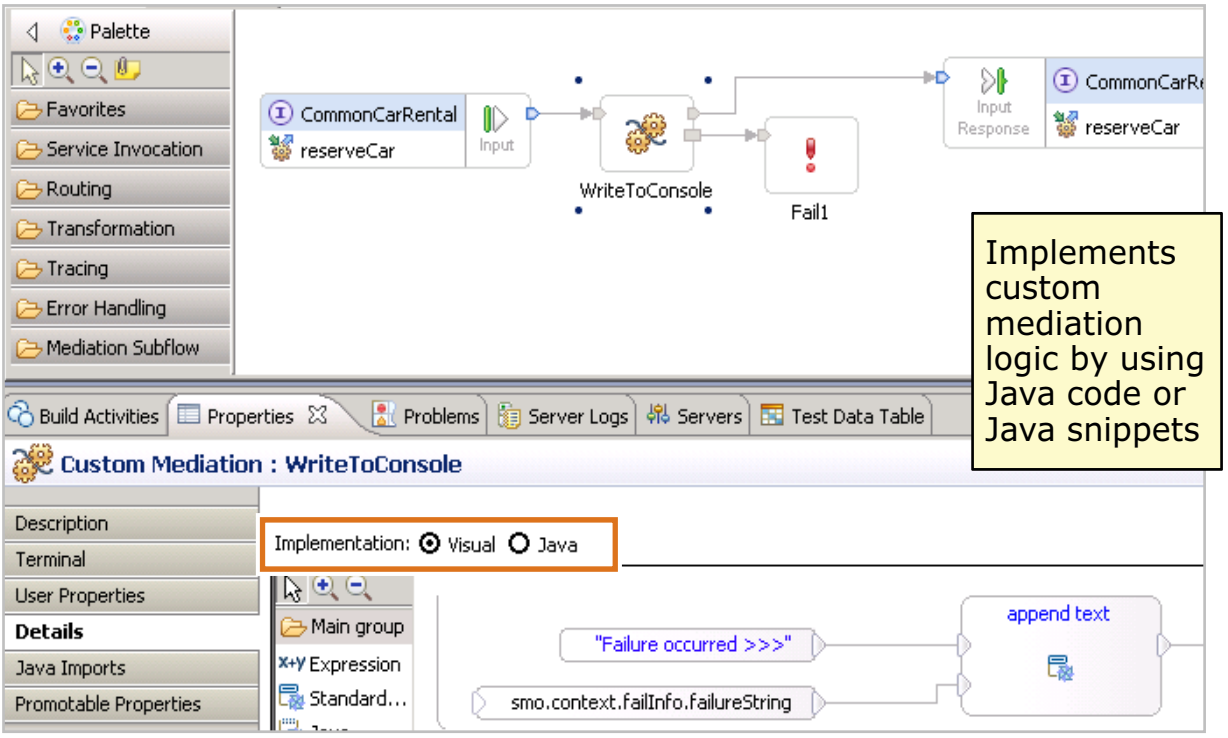
- Which fields in the message are refined with more specific typing information, and the data type to use for them
- A *reset message type* flag: when “true,” it causes the current primitive to “forget” any message field refinements that occurred in previous set message type primitives in the mediation

© Copyright IBM Corp. 2017

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

IBM Training

Custom mediation primitive



Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-7. Custom mediation primitive

You can use the custom mediation primitive to write your own Java code to manipulate the SMO. This primitive is useful when no other primitive provides the needed ability.

You can define more input and output terminals and properties on the primitive at development time for increased flexibility.

To implement the custom logic, you must:

- Create the Java code, either in Java snippet form, or in visual snippet form.
- Create more terminals on the primitive, if any.
- Set the message types for all terminals. IBM Integration Designer does not assist with terminal message types when using the custom mediation primitive.
- Declare any Java imports that the code requires.

© Copyright IBM Corp. 2017

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

IBM Training

Custom mediation: Error handling

- A *MediationRuntimeException* is thrown if
  - No operation or service reference is specified in the properties
  - No matching reference exists on the mediation flow component
- *MediationRuntimeException* and fail terminal behavior, note the difference with custom mediation primitives:
  - Unlike most MediationRuntimeExceptions, these exceptions fire the *fail* terminal if it is wired
  - If the reference on the mediation flow component is not wired
  - If custom code returns **`null`**, then it must return a DataObject
  - Beware: Default code is generated with "**`returns null;`**"
  - You must change the custom code when you configure the custom mediation primitive
- Any exception that the custom Java code throws:
  - Is thrown as is; no **`MediationXxxxxException`** type wraps it
  - If the fail terminal is wired, it is fired

© Copyright IBM Corporation 2017

Figure 11-8. Custom mediation: Error handling

When using the custom mediation primitive, you must be aware of the various types of exceptions that can arise. The content of the message that is propagated through the Fail terminal of the custom mediation primitive can also be different from messages that are propagated through the Fail terminal of other primitives.

© Copyright IBM Corp. 2017

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

11.2. Message validation and logging primitives

© Copyright IBM Corp. 2017

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

IBM Training

Message validation and logging primitives

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 11-9. Message validation and logging primitives

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Message validator primitive

- Examines the incoming message to determine whether it is valid
- "Valid" means from an XML type perspective; also, validates any weakly typed message fields that are set to strongly typed message fields earlier in the mediation flow
- If validation succeeds, the out terminal propagates the validated message
- If a validation exception occurs, the fail terminal is fired, and stores the exception information in the `FailInfo` element of the service message object (SMO)
- **Root** property specifies the element of the SMO that is going to be validated
  - Can be a specific element, **`/root`**, **`/body`**, and other elements
  - Note: The entire inbound message is not validated, unless that is specified in the **Root** property field

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-10. Message validator primitive

The message validator primitive is used to determine whether an incoming message is composed of valid XML types. It can be used to prevent later runtime exceptions within a mediation flow by screening out invalid messages.

A sample validation failure message is as follows:

```
CWSXM3802E: The type at /body/operation1/input1/float is 'string', while the
asserted type states it should be 'float'.
```

© Copyright IBM Corp. 2017

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

IBM Training

Event emitter primitive

- Sends out (emits) an event to the Common Event Infrastructure server
- “Event” refers to a business event, such as might cause a mediation flow failure or a mediation to take an unusual branch
- Events are reported in Common Base Event format
- You can include or exclude the inbound message in the emitted message

© Copyright IBM Corporation 2017

Figure 11-11. Event emitter primitive

The event emitter primitive is part of the Common Event Infrastructure (CEI), which is a larger environment for handling events. The Common Event Infrastructure provides a framework to capture events and publish them for event consumers. The event emitter primitive provides a means to generate an event message from within a mediation flow to send to the CEI server.

A good practice is to use the event emitter primitive to indicate an unusual condition or event to reduce the performance impact. When you place an event emitter in a flow, consider the possible number of events that can be generated, and the size of the messages that are emitted and stored. Placing an event emitter in a “normal” flow path generates much more event traffic than one placed in a branch that is run rarely. You must also consider whether it is necessary to store the entire message content, as opposed to significant data, to reduce the message size.

One of the properties of the event emitter is **transaction mode**. If you use the event emitter primitive, be sure that the transaction mode is set correctly for your application. If a rollback occurs and the event emitter is part of that transactional unit, the event that it emits is also rolled back, depending on the value of this property.

© Copyright IBM Corp. 2017

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

IBM Training

Message logger primitive

- Stores a message in a relational database table
- Can also write a message to another medium by using the custom logging facility
- Saves selectable amount of message content plus identifying information (time stamp, message identifier, primitive and module names, and other information)
- Messages can be logged for later review, post-processing, auditing, or whatever purposes you require
- Original inbound message passes through without alteration

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-12. Message logger primitive

The message logger primitive stores a copy of a message in a relational database table or other storage medium.

The message logger primitive contains a number of configurable properties to give you flexibility and control in logging messages, including:

- **Enabled**: Determines whether the logger does its function. It allows logging to be disabled without removing the primitive. It is a promotable property, and the administrator can set it.
- **Root**: XPath expression that describes the message to log. The message to log is converted to XML from the point that the root specifies.
- **Transaction mode**: Determines whether the message logger participates in mediation transaction or independently.
- **Logging type**: The destination type for the logging event: database or custom (uses the custom log facility to write to a flat file, for example).
- **Data source name**: The JNDI name of the data source that defines where the data is logged. The default value points to a database named **CommonDB**.
- **Handler**, **Formatter**, **Filter**, **Literal**, **Level**: Implementation classes that are used when custom logs are enabled.

© Copyright IBM Corp. 2017

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

By default, messages are written to a data source named CommonDB, to a table named MSGLOG.

© Copyright IBM Corp. 2017

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

IBM Training

Trace primitive

- Writes a message to the server log, user trace log file, or other file
- Can be used for debugging or tracing a mediation flow
- Saves selectable amount of message content plus identifying information (time stamp, message identifier, primitive and module names, and other information)
- Original inbound message passes through without alteration

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-13. Trace primitive

The trace primitive is similar to the message logger primitive, but it writes its output to the server log, a trace log file, or another file. Its configurable properties are similar to those properties of the message logger primitive.

© Copyright IBM Corp. 2017

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

11.3. Starting services statically and aggregating messages

© Copyright IBM Corp. 2017

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

IBM Training

Starting services statically and aggregating messages

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

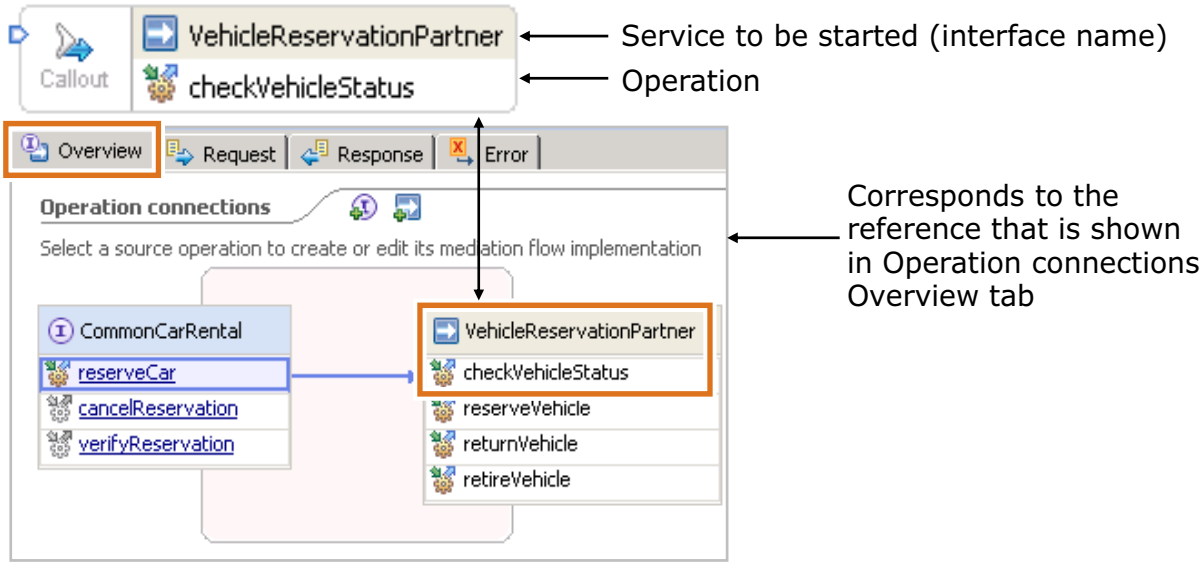
Figure 11-14. Starting services statically and aggregating messages

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

IBM Training

Callout node

- Endpoint in a request flow
- Sends message to ("invokes") requested service and operation
- Has *in* terminal, but no *out* terminal



© Copyright IBM Corporation 2017

Figure 11-15. Callout node

The callout node is one means to start services in a mediation flow. Additionally, you can dynamically alter the endpoint information that is used to start services, so you can determine which service to start dynamically, while the mediation is running.

The callout node is shown in the request flow of a mediation. It corresponds to the reference that is shown in the Operation connections view. The callout node is not a true mediation primitive. It is the means by which a mediation flow starts a service at the end of a request flow. The mediation flow transforms the message in whatever way is required; at the end, the message is passed to the callout node, which starts the service. Some of the configurable properties of the callout node include:

- **Dynamic endpoint**: Callout starts the endpoint that is set in the *target* element of the SMOHeader in the message.
- **Timeout values**: Requires the callout node to wait for a response indefinitely, or for a fixed duration. If the timeout threshold is exceeded, the fail terminal is fired, and it is treated as an unmodeled fault.
- **Require mediation flow to wait for service response**: Allows an asynchronous invocation to act like a synchronous invocation.
- **Retry**: You can specify the number of attempts to try again in response to a callout failure.

© Copyright IBM Corp. 2017

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

IBM Training

Service invoke primitive

- Like callout node, but can be used anywhere in request flow
- Sends message to ("invokes") requested service and operation
- Has *in* and *out* terminals for inbound and outbound messages
- Has *timeout* terminal that is fired when timeout threshold is exceeded, like callout node
- Has a *<fault message name>* terminal for each modeled fault
  - These terminals propagate the modeled message that is returned from the invocation when a modeled fault occurs

Sample flow-containing service invoke

© Copyright IBM Corporation 2017

Figure 11-16. Service invoke primitive

The service invoke primitive does the same role that the callout node does, except that the service invoke can occur anywhere in a mediation flow, not just at the end (as with the callout node). Otherwise, the properties of the two are identical.

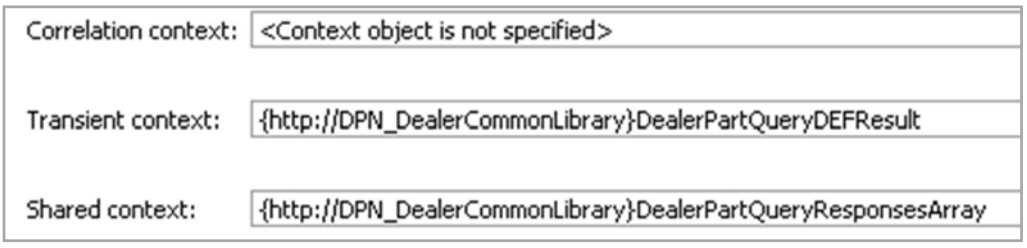
© Copyright IBM Corp. 2017

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

IBM Training

Service message object contexts

- The service message object (SMO) contains system-defined context objects and user-defined context objects
- Four specific user-defined context objects:
  - User context
  - Correlation context
  - Transient context
  - Shared context
- Define structure of user-defined context objects by specifying business objects on the input node of the mediation flow (except for user context, which is set on the export)



Mediation primitives © Copyright IBM Corporation 2017

Figure 11-17. Service message object contexts

The service message object contains:

- The message transport headers.
- Context information that is used during the lifetime of the SMO as it flows through a mediation.
- The message data itself, the body of the message (sometimes called the payload). The body is not required; it is possible for an SMO to contain only headers and context information.

The service message object **context** allows mediation primitives to pass data that is not part of the message payload between themselves. The contexts are used for passing data between the request and response flows, or within a request or response flow. You can define the structure of context elements. The business objects that are used to format the correlation, transient, and shared contexts are set on the input node. There are four user-defined context objects:

- **User context**: Is used to pass data between SCA components.
- **Correlation context**: Is used to pass values from request flow to response flow.
- **Transient context**: Is used to pass values within a request flow or within a response flow; cannot link between request and response flows.
- **Shared context**: One for request flow, and a separate one for response flow (one shared context per thread, per flow). Shared context is used with fan-out and fan-in primitives to store

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

values that are associated with operations that occur between the two primitives (an “aggregation block”). Do not write to shared context outside a pairing of fan-out and fan-in (but it is permissible to read from it).

© Copyright IBM Corp. 2017

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

IBM Training

Fan-out and fan-in primitives: Overview

- Allows splitting and aggregating messages within a flow
  - Fan-out (splitting) can be used alone or with fan-in; for example, to send the same message on multiple flow paths
  - Fan-in (aggregating) can be used only with a preceding fan-out; for example, to combine responses from multiple paths or iterations
- Fan-out has two modes of operation:
  - Default mode
  - Iterate mode
- A fan-in must be associated with a specific fan-out instance
- Usage examples:
  - Broadcasting of messages for one-way service: sending a separate message for each occurrence of a repeating element in a message
  - Aggregating responses from multiple (“parallel”) service invocations

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-18. Fan-out and fan-in primitives: Overview

You can use the fan-out and fan-in primitives to:

- Send a message through multiple paths within a mediation flow, and then “reassemble” the messages into one
- Repetitively send the elements of a repeating group in a message through a path in a mediation flow

© Copyright IBM Corp. 2017

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

IBM Training

Fan-out primitive (1 of 2)

Default operating mode:

- The out terminal is fired one time for each input message, for each output wire that is connected to it
- The same message is propagated to each output wire, in succession
- The output message is identical to the input message

Iterate operating mode:

- The output terminal is fired one time for each occurrence of a repeating element in the incoming message
- Each time the terminal is fired, the next occurrence of the repeated element is placed in the FanOutContext SMO element
- A message is propagated to downstream primitives until one of the following happens:
  - Fan-in primitive is encountered (assuming decision point was reached)
  - Callout is encountered
  - Stop, fail, or unwired out terminal is reached
- The next message is then fired through the out terminal

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-19. Fan-out primitive (1 of 2)

The two operating modes for the fan-out primitive are:

- **Default** mode:
  - The output terminal is fired one time for each input message.
  - The output message is identical to the input message.
- **Iterate** mode:
  - The output terminal is fired one time for each occurrence of the repeating element in the incoming message (which you specify with an XPath expression).
  - Each time the terminal is fired, the value of the current element is placed in the FanOutContext element.
  - Each occurrence of the repeating element is stored in a FanOutContext field of type anyType.

© Copyright IBM Corp. 2017

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

IBM Training

Fan-out primitive (2 of 2)

- *out* terminal:
  - In *default* mode, unmodified message is routed through the *out* terminal
  - In *iterate* mode, the out terminal propagates the input message; it is fired one time for each occurrence of the repeating element
- Message content in *iterate* mode:
  - Each time the terminal is fired, the value of the current element is placed in the FanOutContext
  - The outgoing message is the original message + the FanOutContext
- *noOccurrences* terminal:
  - Used only in *iterate* mode
  - Used when the input message does not contain any occurrences in the repeating element (as if primitive is set to *default* mode)

© Copyright IBM Corporation 2017

Figure 11-20. Fan-out primitive (2 of 2)

This visual shows how you set the operating mode for the fan-out, and what the implications are for the message content and the terminals of the fan-out primitive.

© Copyright IBM Corp. 2017

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

IBM Training

Fan-in primitive (1 of 2)

- Is used to aggregate messages that the corresponding fan-out primitive propagates
- Receives messages until a *decision point* is reached, and then the last message that is received is propagated to the output terminal
- Three types of decision points are supported:
  - **Count**: When a set number of messages are received, the fan-in primitive fires the output terminal
  - **XPath decision**: If an XPath evaluation of the input message evaluates to true, the fan-in primitive fires the output terminal
  - **Iterate**: The fan-in primitive waits to receive all of the messages that the corresponding fan-out primitive produces, and then fires the output terminal

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-21. Fan-in primitive (1 of 2)

The fan-in primitive acts as the endpoint in a message aggregation. It defines the end of an aggregation block. While a fan-out can be used alone (and typically is when used in iterate mode), a fan-in *must* always be paired with a corresponding fan-out. IBM Integration Designer prevents you from adding a fan-in to the canvas until you define which fan-out it is associated with. The job of the fan-in is to determine whether the “decision point” that terminates an aggregation activity was reached.

© Copyright IBM Corp. 2017

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

IBM Training

Fan-in primitive (2 of 2)

- Configuration of *fan-out* and the *fan-in* completion criteria must complement each other
- IBM Integration Designer requires you to name the corresponding *fan-out* primitive when *fan-in* is added to canvas
- In addition to *in*, *out*, and *fail* terminals, the *fan-in* primitive also has:
  - *stop* terminal (input): Causes the firing of the incomplete terminal that stops the associated *fan-out* primitive from sending any more messages
  - *incomplete* terminal (output): Works with *stop* terminal; is also fired when a timeout occurs
- You specify:
  - Decision type (count, XPath, iterate)
  - Timeout value: Causes the firing of the incomplete terminal if this value does not reach the decision point

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-22. Fan-in primitive (2 of 2)

This visual describes the operating mode for the fan-in, and what the implications are for the message content and the terminals of the fan-in primitive.

© Copyright IBM Corp. 2017

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

IBM Training

Aggregation and shared context

- Problem: How to aggregate data
  - Each time the fan-out output terminal is fired, a new SMO instance is created
  - Each new SMO instance is a deep copy
  - How to collect the relevant results for each iteration or flow between the fan-out and fan-in?
- Solution: Introduction of the shared context in the SMO
  - Each SMO instance references the shared context
  - A business object defines it (similar to transient and correlation contexts)
- Shared context usage
  - Flows between fan-out and fan-in set values that aggregates in the shared context
  - After the fan-in completes, subsequent primitives are used to build aggregated message
- No parallelism is allowed between a fan-out and fan-in
  - Service invoke primitives have the force sync property automatically set

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-23. Aggregation and shared context

When a mediation that contains an aggregation block is operating (the primitives encompassed between paired fan-out and fan-in primitives), a separate copy of the message is sent to each branch of the aggregation. A common use for aggregation is to send a message to multiple service invocations. You must track the data that is generated within the aggregation block; that is the role of the SMO shared context. Each SMO instance is a single memory area.

© Copyright IBM Corp. 2017

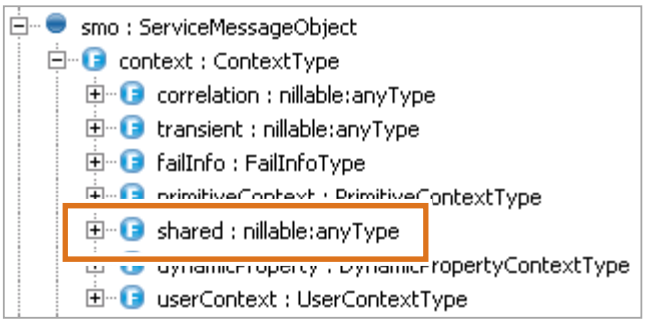
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

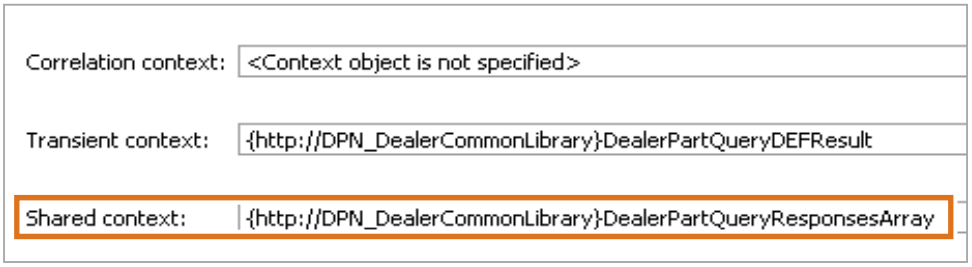
IBM Training

SMO structure: Context shared

Shared context:

- Use when operating within *fan-out* or *fan-in* primitives
  - Aggregate results of processing between the fan-out and fan-in
- Single memory area that all the SMO instances share
- Business object that is configured on the input node defines it





© Copyright IBM Corporation 2017

Figure 11-24. SMO structure: Context shared

The shared context is where data is placed in a mediation flow for fan-in primitives to access during an aggregation. Similar to the transient and correlation contexts, the shared context is a user-provided business object that is set at the input node of a mediation flow. The shared context is used when a mediation flow contains fan-out and fan-in primitives. It is used to aggregate the results of the fan-out operation.

- The shared context is a thread-based storage area. Generally, one request flow has one thread, even if the path splits. Therefore, generally, one request flow shares one shared context.
- The content of the shared context business object does *not* persist across a request and response flow, through callout invocation. Whatever data is in the shared context of the request flow, *cannot* be reused during the response flow.
- The shared context can be used to aggregate data when using the fan-out and fan-in mediation primitives: it is *not* intended for general data storage during a flow. Use the **correlation context** and **transient context** for general data storage.

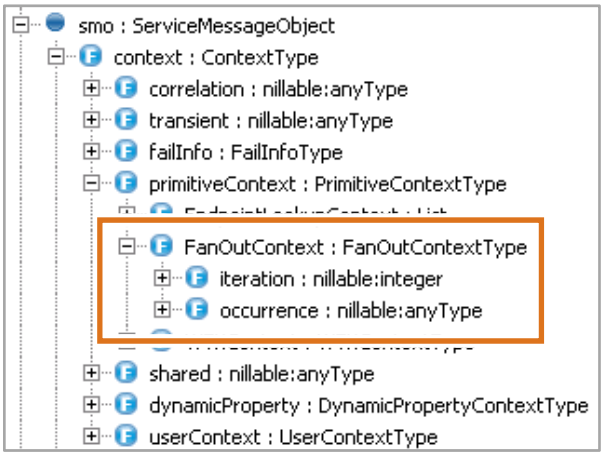
© Copyright IBM Corp. 2017

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

IBM Training

SMO structure: Context fan-out

- *FanOutContext*:
  - Used when processing a message in a fan-out primitive by using “iterate” mode
  - Contained within the *primitiveContext* in the SMO
- *FanOutContextType* contains:
  - **Iteration**: Indicates which element in the array of child objects of the BO (0 based)
  - **Occurrence**: Contains a single element of the array that is being iterated over



© Copyright IBM Corporation 2017

Figure 11-25. SMO structure: Context fan-out

The fan-out context is used only when a fan-out primitive is used in a mediation flow, and then only when the primitive is configured by using “iterate” mode.

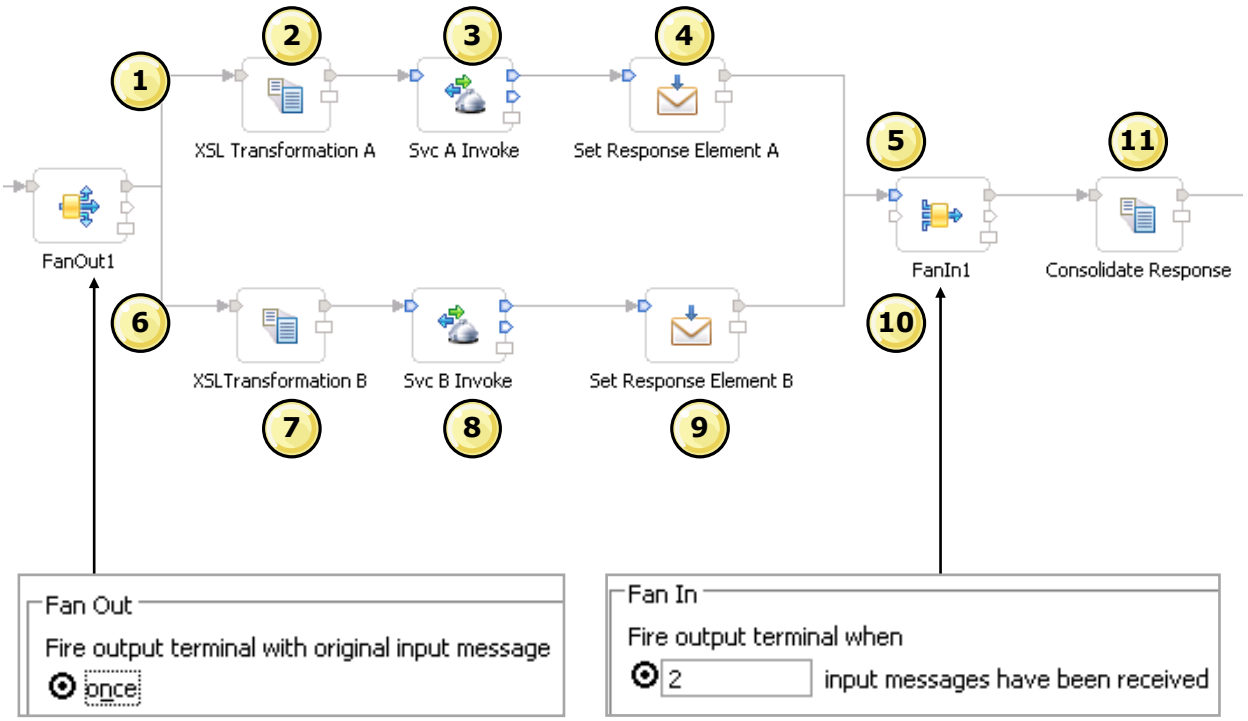
When a FanOutContext is present, it contains an iteration count that indicates which iteration of the SMO element is being processed. It also contains an occurrence that is the actual instance of data that is sent out from the fan-out primitive. The iteration count is zero based, which means the first iteration starts at zero. These values, like any part of the SMO, are accessible to your mediation flow. You can query or use these values within the flow.

© Copyright IBM Corp. 2017

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

IBM Training

Fan-out and fan-in: Aggregation example



Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-26. Fan-out and fan-in: Aggregation example

The following list is an example of an aggregation flow where two services are started.

- Step 1: The message flows into the *FanOut1* primitive. A copy of the message is sent to the *XSL Transformation A* primitive.
- Step 2: *XSL Transformation A* is an XSL transformation that prepares the business objects for the invocation of Service A (initializes values, and other functions).
- Step 3: *Svc A Invoke* is a service invoke primitive that calls Service A. The results from the call are propagated through the out terminal of the primitive.
- Step 4: *Set Response Element A* is a message element setter primitive that saves important information from the invocation of Service A. It is stored in the shared context of the SMO to retrieve after the aggregation block ends.
- Step 5: The *FanIn1* primitive determines whether the decision point was reached. In this case, it was not (only one input message was received), so control returns to the fan-out.
- Step 6: The *FanOut1* primitive sends another copy of the message to the next wire connected to the out terminal. This message is identical to the one sent on the other wire.
- Steps 7, 8, 9: The same processing occurs for the invocation of Service B as it did for Service A (initialization [7], service call [8], saving response values [9]).

© Copyright IBM Corp. 2017

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

- Step 10: The *FanIn1* primitive is reached again. This time, it determines that the decision point was reached (two input messages were received). The message passes to its out terminal.
- Step 11: An XSL transformation retrieves the values from the SMO shared context that were stored in steps 4 and 9, and assembles a new output message.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe the role of mediation services in applications
- Describe the prebuilt mediation primitives that are available in IBM Integration Designer

© Copyright IBM Corporation 2017

Figure 11-27. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: In a typical mediation scenario, all requests and responses pass through a single point.
2. True or False: A primitive modifies at least one element of the SMO header that is used to prepare for dynamic service invocation.
3. True or False: The SetMessageType mediation primitive has no effect on the structure or content of a message.
4. True or False: The EventEmitter primitive can have a tremendous performance impact because it always logs the contents of the message along with failure information.

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-28. Review questions

Write your answers here:

1.

2.

3.

4.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: In a typical mediation scenario, all requests and responses pass through a single point.
   The answer is False. In a mediation scenario, all requests and responses need not necessarily pass through a single point.
2. True or False: A primitive modifies at least one element of the SMO header that is used to prepare for dynamic service invocation.
   The answer is True.
3. True or False: The SetMessageType mediation primitive has no effect on the structure or content of a message.
   The answer is True.
4. True or False: The EventEmitter primitive can have a tremendous performance impact because it always logs the contents of the message along with failure information.
   The answer is False. EventEmitter logging can be set to the minimum level to avoid performance impact.

Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-29. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Implementing a mediation flow

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 11-30. Exercise 10

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

IBM Training

Exercise objectives

- Implement a service aggregation flow in a mediation module with multiple mediation primitives, including fan-out, fan-in, service invoke, XSL transformation, message element setter, and custom mediation
- Test a mediation module in the IBM Integration Designer integrated test environment

Mediation primitives

© Copyright IBM Corporation 2016

Figure 11-31. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

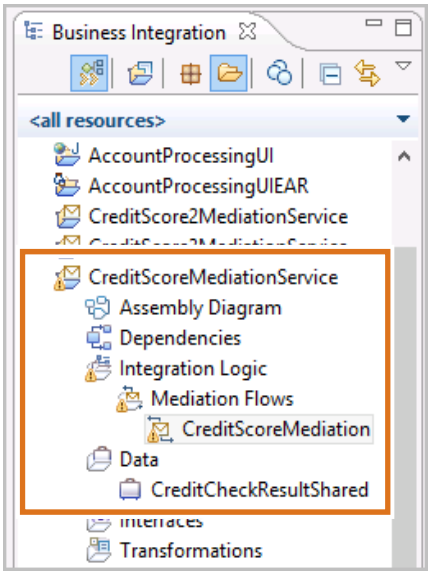
Components that are required for Exercise 10

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreMediationService**
3. **CreditScoreMediation** mediation flow
4. **AccountVerification** BPEL

New components that you create in the lab:

1. **CreditCheckResultShared** business object
2. **CustServiceCountryCode** business object map



Mediation primitives

© Copyright IBM Corporation 2017

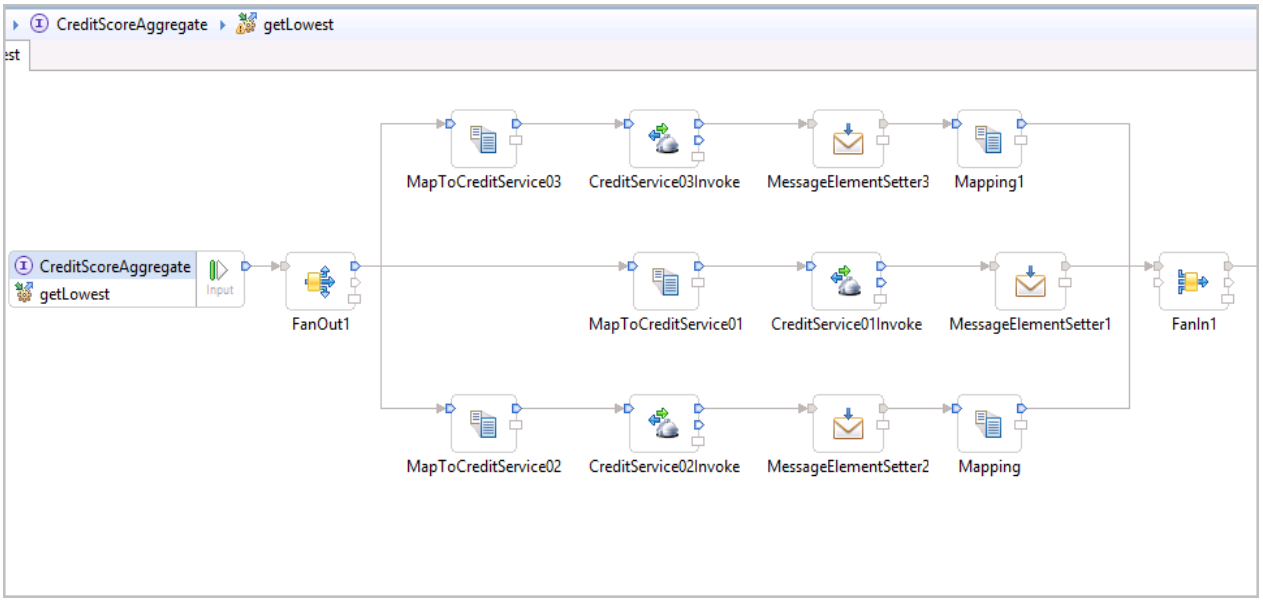
Figure 11-32. Components that are required for Exercise 10

© Copyright IBM Corp. 2017

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

IBM Training

Implement the CreditScoreMediation in Exercise 10



Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-33. Implement the CreditScoreMediation in Exercise 10

In this section, you implement the logic in the mediation module to start the three credit score providers. The logic also determines which of the returned credit scores is the lowest, and then returns that score to the invoking service.

To implement the mediation flow, you use the following primitives:

- A fan-out to start the aggregation. From this fan-out, three paths exist, one for each of the credit score service imports.
- A service invoke for each of the credit score services.
- An XSL transformation before each of the credit score invocations to convert the inbound message to the data type that each service invocation requires.
- A message element setter, which follows each of the credit score invocations to copy the response messages from the credit score services into the shared context. The custom mediation primitive queries the shared context.
- Two more XSL transformations, which follow the message element setter primitives for the second and third service invocations (described in further detail in the exercise).
- A fan-in to mark the end of the aggregation block.

© Copyright IBM Corp. 2017

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

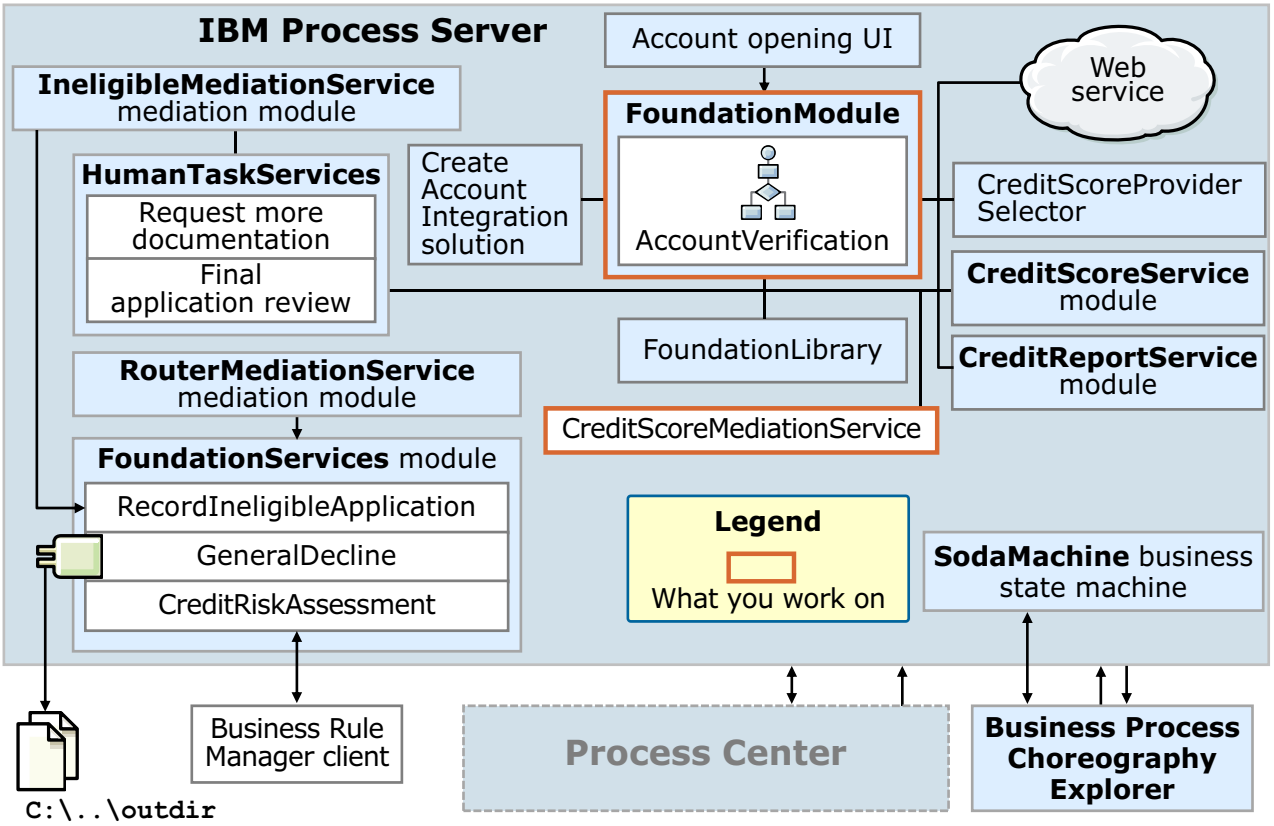
- A custom mediation primitive that determines which of the three credit scores that are returned is the lowest. This primitive constructs the output message based on the response message that returned the lowest credit score.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you with in Ex 10



Mediation primitives

© Copyright IBM Corporation 2017

Figure 11-34. Course business scenario: What you with in Ex 10

In this lab, you work with IBM Integration Designer.

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

Unit 12. Mediation flow control

Estimated time

01:00

Overview

This unit describes the primitives that can be used to control the flow of mediation. The unit also describes the function of event sequencing, and offers suggestions for integrating IBM Process Server with WebSphere Service Registry and Repository.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Selectively route messages in a mediation by using the message filter primitive and type filter primitive
- Use the flow order primitive to specify the order in which branches of a flow are run
- Explain how subflows work and the advantages of using them
- Describe the role of event sequencing in a mediation

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Flow control with mediation primitives
- Event sequencing
- Starting services dynamically

© Copyright IBM Corporation 2017

Figure 12-2. Topics

© Copyright IBM Corp. 2017

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

12.1. Flow control with mediation primitives

© Copyright IBM Corp. 2017

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

IBM Training

Flow control with mediation primitives

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 12-3. Flow control with mediation primitives

© Copyright IBM Corp. 2017

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

IBM Training

Mediation primitives: Routing (1 of 2)

These primitives control routing within the mediation flow:

Mediation primitive	Description
Endpoint lookup	Allows dynamic routing of requests by searching for service endpoints from a WebSphere Service Registry and Repository registry
Fan-in	Aggregates multiple messages (which the fan-out primitive creates) based on a decision point, then outputs a single message
Fan-out	Splits a message that contains repeating elements into multiple messages, or sends the same message more than one time to implement message aggregation
Flow order	Specifies the order in which branches of a flow are run
Gateway endpoint lookup	Routes service requests based on a service name or an action

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-4. Mediation primitives: Routing (1 of 2)

This group of built-in mediation primitives is used to route the flow of messages within a mediation, or to dynamically control how services are started. This table lists all transformation primitives that IBM Integration Designer offers. You are already familiar with some of these primitives.

© Copyright IBM Corp. 2017

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

IBM Training

Mediation primitives: Routing (2 of 2)

These primitives control service invocations inside the mediation flow and message routing outside the mediation flow:

Mediation primitive	Description
Message filter	Routes messages within a mediation flow that is based on the message content
Policy resolution	Dynamically configures a mediation flow by retrieving mediation policy and service endpoint information from a WebSphere Service Registry and Repository registry
Service invoke	Calls a service from within a mediation flow, instead of waiting until the end of the mediation flow and then using the callout mechanism
Type filter	Routes messages within a mediation flow, based on the message type

© Copyright IBM Corporation 2017

Figure 12-5. Mediation primitives: Routing (2 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Flow order primitive

- Defines the order in which the branches of a mediation are run
- At development time, you define a named output terminal for each branch that you want in the mediation, and then wire that terminal
- Any number of output terminals can be created; no configurable properties exist on the primitive
- At run time, the output terminals run in the order named (out1, out2, out3, . . .)
  - The same input message is propagated on each iteration
- Each branch of the flow is completed before the next terminal is fired, except:
  - If asynchronous service invoke is run, the next branch from the flow order starts immediately
  - An exception that is thrown downstream and not handled locally causes the remaining flow order branches not to fire

Mediation flow control © Copyright IBM Corporation 2017

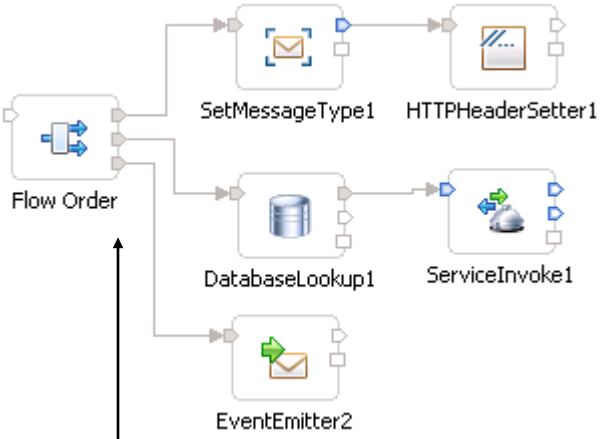
Figure 12-6. Flow order primitive

You can use the flow order primitive to control the order of execution of multiple branches in a mediation. You define as many output terminals as are needed to implement the number of mediation flow branches that you need for your application. At execution time, the branches run in the order that the terminal name dictates (out1, out2, out3, . . .). The flow order primitive has no configurable properties.

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

IBM Training

Flow order primitive: Example



Output terminals are shown in order
out1
out2
out3
. . .
and are run in that order

© Copyright IBM Corporation 2017

Figure 12-7. Flow order primitive: Example

This example shows a flow order primitive and the various output terminals that it contains. The message is propagated to the three output terminals in order (out1, out2, out3).

© Copyright IBM Corp. 2017

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

IBM Training

Message filter primitive

- Routes a message within a mediation flow, which is based on the message content
- Acts like an “if ... goto” statement to send an incoming message through one or more paths if filter criteria are met
- Any number of output terminals can be defined at development time
- Filter criteria are XPath expressions that you enter in a table at development time
- If no match occurs, the message is propagated through the *default* terminal

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-8. Message filter primitive

Use care when wiring the message filter outbound terminals. If you do not wire a flow to a terminal and the message is propagated to that unwired terminal at run time, the message is lost without warning or exception. It is especially important with the *default* terminal, which receives a message that does not match any of the XPath expressions in the table.

© Copyright IBM Corp. 2017

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

IBM Training

Message filter primitive: Usage examples

Use the message filter primitive to:

- Verify that the contents of a message meet certain conditions, such as a required field is present
  - If conditions are not met, start the error handler subflow or fail primitive to stop processing
- Route the message based on its contents
  - For example, a specific value in a field might indicate that ServiceProviderA should be started, while other values indicate that ServiceProviderB should be started
- Conditionally bypass steps in a mediation, based on message content

© Copyright IBM Corporation 2017

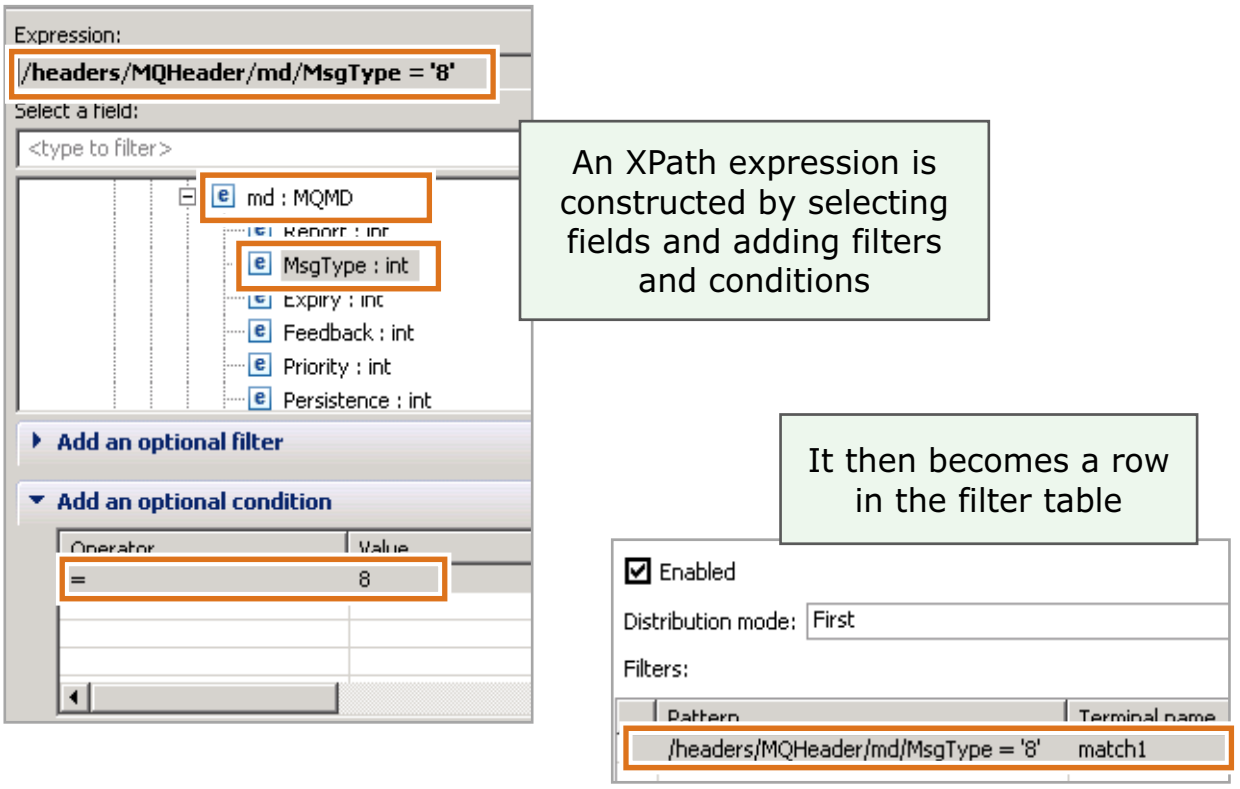
Figure 12-9. Message filter primitive: Usage examples

© Copyright IBM Corp. 2017

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

IBM Training

Message filter primitive: Configuration



© Copyright IBM Corporation 2017

Figure 12-10. Message filter primitive: Configuration

To configure the message filter primitive:

1. Construct a table of XPath expressions to evaluate the contents of elements or variables. You can reference any content in the SMO, not just that of the incoming message. If you want help, the XPath expression builder helps you construct the expression.
2. Create a terminal, or use an existing one, to receive the incoming message, if that condition is met.
3. Repeat those steps for all conditions for which you want to test.
4. Set the distribution mode. “First” sends the message only on the first “true” condition that is found as the table of expressions is scanned at run time. “All” sends the message to all terminals where the condition is true. It implies that the order in which expressions are in the table can be significant.
5. Set the enabled flag. You can disable a message filter without removing the primitive from the mediation flow. This flag is also a promotable property, so it can be changed administratively without altering the mediation flow.

At run time, the table of expressions is scanned in sequential order. If an XPath expression evaluates to “true,” the incoming message is sent out through the terminal that is listed in the table

© Copyright IBM Corp. 2017

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

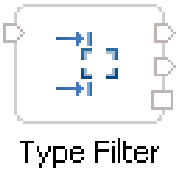
for that expression. Depending on the value of the distribution mode property, the table scan either stops or continues.

© Copyright IBM Corp. 2017

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

IBM Training

Type filter primitive



- Routes a message within a mediation flow, which is based on the message type
- Acts like an “if … goto” statement to send a message down a specific path if the filter criterion is met
- Any number of output terminals can be defined at development time
- Filter criteria are XPath expressions that you enter in a table at development time
- If no match occurs, the message is propagated through the default terminal

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-11. Type filter primitive

The type filter is similar to the message filter, but it evaluates the type of an element, rather than the content of an element.

Unlike the message filter primitive, the type filter primitive has no distribution mode property. At run time, the first expression that matches during the scan of the table propagates the inbound message to the corresponding terminal. It is not possible to match multiple rows in the expression table, so the order in which the expressions are shown can be important.

© Copyright IBM Corp. 2017

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

IBM Training

Mediation subflow: Review

- Facilitates reuse of mediation logic without rewriting
- Composed of:
  - Mediation primitives that are wired together
  - Starts with an *in* primitive and ends with an *out* primitive, acts like input and callout nodes of a mediation flow
- Parent flow uses it as if it were a primitive:
  - After a subflow is developed, it is dropped onto the canvas from the primitive as if it were a built-in primitive
  - Wired into the calling flow like any other primitive
- The caller can set the promoted properties of a subflow
- Subflows can be nested
- Can be contained in a library or module (library increases reusability)

Mediation flow control © Copyright IBM Corporation 2017

Figure 12-12. Mediation subflow: Review

A subflow is a preconfigured set of mediation primitives that you wire together to implement a common function. A mediation subflow is run in the context of a parent flow, and can be reused in mediation flows or in subflows.

A mediation subflow can be **created** in a module, mediation module, or library. A mediation subflow can be **used** only within a mediation flow. It is good practice to store subflows in libraries so that they can be easily shared between mediation flows in different modules.

A mediation subflow has one or more **in** nodes, with one or more mediation primitives that follow, and one or more **out** nodes. Unlike mediation flows, the inputs and outputs of a subflow are not tied to specific interfaces. Instead, the **in** and **out** nodes represent the start and endpoints of a subflow. In a wired mediation flow, these nodes have message types that are associated with them. The message types on these nodes are not defined when the subflow is created, and they can be changed later.

You define a subflow implementation by using the subflow editor. After it is defined, it is visible as if it is a new type of primitive that you embed in the parent flow, at which point it is expanded into the parent flow. From that perspective, it is more like a macro than an externally callable function.

Because subflows are run in the context of a parent flow, properties that are promoted in a subflow are propagated up to the mediation flow in which the subflow is used. If you define a mediation flow

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

that starts a subflow, the promoted properties for the subflow are visible with the promoted properties of the parent flow.

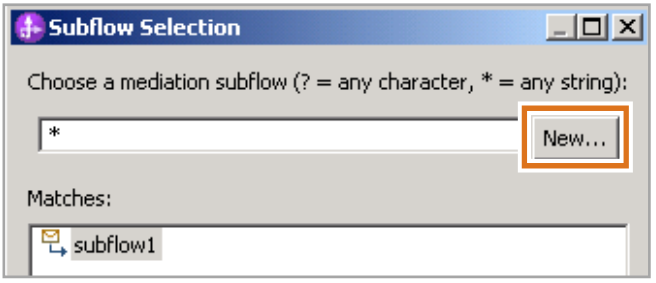
© Copyright IBM Corp. 2017

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

IBM Training

Mediation subflow: Implementation

- To create a subflow, select the subflow primitive from the palette and place it on the canvas
  - The subflow selection dialog is shown
  - You can choose an existing subflow or create a new one



- When creating a subflow, the subflow editor opens and places *in* and *out* nodes on the canvas
  - You add remaining nodes, and configure and wire them

- After the new subflow is saved, it is listed in the subflow selection dialog whenever a subflow is requested

© Copyright IBM Corporation 2017

Figure 12-13. Mediation subflow: Implementation

© Copyright IBM Corp. 2017

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

IBM Training

Mediation subflow: Restrictions

- You cannot use a policy resolution primitive in a subflow
- You cannot specify a context element within a subflow
  - Context elements that are available in the parent flow are also available to mediation primitives in the subflow
- When debugging a subflow, be aware of these limitations:
  - You cannot step into a subflow
  - You must set a breakpoint in the subflow to view the subflow execution
  - You cannot step out of a subflow
  - If you try to step over the last primitive in the subflow, the flow runs to completion
  - To work around this limitation, set a breakpoint in the main flow after the subflow primitive

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-14. Mediation subflow: Restrictions

The use of subflows has a few restrictions, in both programming and debugging.

© Copyright IBM Corp. 2017

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

12.2. Event sequencing

© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 12-15. Event sequencing

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

IBM Training

Event sequencing

- Event sequencing is not a specific type of primitive
  - It is a quality of service attribute
  - Ensures that events that are received from asynchronous invocations are processed in the order in which they are received
- An *event* is a request or response that is sent from one component to another
  - An event includes the data objects and the “invocation metadata” (the name of the target component, the operation, and the parameters, for example)
- Event sequencing is activated by configuring *event sequence qualifiers* in IBM Integration Designer
  - The qualifiers must be set on each method that requires event sequencing
  - These qualifiers tell the runtime environment that invocations to these methods need to be sequenced
- Event sequencing applies to the entire integration scenario

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-16. Event sequencing

© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing: How it works

- Each qualifier has an *event sequencing key* that determines how events are sequenced
  - The value of the key consists of one or more attributes of the business objects that are associated with an invocation
  - All events that share the key are grouped and processed in sequence
  - Events that do not have an event sequencing key continue to be processed as normal, in parallel with the sequenced events
- A sequenced event acquires a lock before being sent to the target component for processing
  - After the business logic for the event runs, the lock is released and given to the next event with the same event sequencing key
  - If the event cannot acquire the necessary lock, the execution of the invocation is suspended until the lock is acquired
- Event sequencing is supported only for requests that are sent with an asynchronous invocation

© Copyright IBM Corporation 2017

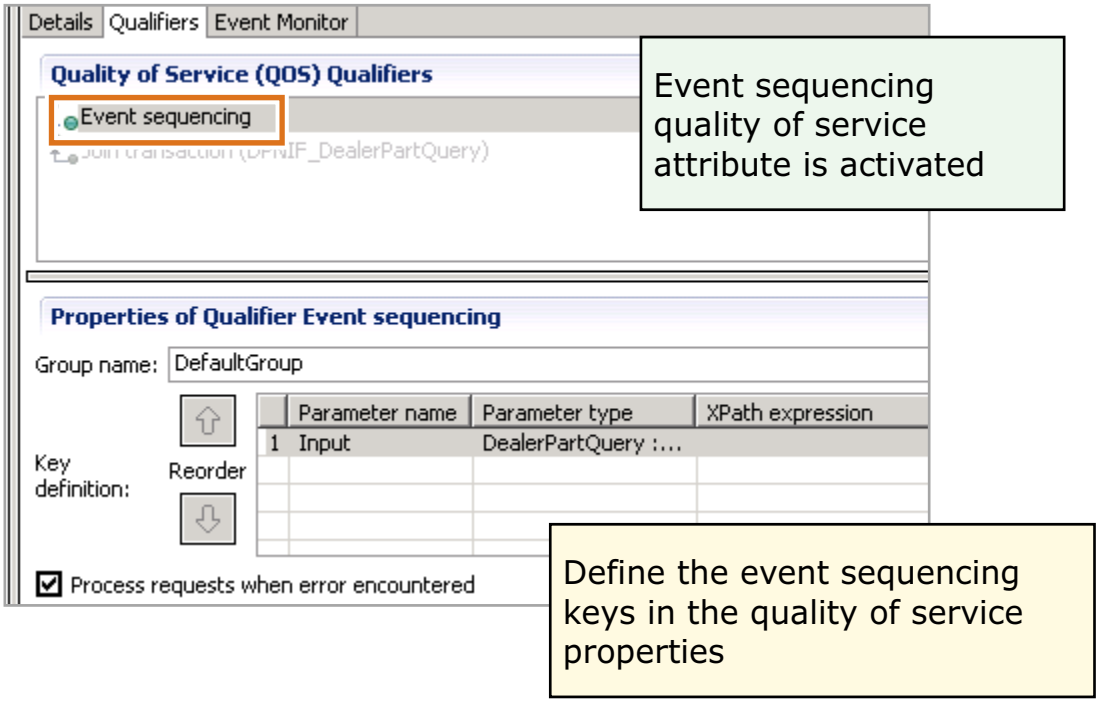
Figure 12-17. Event sequencing: How it works

© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing: Defining the keys



© Copyright IBM Corporation 2017

Figure 12-18. Event sequencing: Defining the keys

Define the keys for event sequencing in the Properties view of the primitive to which you want the QoS attributes to be applied.

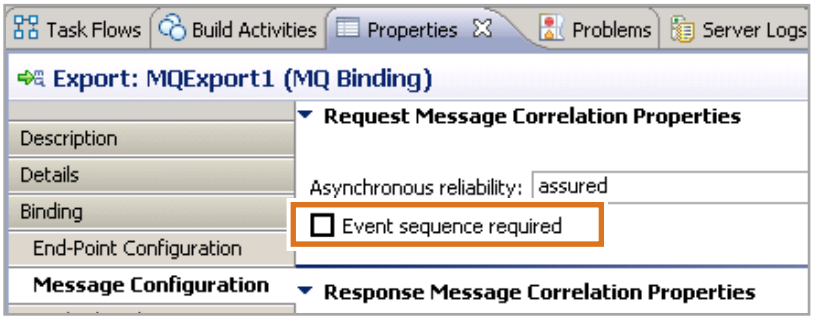
© Copyright IBM Corp. 2017

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

IBM Training

Event sequencing in IBM Process Server

- In IBM Process Server, event sequencing is activated in the export bindings, since they are the main entry points to the target SCA components
- To activate event sequencing, select the **Properties** view for the export and then select the **Event sequence required** option



- Some additional configuration is required when using the EIS export, as the management of event sequencing is vendor-specific

© Copyright IBM Corporation 2017

Figure 12-19. Event sequencing in IBM Process Server

Set the event sequence required option in the **Message Configuration** tab of the Properties view of IBM Integration Designer, as shown in the figure here.

© Copyright IBM Corp. 2017

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

12.3. Starting services dynamically

© Copyright IBM Corp. 2017

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

IBM Training

Starting services dynamically

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 12-20. Starting services dynamically

© Copyright IBM Corp. 2017

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

IBM Training

Introduction to dynamic message processing

- All of the service invocations that were presented so far were static
  - The connections to the external services were “fixed” or “hard-wired”
- Start a service dynamically, by using some capability at run time to be able to determine what service to call, and how to call it
- You can also change the processing that occurs within a mediation flow, based on policies that you retrieve from WebSphere Service Registry and Repository
- You typically discover service descriptions by using a Universal Description, Discovery, and Integration (UDDI) registry or WebSphere Service Registry and Repository
- Several mediation primitives are built to work with WebSphere Service Registry and Repository

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-21. Introduction to dynamic message processing

Generally, you issue queries against the registries (WebSphere Service Registry and Repository or UDDI) to discover the service artifacts at development time. After you discover these artifacts, you import the ones that you need into the projects in your workspace. The ability to dynamically route messages to the appropriate service endpoints requires that you add the actual service endpoint information to the registry ahead of time.

© Copyright IBM Corp. 2017

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

IBM Training

Mediation primitives: Dynamic routing

These primitives are used with dynamic service invocations:

Mediation primitive	Description
Endpoint lookup	Allows dynamic routing of requests by searching for service endpoints from a WebSphere Service Registry and Repository registry
Gateway endpoint lookup	Routes service requests based on a service name or an action
Policy resolution	Dynamically configures a mediation flow by retrieving mediation policy and service endpoint information from a WebSphere Service Registry and Repository registry
SLA check	Enforces service level agreements (SLA); determines whether a consumer has an SLA in place to use a service
UDDI endpoint lookup	Dynamically routes messages to service endpoints obtained from UDDI

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-22. Mediation primitives: Dynamic routing

The primitives that are listed on this visual are used in the context of dynamic service invocation.

© Copyright IBM Corp. 2017

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

IBM Training

WebSphere Service Registry and Repository

- WebSphere Service Registry and Repository is a separately purchased and installed product
- Facilitates storing, accessing, managing service information (called metadata), so you can easily select, start, govern, and reuse services
- You can use it to store, access, and manage information about services
  - You can use this information to select, start, and reuse services
- Use WebSphere Service Registry and Repository to store information about services in other systems that you plan to use, or of which you want to be aware
  - For example, an application can check WebSphere Service Registry and Repository before it starts a service to locate the most appropriate service that satisfies its functional and performance needs
  - This capability helps make your deployment more dynamic and more adaptable to changing business conditions
  - You can access WebSphere Service Registry and Repository through mediation flows

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-23. WebSphere Service Registry and Repository

Several of the mediation primitives that are used in dynamic message routing use WebSphere Service Registry and Repository.

WebSphere Service Registry and Repository is the master metadata repository for service interaction endpoint descriptions. It is a separately priced and separately installed product from IBM Process Server.

A broad definition of “service” applies here. It includes traditional web services that implement WSDL interfaces with SOAP/HTTP bindings. It also includes a broad range of SOA services that can be described with WSDL, XSD, and policies. They might also use a range of protocols and be implemented according to various programming models.

As the integration point for service metadata, WebSphere Service Registry and Repository establishes a central point for finding and managing service metadata that is acquired from a number of sources. These sources include service application deployments, endpoint registries and repositories, such as Universal Description, Discovery, and Integration (UDDI), and other service metadata. It is where service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service. After it occurs, several things happen:

- Visibility is controlled.

© Copyright IBM Corp. 2017

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

- Versions are managed.
- Proposed changes are analyzed and communicated.
- Usage is monitored.
- Other parts of the SOA foundation can access service metadata.

These events happen with the confidence that they found the copy of record.

WebSphere Service Registry and Repository does not manage all service metadata, and it does not manage service metadata across the whole SOA lifecycle. It focuses on a minimalist set of metadata that describes capabilities, requirements, and the semantics of deployed service endpoints. It interacts and federates with other metadata stores that play a role in managing the overall lifecycle of a service.

During the development stages of the business service lifecycle, WebSphere Service Registry and Repository is used to locate the copies of record of candidate service interaction endpoints or mediating intermediaries, and policies that govern the interactions.

As in the other SOA lifecycle stages, repositories that specialize in managing SOA artifacts during the development stages of the lifecycle complement WebSphere Service Registry and Repository. For example, a development artifact management system such as Rational ClearCase takes care of service and composite application building blocks. These building blocks contain source code, service interface declarations, software architecture models, or business process models that are under construction. A reusable asset manager and asset repository manage bundles of artifacts, which describe assets according to the reusable asset specification (RAS) standard. It implements governance processes that control the promotion of artifacts to assets and the approval process that is associated with it.

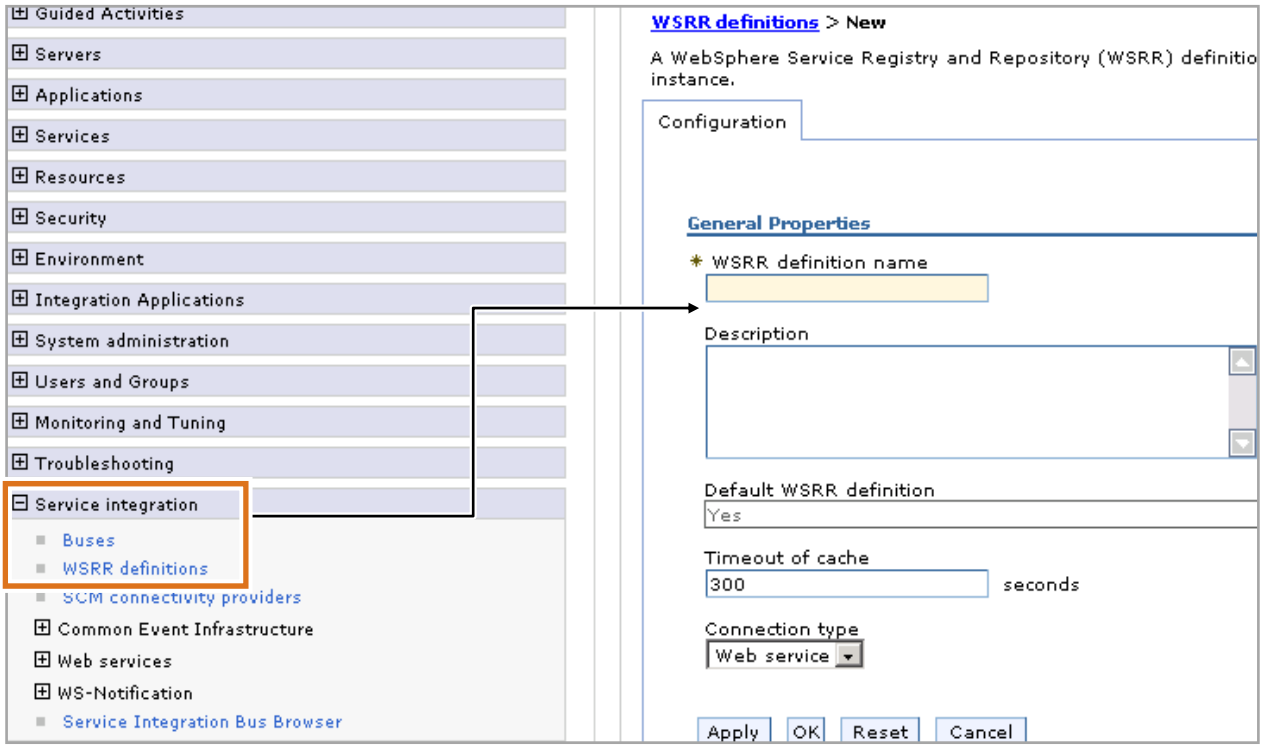
After the development team finishes their work and testing is complete, deployers further augment the service metadata. Deployment teams provide binding information for service endpoints that are used in composite applications. They also manage deployment of the metadata from the development environment to the staging or production Service Registry and Repository as part of the service deployment process. Governance over the service metadata takes place, as metadata is promoted from test to staging to production environments that might have separate WebSphere Service Registry and Repository installations. In the production environment, WebSphere Service Registry and Repository is made available to a broader audience and shared. It is available to the runtime systems and those user roles that are responsible for the management of the IT systems.

© Copyright IBM Corp. 2017

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

IBM Training

WebSphere Service Registry and Repository Registry Definition



Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-24. WebSphere Service Registry and Repository Registry Definition

You can use service integration settings in IBM Process Server to define WebSphere Service Registry and Repository references at a single point.

© Copyright IBM Corp. 2017

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

IBM Training

Endpoint lookup primitive: Overview

- Searches WebSphere Service Registry and Repository, by using a query that is constructed from the properties of the primitive
- Query can return 0, 1, or more service endpoints
  - You can choose whether to be informed of all endpoints that match the query, or just one endpoint that matches the query
- Use to dynamically route messages, based on some criteria in the message
  - For example, send messages for customers of type A to URL *a*, and messages for customers of type B to URL *b*
  - If you set up the WebSphere Service Registry and Repository registry to key URLs against customer types, you can dynamically route customer requests according to customer type

Mediation flow control © Copyright IBM Corporation 2017

Figure 12-25. Endpoint lookup primitive: Overview

The endpoint lookup primitive searches WebSphere Service Registry and Repository to find one or more sets of endpoints that are then used to route a message dynamically.

Endpoint lookup is supported for:

- Web services with SOAP/HTTP or SOAP/JMS
- SCA module exports with web service bindings, with SOAP/HTTP or SOAP/JMS
- SCA module exports with these bindings: default SCA, IBM MQ, IBM MQ JMS, JMS, generic JMS, or HTTP
- Manually defined IBM MQ, JMS, or HTTP endpoints with associated interface

The **match policy** property of the endpoint lookup primitive instructs the primitive how to act, depending on the matches found in WebSphere Service Registry and Repository. The match policy can be one of these values:

- **Return the first matching endpoint and set the routing target**. If the registry query returns matches, you get the following results:
  - The dynamic callout address, in the SMO header, is updated with one service address from the result returns.

© Copyright IBM Corp. 2017

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

  - The SMO context is updated with registry information, which relates to the address in the dynamic callout address.
  - The alternative targets list in the SMO header is cleared.
- **Return all matching endpoints**. If the registry query returns matches, you get the following results:
  - The SMO context is updated with registry information, which relates to all services that the registry returns.
  - The dynamic callout address, in the SMO header, is cleared.
  - The alternative targets list, in the SMO header, is cleared.
  - You must post-process the SMO to select a service endpoint for dynamic invocation.
- **Return all matching endpoints and set alternative routing targets**. If the registry query returns matches, you get the following results:
  - The dynamic callout address, in the SMO header, is set with the first match.
  - The alternative targets list, in the SMO header, is updated with the remaining matches.
  - All services that are found are also placed into the primitive context.
- **Return endpoint that matches most recent compatible service version**. If the registry query returns matches, you get the following results:
  - The dynamic callout address, in the SMO header, is updated with one service address from the results returned.
  - The SMO context is updated with registry information, which relates to the address in the dynamic callout address.
  - The alternative targets list, in the SMO header, is cleared.

© Copyright IBM Corp. 2017

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

IBM Training

Endpoint lookup primitive: Actions

- Successful query updates both the SMO context (the primitiveContext element) and the SMO headers:
  - **`/headers/SMOHeader/Target/address`**
    contains the address of a service to start dynamically
    (the dynamic callout address)
  - **`/context/primitiveContext/EndpointLookupContext`**
    contains the results of the WebSphere Service Registry and Repository query
  - **`/headers/SMOHeader/AlternateTarget`**
    contains a list of alternative service addresses
- The endpoint lookup primitive does not start a service
  - It sets only the SMO headers for the next primitive that can use that routing information (for example, a service invoke)
- Primitive contains a *noMatch* terminal
  - Inbound message is routed there when no policy match is found in WebSphere Service Registry and Repository

© Copyright IBM Corporation 2017

Figure 12-26. Endpoint lookup primitive: Actions

If a query to WebSphere Service Registry and Repository from the endpoint lookup is successful, the SMO context and SMO header elements are updated as shown. The endpoint lookup mediation does not do a service invocation; it sets the context and headers of the SMO to prepare for a dynamic invocation later in the mediation flow.

Consider your objectives or goals when using WebSphere Service Registry and Repository with WebSphere IBM Process Server. Dynamic endpoint lookup, for example, assists you with decoupling hardcoded service URLs from mediation modules. It stores and manages service endpoints in a centralized repository (WebSphere Service Registry and Repository). The functions with the dynamic endpoint lookup primitive can include matches on classifications on the WSDL artifact. Dynamic endpoint lookup facilitates runtime discovery of service endpoint location. WebSphere Service Registry and Repository query criteria are set as properties on the primitive.

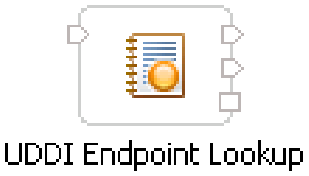
Result selection can be automatic or programmatic. Automatic result selection allows the primitive to select one service match. In this case, the SMO Dynamic Callout Header property automatically updates with a single endpoint reference. With programmatic selection, the user selects a service match from the set of all returned matches. In this case, the SMO Primitive Context is populated with endpoint references from which to choose.

© Copyright IBM Corp. 2017

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

IBM Training

UDDI endpoint lookup primitive: Overview



- Retrieve service endpoint information from a Universal Description, Discovery, and Integration (UDDI) registry
  - The service endpoint information relates directly to web services by using SOAP/HTTP or SOAP/JMS
- Query can return 0, 1, or more service endpoints
  - You can choose whether to be informed of all endpoints that match the query, or just one endpoint that matches the query
- Uses the endpoint reference structure that the WS-Addressing specification defines
- Usage and behavior are similar to that of the endpoint lookup primitive

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-27. UDDI endpoint lookup primitive: Overview

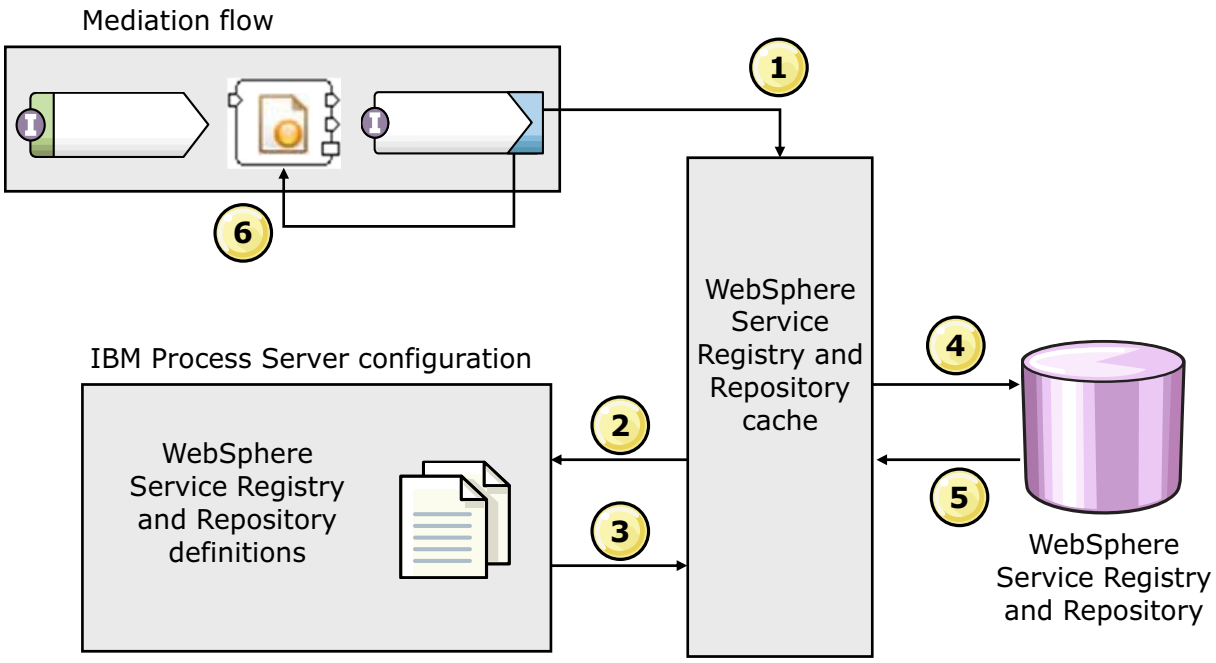
The UDDI endpoint lookup does the same function as the endpoint lookup primitive, but it queries a UDDI registry rather than WebSphere Service Registry and Repository. The UDDI endpoint lookup is supported for web services with SOAP/HTTP or SOAP/JMS.

© Copyright IBM Corp. 2017

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

IBM Training

Endpoint lookup component interaction



© Copyright IBM Corporation 2017

Figure 12-28. Endpoint lookup component interaction

1. ELP makes a query to the cache.
2. The WebSphere Service Registry and Repository connection details are requested from the admin (if not already known to the cache).
3. WebSphere Service Registry and Repository connection details returned.
4. A query is sent to the WebSphere Service Registry and Repository unless the result is already in the cache.
5. Results are returned to the cache and stored.
6. The results are forwarded to the ELP.

Use the information in WebSphere Service Registry and Repository for the automatic selection of a target endpoint. This step requires that either one of the following match policies is selected:

- Return all matching endpoints, and set alternative routing targets
- Return the first matching endpoint, and set the routing target

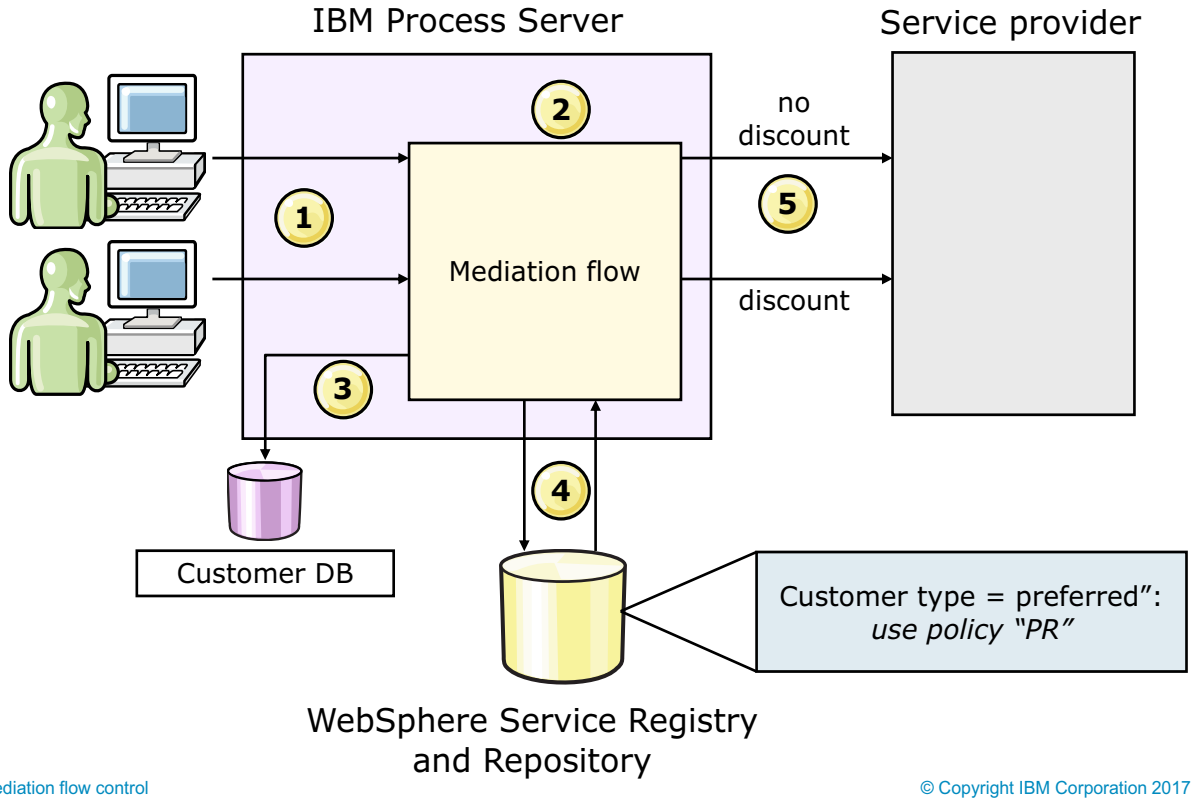
The “Use dynamic endpoint” module property controls the usage.

© Copyright IBM Corp. 2017

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

IBM Training

Mediation policies and dynamic service interactions



Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-29. Mediation policies and dynamic service interactions

Mediation policies can be used to dynamically alter message processing. In this example, a preferred customer earns a discount when placing an order. It is accomplished through dynamic modification of the properties of mediation primitives within the mediation flow. The flow uses values that are obtained from WebSphere Service Registry and Repository, as follows:

1. The customer places an order.
2. The mediation flow processes the message.
3. The mediation flow determines the type of customer by looking up the customer ID in a database. The retrieved customer type is stored in the SMO.
4. A policy resolution primitive queries the WebSphere Service Registry and Repository by using the customer type. If the customer type is "preferred", a policy match is found; otherwise, no policy match is found.
5. If policy match was found, promoted properties of primitives in the mediation flow are overridden (for example, message element setter) to set the "discount" element. The service provider applies a discount as the result of the "discount" element that is set to a nonzero value. (Promoted properties whose values are derived from policies are called "dynamic properties".)

A policy is not needed for "regular" (nonpreferred) customers; since no policy match is found in WebSphere Service Registry and Repository, no overrides are applied to the subsequent mediation

© Copyright IBM Corp. 2017

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

primitives. This scenario does not select a different service provider, only a change in the processing within the mediation flow. (To associate a mediation policy with a target service rather than a module, add an endpoint lookup mediation primitive before the policy resolution mediation primitive.)

© Copyright IBM Corp. 2017

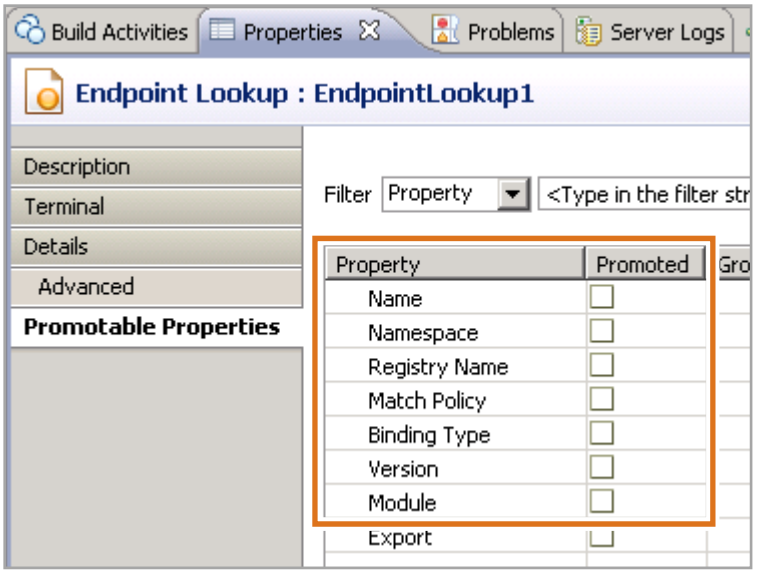
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Endpoint lookup (ELP) promoted properties

ELP properties can be promoted for change in the administration console at run time

- Registry name (WebSphere Service Registry and Repository definition)
- Port type
  - Name
  - Namespace
  - Version
- Classification
- User property value
- Match policy



© Copyright IBM Corporation 2017

Figure 12-30. Endpoint lookup (ELP) promoted properties

The **Name**, **Namespace**, and **Version information** can be changed for the port type property. Only the user property value type cannot be promoted.

Promoted properties can be used to add dynamicity to the retrieval and selection logic of service endpoints.

© Copyright IBM Corp. 2017

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

IBM Training

Gateway endpoint lookup primitive: Overview

- Use to route service requests, based on a *virtual service name*, when the module is acting as a proxy gateway
- Can also use to route service requests based on an *action*, when the module is acting as a service gateway

Gateway Endpoint Lookup

- If the gateway endpoint lookup mediation primitive finds suitable endpoints, it updates the SMO headers and context:
  - The dynamic callout address **`/headers/SMOHeader/Target/address`** is set with the first match
  - The alternative targets list **`/headers/SMOHeader/AlternateTarget`** is updated with the remaining matches
  - All services that are found are also placed in the primitive context **`/context/primitiveContext/EndpointLookupContext`**

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-31. Gateway endpoint lookup primitive: Overview

You can create a proxy gateway that routes requests to virtual services defined in *proxy groups*. A proxy gateway is a specific type of service gateway that defines proxy groups with which the administrator can associate real endpoints. Alternatively, you can create other types of service gateways that route to web service endpoints, according to the action that is specified in the message.

If you create a proxy gateway, the service definitions are stored in a built-in configuration store. If you create another type of service gateway, the service definitions are stored in WebSphere Service Registry and Repository.

© Copyright IBM Corp. 2017

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

IBM Training

SLA check primitive: Overview

- Use the SLA check mediation primitive to enforce service level agreements (SLAs)
- A message that the IBM Process Server mediates can be routed to a target service
  - To determine whether the consumer of the target service has a service level agreement in place to use the service, use the SLA check mediation primitive, which checks that the correct SLA exists in WebSphere Service Registry and Repository
- To find a matching SLA in WebSphere Service Registry and Repository, the SLA check primitive uses information in the incoming message
- The SLA is matched on three parameters:
  - Endpoint
  - Consumer identifier
  - Context identifier

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-32. SLA check primitive: Overview

The SLA check primitive has two output terminals: **accept** and **reject**. If the information passed on the incoming message is used to successfully find a matching SLA in WebSphere Service Registry and Repository, the accept terminal is fired. If no matching SLA is found, the reject terminal is fired.

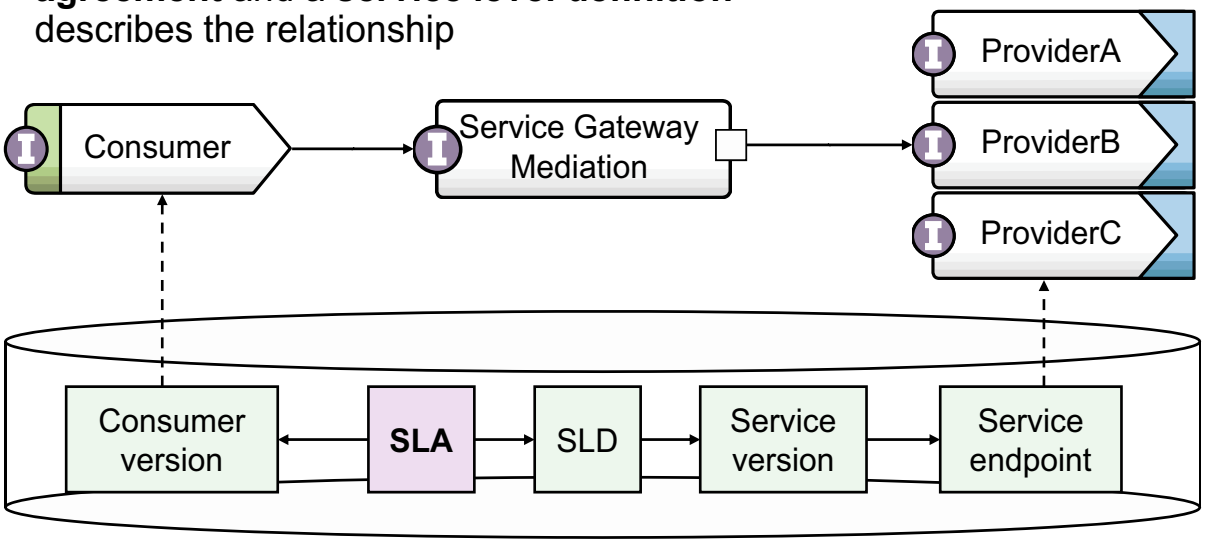
© Copyright IBM Corp. 2017

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

IBM Training

Service level agreements

- A consumer wants to use a particular service
- The dependency between a **consumer version** and a **service version** describes the relationship
- More specifically, an operational dependency between a **service level agreement** and a **service level definition** describes the relationship



© Copyright IBM Corporation 2017

Figure 12-33. Service level agreements

© Copyright IBM Corp. 2017

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

IBM Training

Policy resolution primitive: Overview

- You can use it to dynamically configure a mediation flow by using mediation policies
  - Mediation policies are stored in, and retrieved from, WebSphere Service Registry and Repository

- With the policy resolution primitive, you can retrieve mediation policies that are associated with the current SCA module
- Policy queries are created by using the properties of the primitive
- If valid mediation policies are found in the registry:
  - Use their contents to override the dynamic properties of mediation primitives that come after the policy resolution primitive
- It is acceptable for a query to return no matching policies
- If an error occurs while trying to look up the policy, the inbound message is propagated through the policyError terminal

© Copyright IBM Corporation 2017

Figure 12-34. Policy resolution primitive: Overview

If you installed WebSphere Service Registry and Repository, you can use it to dynamically override the values of promoted properties within a mediation flow. The **policy resolution** primitive accesses the WebSphere Service Registry and Repository registry to retrieve a policy that you specify. If a matching policy is found, any corresponding promoted properties in the mediation flow that follow the policy resolution primitive are overridden.

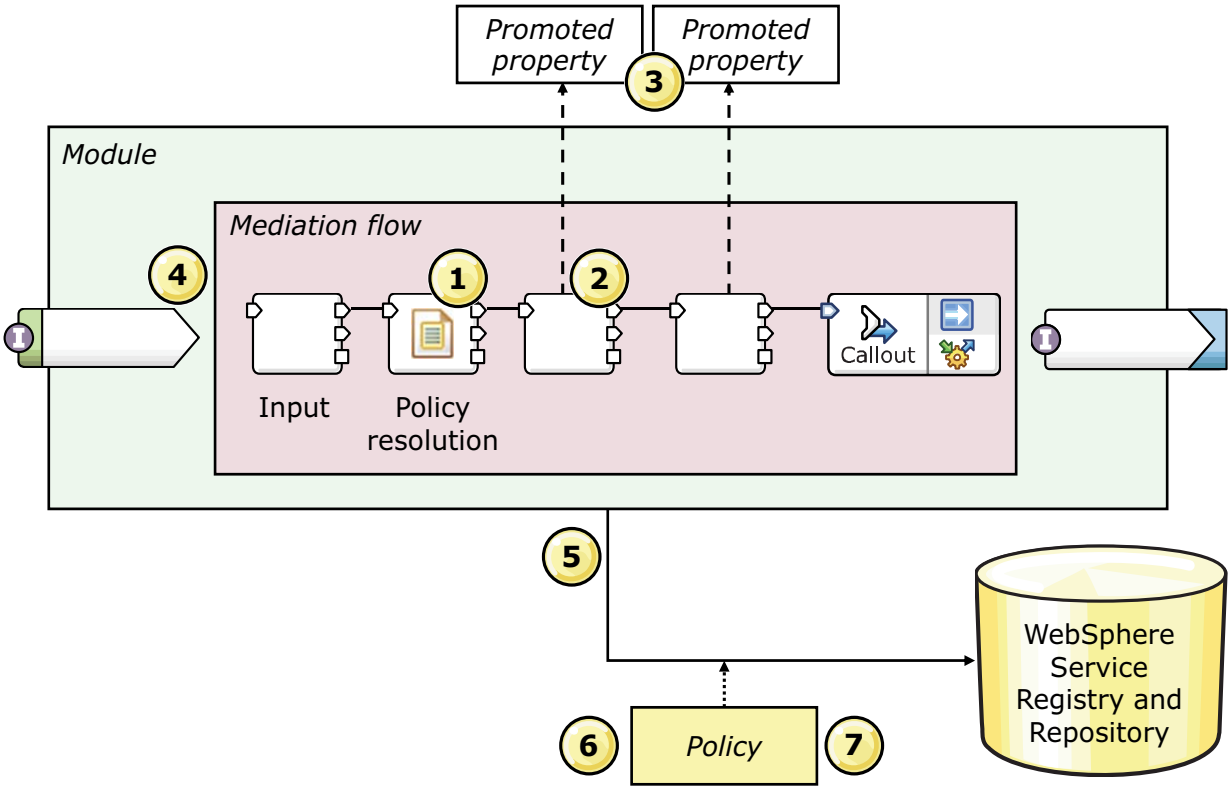
For example, suppose that you create a module that contains a mediation flow component, which contains two mediation primitives: a policy resolution primitive with an XSL transformation primitive. If you promote the “mapping file” property in the XSL transformation primitive, you can override its value at run time with mediation policies that are retrieved from the WebSphere Service Registry and Repository registry. If a matching policy is found that contains overrides to promoted properties in the subsequent mediation flow primitives, the SMO is propagated through the out terminal, along with the overridden properties. (If the policy resolution primitive does not find a matching policy, or if a policy is found but no overrides are available to promoted properties, the message assembly is still propagated through the out terminal.) If an error occurs while the policy resolution primitive is attempting to retrieve a policy, the message is propagated through the policyError terminal. However, it is still possible that property overrides can occur, depending on the characteristics of the mediation policy property model.

© Copyright IBM Corp. 2017

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

IBM Training

Interaction with mediation policy



Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-35. Interaction with mediation policy

In this example, a developer created a mediation flow with the intent of making it policy controlled.

The **developer** does the following steps:

1. The developer inserts a policy resolution primitive into the flow and configures it.
2. The position of the policy resolution primitive in this flow allows any subsequent mediation primitives to be mediation-policy controlled.
3. If the developer promotes the properties of any primitive that must be policy controlled, the value of each such property can now be overridden at run time through “dynamic property support.”
4. The module definition can now be exported, and is ready for deployment.

The **administrator** responsibilities include:

1. Import the module into WebSphere Service Registry and Repository. In this case, a logical representation of the module is created in WebSphere Service Registry and Repository. A default policy instance is created by using the values that are specified for the promoted properties. This option also primes the WebSphere Service Registry and Repository policy authoring infrastructure to initialize with the values that are specified for the promoted properties.

© Copyright IBM Corp. 2017

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

2. The administrator attaches one or more policies to the module.
3. Optionally, the administrator also specifies a condition on a policy attachment, which determines when the relevant policy applies.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Selectively route messages in a mediation by using the message filter primitive and type filter primitive
- Use the flow order primitive to specify the order in which branches of a flow are run
- Explain how subflows work and the advantages of using them
- Describe the role of event sequencing in a mediation

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-36. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. Which primitive is used to verify that the contents of a message meet certain conditions, and routes the message based on its contents?
   A. Flow order primitive
   B. Type filter primitive
   C. Message filter primitive
   D. Endpoint lookup primitive
2. True or False: You can set the endpoints for dynamic service invocation and dynamically change the behavior of a mediation flow.
3. True or False: WebSphere Service Registry and Repository is included with IBM Process Server

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-37. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. Which primitive is used to verify that the contents of a message meet certain conditions, and routes the message based on its contents?
   A. Flow order primitive
   B. Type filter primitive
   C. Message filter primitive
   D. Endpoint lookup primitive

   The answer is C.

2. True or False: You can set the endpoints for dynamic service invocation and dynamically change the behavior of a mediation flow.
   The answer is True.

3. True or False: WebSphere Service Registry and Repository is included with IBM Process Server.
   The answer is False. WebSphere Service Registry and Repository is available as a separate product.

Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-38. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Writing a generic error handler for IBM Process Server

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 12-39. Exercise 11

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

IBM Training

Exercise objectives

- Write a mediation subflow
- Implement mediation logic to manage runtime errors
- Use mediation primitives to control the flow of execution in a mediation, including the message filter and flow order primitives
- Implement the custom mediation primitive

Mediation flow control

© Copyright IBM Corporation 2016

Figure 12-40. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

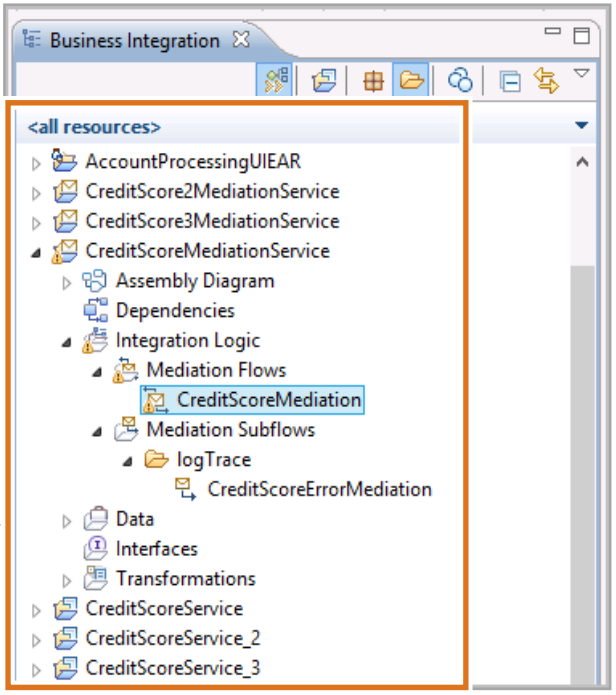
Components that are required for Exercise 11

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreMediationService**
3. **CreditScoreMediation** mediation flow
4. **AccountVerification** BPEL
5. **CreditScoreService**
6. **CreditScoreService_2**
7. **CreditScoreService_3**

New components that you create in the lab:

1. **CreditScoreErrorMediation** mediation subflow
2. **HandleCreditServiceError** mediation subflow



Mediation flow control

© Copyright IBM Corporation 2017

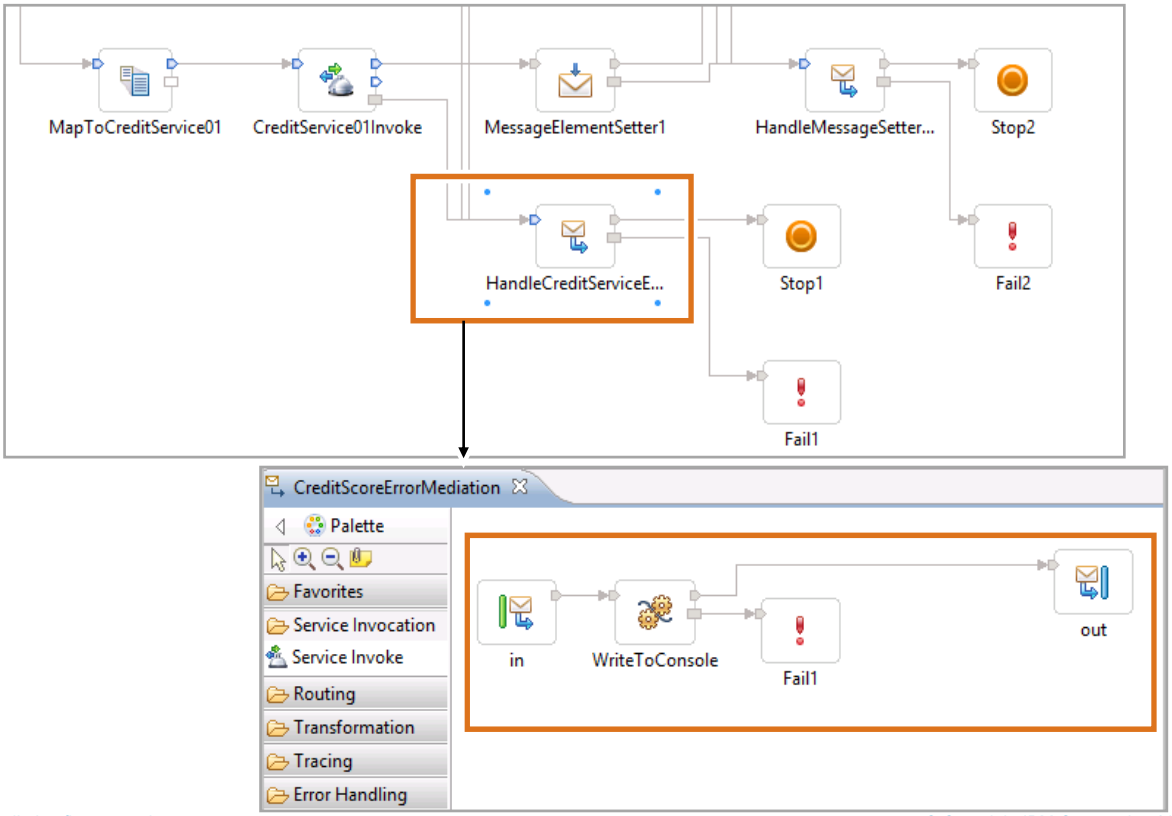
Figure 12-41. Components that are required for Exercise 11

© Copyright IBM Corp. 2017

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

IBM Training

Implement the CreditScoreErrorMediation subflow in Ex 11



© Copyright IBM Corporation 2017

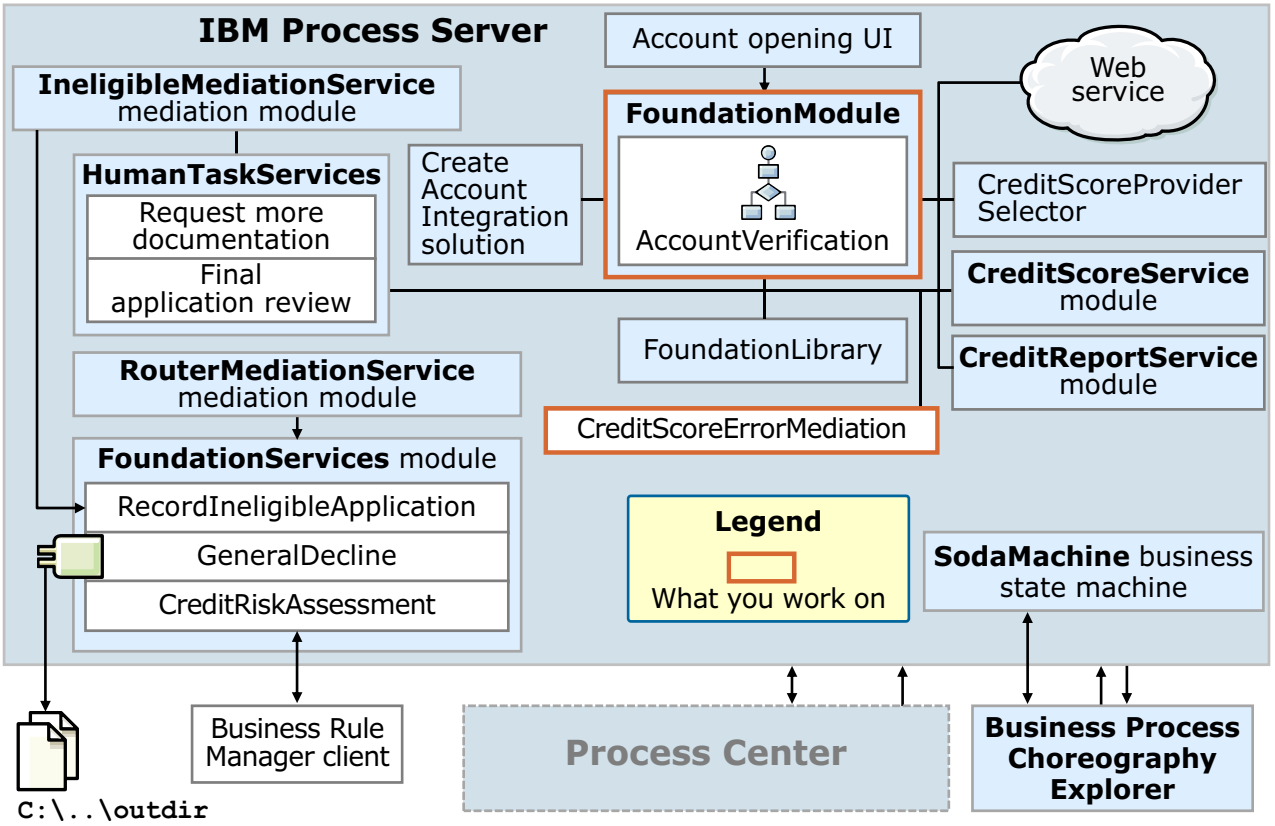
Figure 12-42. Implement the CreditScoreErrorMediation subflow in Ex 11

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 11



Mediation flow control

© Copyright IBM Corporation 2017

Figure 12-43. Course business scenario: What you work with in Ex 11

In this lab, you work with IBM Integration Designer.

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

Unit 13. Security

Estimated time

01:15

Overview

This unit provides a general overview of security in IBM Process Server, including server security and application security topics.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain the role of security quality of service qualifiers in SCA applications
- Explain the basic security requirements for IBM Process Server
- Describe the methods that are used to secure SCA applications
- Identify the role-based security elements of processes and human tasks

Security

© Copyright IBM Corporation 2017

Figure 13-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Topics

- Securing IBM Process Server runtime engine
- Application security overview
- SCA security qualifiers
- Securing business processes

© Copyright IBM Corporation 2017

Figure 13-2. Topics

© Copyright IBM Corp. 2017

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

13.1. Securing IBM Process Server runtime engine

© Copyright IBM Corp. 2017

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

IBM Training

Securing IBM Process Server runtime engine

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 13-3. Securing IBM Process Server runtime engine

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

IBM Training

IBM Business Process Manager: Users and groups management (1 of 2)

- IBM Process Server and WebSphere Application Server are the foundation
- Two consoles are available to manage users and groups
  - Integrated Solution Console
  - Process Admin Console

Security

© Copyright IBM Corporation 2017

Figure 13-4. IBM Business Process Manager: Users and groups management (1 of 2)

IBM Business Process Manager is built on top of IBM Process Server and WebSphere Application Server. The users and groups management starts at the WebSphere Application Server level and is shared with the Process Center. You can manage users and groups in two places:

- Integrated Solution Console, also known WebSphere Admin Console
- Process Admin Console

The users and groups that are created in the WebSphere Admin Console are stored in the WebSphere user registry, and the users that are created in Process Admin Console are stored in the file registry.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager: Users and groups management (2 of 2)

- Users and groups that are created by using Integrated Solution Console cannot be modified or deleted in the Process Admin Console
  - However, users and groups can be added as the member of the groups that are created in the Process Admin Console
- Users that are created in Process Admin Console are visible in Integrated Solution Console under “Manage users” but cannot be modified or deleted
  - These users cannot be added as member of the groups that are created in the Integrated Solution Console
- Groups that are created in Process Admin Console are not visible in Integrated Solution Console
  - It is a good practice to use LDAP registry as a single centralized user management console to solve this problem
  - The users and groups that are defined in the LDAP registry are available in the Process Admin Console, and in the Integrated Solution Console
  - They cannot be edited through these consoles but are available for use
  - LDAP must manage all users and groups except for the WebSphere user registry for administrative purposes

Security

© Copyright IBM Corporation 2017

Figure 13-5. IBM Business Process Manager: Users and groups management (2 of 2)

In a typical IBM BPM deployment scenario, Process Center is used in a center, and several Process Servers are attached to it; for example, Process Servers (QA test), Process Servers (staging test), and Process Servers (production).

The administrators of these Process Servers must manually populate and maintain the users and groups for each of these environments separately. The reason is that the groups that are created in Process Center by using the Process Admin Console are not automatically copied in the WebSphere user registry of the attached Process Servers.

To solve this business problem, it is good to be aware of the following practices:

1. Use the LDAP registry as a single centralized user management console for most of your needs.
2. The users and groups that are defined in the LDAP registry are available at both places: in the Process Center and in the Process Server.
3. The users and groups cannot be edited by using these consoles but are available for use.
4. LDAP must manage all the user and groups except the ones that are for administrative purposes in WebSphere user registry.

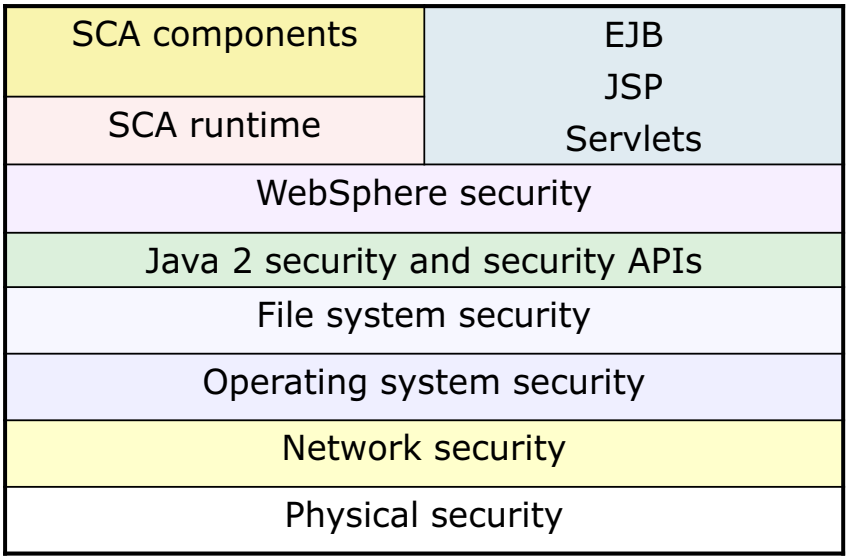
© Copyright IBM Corp. 2017

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

IBM Training

IBM Process Server security overview

IBM Process Server supports all security configuration and management capabilities that WebSphere Application Server offers



Security

© Copyright IBM Corporation 2017

Figure 13-6. IBM Process Server security overview

Several aspects of security need to be considered when securing an environment. WebSphere provides only part of the total security that must be applied.

IBM Process Server security is based on the WebSphere Application Server version security. WebSphere Application Server provides a number of security features and capabilities that IBM Process Server can use. Additionally, external services can be used to secure the environment and applications.

You can find the documentation on the WebSphere Application Server Network Deployment security at: https://www.ibm.com/support/knowledgecenter/SSAW57_7.0.0/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/welc6topsecuring.html

You might also want to review the IBM Redbooks publication *WebSphere Application Server V7.0* Security Guide at: http://www.redbooks.ibm.com/abstracts/sg247660.html?Open

© Copyright IBM Corp. 2017

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

IBM Training

Enabling administrative security

- Determines whether security is used at all
  - A “large switch” that activates a wide variety of security settings for IBM Process Server
- Type of registry against which authentication takes place
  - Access can be restricted by using administrative roles
- Enabled by default
- Console and other administrative tools access is initially restricted to only the primary user ID
  - Other users or groups need to be defined as console users and must be mapped to appropriate roles

Security

© Copyright IBM Corporation 2017

Figure 13-7. Enabling administrative security

Administrative security determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults. Good planning is required because incorrectly enabling administrative security can lock you out of the administrative console or cause the server to end abnormally.

Administrative security can be thought of as a “significant switch” that activates a wide variety of security settings for IBM Process Server. Values for these settings can be specified, but they do not take effect until administrative security is activated. The settings include the authentication of users, the use of Secure Sockets Layer (SSL), and the choice of user account repository. In particular, application security, including authentication and role-based authorization, is not enforced unless administrative security is active. Administrative security is enabled by default.

The administrative security configuration applies to every server within the security domain.

You must activate administrative security before any other security configuration for the WebSphere processes can take effect. Administrative security settings are defined in the global security settings.

You enable, disable, and configure security through the administrative console. After you log on to the administration console, expand **Security > Global Security** in the task pane.

© Copyright IBM Corp. 2017

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

IBM Training

Enable application security

- Enables security for the applications in your environment
- Provides application isolation and requirements for authenticating application users
- Enabled by default
- Is in effect only when administrative security is enabled

Security

© Copyright IBM Corporation 2017

Figure 13-8. Enable application security

Application security enables security for the applications in your environment. This type of security provides application isolation and requirements for authenticating application users.

The administrative security of IBM Process Server is enabled by default. Application security is also enabled by default. Application security is in effect only when administrative security is enabled.

© Copyright IBM Corp. 2017

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

IBM Training

Enabling Java EE security

- Provides a policy-based, fine-grained access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources
- Guards access to system resources such as file I/O, sockets, and properties
  - Also, guards access to web resources such as servlets, JSP files, and EJB methods

Security

© Copyright IBM Corporation 2017

Figure 13-9. Enabling Java EE security

Java 2 security provides a policy-based, fine-grained access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. Java 2 security guards access to system resources such as file I/O, sockets, and properties. It also guards access to web resources such as servlets, JavaServer Pages (JSP) files, and Enterprise JavaBeans (EJB) methods.

Because Java 2 security is relatively new, many existing or even new applications might not be prepared for the fine-grained access control programming model that it can enforce. Administrators must understand the possible consequences of enabling Java 2 security when applications are not prepared for it. Java 2 Security places some new requirements on application developers and administrators.

Select **Use Java 2 security to restrict application access to local resources** to enforce Java 2 security permission checking.

When you enable Java 2 security, an application that requires more Java 2 security permissions than are granted in the default policy might fail to run properly. It runs properly when the required permissions are granted in either the `app.policy` file or the `was.policy` file of the application. Applications that do not have all of the required permissions generate the access control exceptions.

© Copyright IBM Corp. 2017

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

IBM Training

Creating profiles with security (1 of 2)

- During profile creation, all components default to the primary administrative credentials provided
  - Default values provide basic security
- Configure the various components of IBM Process Server to have appropriate security identities
  - Tightens the security
- Use authentication aliases to authenticate the runtime engine component for access to databases and messaging engines
- Three components take the administrator user identity as a default:
  - Service Component Architecture
  - Business Process Choreographer
  - Common Event Infrastructure

Security

© Copyright IBM Corporation 2017

Figure 13-10. Creating profiles with security (1 of 2)

When you create the IBM Process Server profile, default values are used for security credentials. You must configure these security settings on the administrative console after you create the profile.

Three components of IBM Process Server take the administrator user identity as a default. These components are:

- Service Component Architecture (SCA)
- Business Process Choreographer
- Common Event Infrastructure (CEI)

The identities that are associated with these components are used to create authentication aliases that are required when security is enabled. It is important to change these identities to appropriate users from your account repository.

© Copyright IBM Corp. 2017

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

IBM Training

Creating profiles with security (2 of 2)

- Aliases can be modified on the Business Integration Security page of the administrative console
- Provide an appropriate user name and password to use as an authentication alias

© Copyright IBM Corporation 2017

Figure 13-11. Creating profiles with security (2 of 2)

For each of the Service Component Architecture and Business Process Choreographer authentication aliases, provide an appropriate user name and password to use as an authentication alias. In some cases, the Alias column might not provide a link, in which case you select the check box in the Select column corresponding to the alias that you want to edit, and click Edit.

© Copyright IBM Corp. 2017

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

IBM Training

Configuring a user account repository (1 of 2)

- User names and passwords of registered users are stored in a user account repository (registry)
- Local operating system repository
  - The default registry
  - Do not use in a network deployment environment
- Federated repository
  - Enables the use of multiple repositories under a single realm
- LDAP repository
  - Uses LDAP as a user registry
- Custom user registry
  - Allows custom implementation of user registry
  - Database, flat file, operating system-based with more custom logic, or other registries that are not directly supported

Security

© Copyright IBM Corporation 2017

Figure 13-12. Configuring a user account repository (1 of 2)

The user names and passwords of registered users are stored in a user account repository. You can use the user account repository of the local operating system (user account repository is the default), Lightweight Directory Access Protocol (LDAP), federated repositories, or a custom account repository, as follows:

- **Federated repositories**

  Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in:

  - The file-based repository that is built into the system
  - One or more external repositories
  - Both the built-in, file-based repository and in one or more external repositories

  Note: Only a user with administrator privileges can view the federated repositories configuration.

- **Local operating system**

  The local operating system is the default user registry. Do not use the local operating system as the user registry in a network deployment environment.

- **Lightweight Directory Access Protocol (LDAP)**

© Copyright IBM Corp. 2017

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

To access a user registry by using LDAP, you must know the following information:

- A valid user name (ID) and password of the user
- The server host, and port information of the registry server
- The base distinguished name (DN) and if necessary, the bind DN and the bind password

If you configure IBM Process Server to use more than one LDAP server, be careful to note: the value that is shown in the **Global security > stand-alone LDAP Registry** page of the administrative console might not reflect the LDAP server that is used.

- **Custom user registry**

  From the administrative console, open the configuration page for your user registry. Expand **Security**, click **Global security**, and select the user registry that you are employing under the **Available realm definitions** menu. Click **Configure**.

  Optional: Enter a valid user name in the **Primary administrative user name** field. This value is the name of a user with administrative privileges that is defined in the registry. This user name is used to access the administrative console. The `wsadmin` command also uses it.

  Select either the **Automatically generated server identity** or the **Server identity that is stored in the repository** option.

  If you select **Automatically generated server identity**, the application server generates the server identity that is used for internal process communication. You can change this server identity on the Authentication mechanisms and expiration page. To access the Authentication mechanisms and expiration page, click **Security > Global security > Authentication mechanisms and expiration**. Change the value of the **Internal server ID** field.

  If you select the **Server identity that is stored in the repository** option, enter the following information:

  - For **Server user ID or administrative user on a Version 7.0 node**, specify a user ID that is used to run the application server for security purposes.
  - For **Password**, specify the password that is associated with this user.

  Optional: For stand-alone custom registries only, do the following steps:

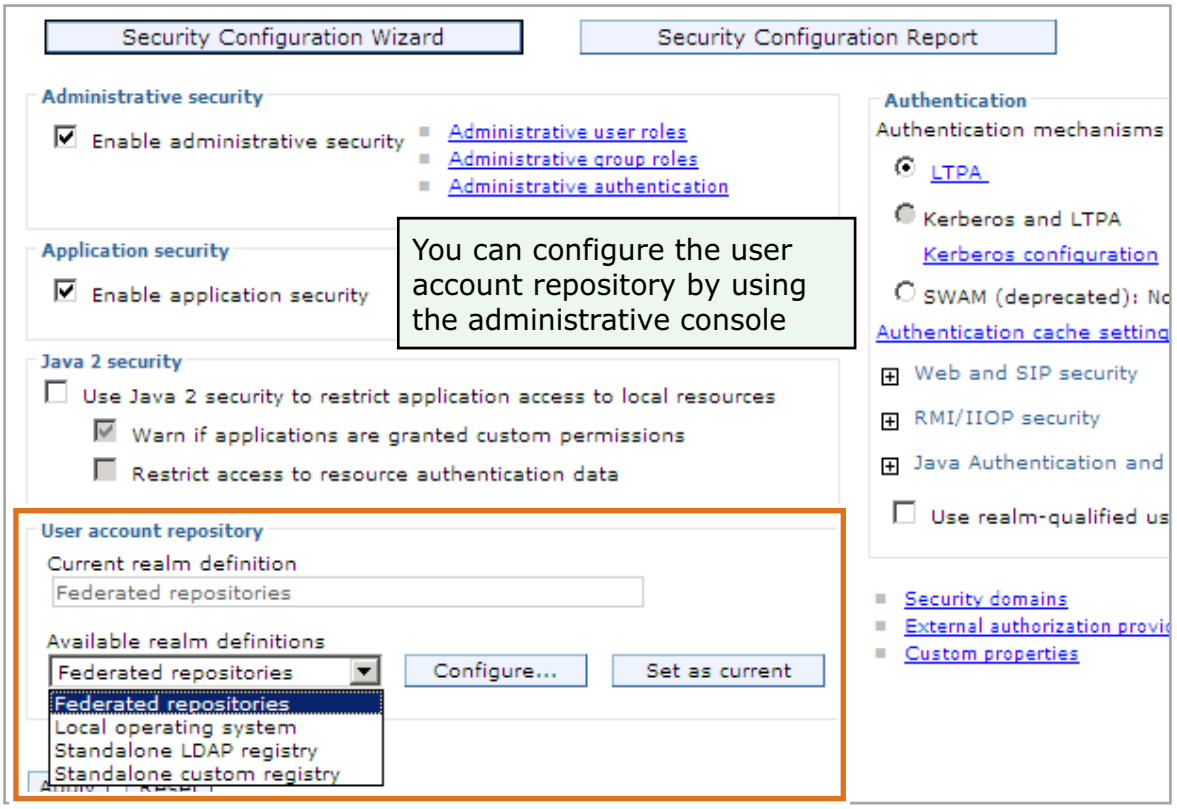
  1) Verify that the value in the **Custom registry class name** is correct, or change it if necessary.
  2) Select or remove the check from **Ignore case for authentication**. When you select this option, the authorization check is not case-sensitive.
  3) Click **Apply**. From the bottom of the page, click **Set as current**. Click **OK** and either **Apply** or **Save**.

© Copyright IBM Corp. 2017

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

IBM Training

Configuring a user account repository (2 of 2)



Security © Copyright IBM Corporation 2017

Figure 13-13. Configuring a user account repository (2 of 2)

You can configure your user account repository by using the administrative console. The steps for configuring the local operating system, which is the default, or a stand-alone custom user account registry are similar.

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

IBM Training

Security domains

- Security domain provides the flexibility to use different security configurations, such as the user registry, for different applications
- Multiple security domains are supported
  - Can configure different security attributes for both administrative and user applications within a cell environment
  - Can configure different applications to use different security configurations by assigning the servers, clusters, or service integration buses to the security domains
- Only users that are assigned to the administrator role can configure multiple security domains

Security

© Copyright IBM Corporation 2017

Figure 13-14. Security domains

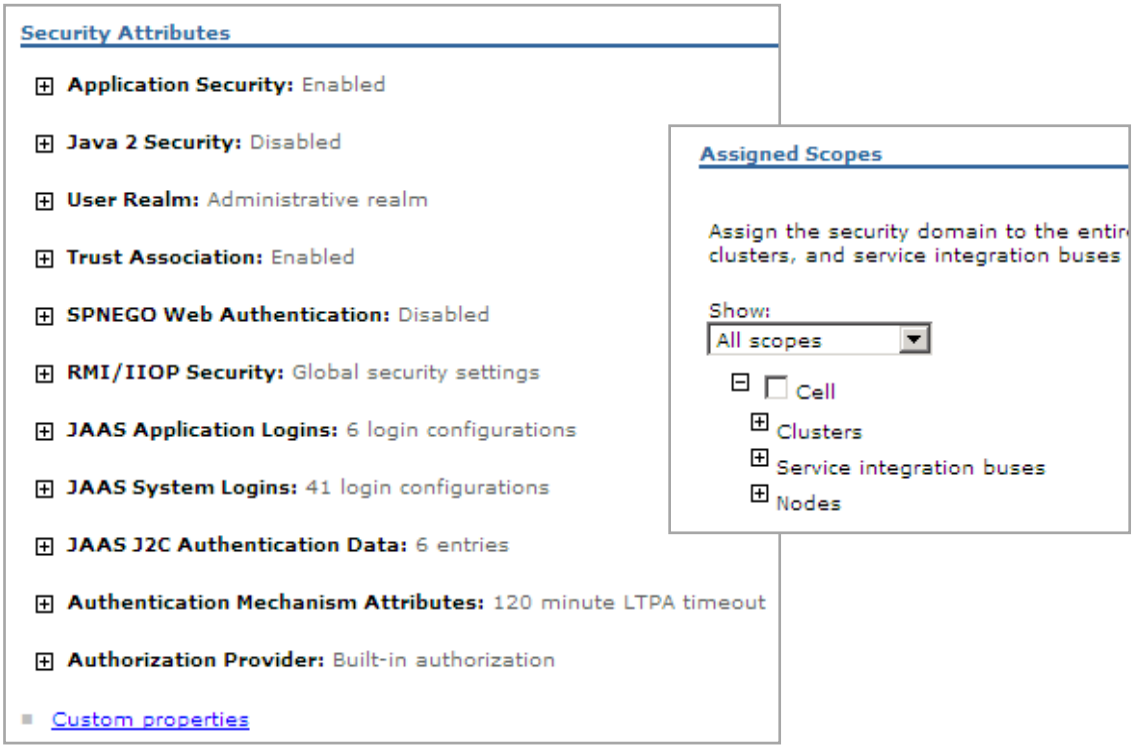
A security domain consists of all servers that are configured with the same user registry realm name. The basic requirement for a security domain is as follows: the access ID that is returned from the registry on one server within the security domain is the same access ID that is returned from the registry on any other server within the same security domain. Configuration of global security for a security domain consists of configuring the common user registry, the authentication mechanism, and other security information that defines the behavior of a security domain.

© Copyright IBM Corp. 2017

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

IBM Training

Security domain configuration



Security

© Copyright IBM Corporation 2017

Figure 13-15. Security domain configuration

You configure security domains from the administration console (from the task pane, select **Security > Security domains**). Assign a name to the domain, and then you can configure the domain as required. A number of security attributes can be altered and also the scope to which the domain applies.

© Copyright IBM Corp. 2017

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

13.2. Application security overview

© Copyright IBM Corp. 2017

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

IBM Training

Application security overview

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 13-16. Application security overview

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

IBM Training

Elements of application level security: Authentication

- Authentication: Identifies who is requesting access to a resource
  - Mechanism by which clients (users or another host) provide proof that "they are who they say they are"
  - "Challenges" are used to offer the client the ability to provide proof
  - Most common challenge is user name and password
  - Digital signature is another example
  - When administrative security is turned on, clients must be authenticated
  - Some of the components of the IBM Process Server infrastructure have authentication aliases that are used to authenticate the runtime code for access to databases and the messaging engine
  - Some runtime components use message-driven beans (MDBs) that are configured with a **RunAs** role

Security

© Copyright IBM Corporation 2017

Figure 13-17. Elements of application level security: Authentication

Authentication is the process of determining that a user is who the user claims to be. After the authentication is done, a token of some type is used to indicate that the user was authenticated. Authorization is the process of determining whether the user is permitted to access the requested resource. Clearly, authentication must be done before authorization can be done.

© Copyright IBM Corp. 2017

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

IBM Training

Elements of application level security: Authorization

- Authorization: Involves granting permission to run actions on resources
  - Determines what the authenticated user is allowed to do
  - “Can a call center agent submit a credit application?”
  - “Can a teller approve a loan application?”
- Control access to resources
  - Security lookup (by server): Determine security privileges; information that is stored in registry
  - Rule enforcement (by server): Obtain rules from registry; given privileges of principal and rules, determine access
- Provide access control for components by using Service Component Architecture qualifiers at development time

Security

© Copyright IBM Corporation 2017

Figure 13-18. Elements of application level security: Authorization

When a general user is authenticated to IBM Process Server, it is important for security that not every possible operation is available to that user. Allowing some users to do certain tasks, while denying these tasks to other users, is termed access control.

Access control can be arranged for components that you develop to make them secure. You provide access control for components by using service component architecture qualifiers at development time.

© Copyright IBM Corp. 2017

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

IBM Training

Elements of application level security: Data integrity and privacy (1 of 2)

- Data integrity: The data that an application access is secured at origin, destination, and in transit
  - Data that is sent over the network cannot be altered in transit
- Privacy and confidentiality: Data that is sent over the network cannot be intercepted and read in transit
- Secure Sockets Layer (SSL) protocol:
  - SSL provides data integrity and privacy by authenticating the endpoints and exchanging information
  - Supports encryption, decryption, and digital signatures

Security

© Copyright IBM Corporation 2017

Figure 13-19. Elements of application level security: Data integrity and privacy (1 of 2)

The privacy and integrity of data that is accessed when IBM Process Server processes are started is critical to your security.

- **Privacy** means that it is not possible for an unauthorized user to intercept and read data.
- **Integrity** means that it is not possible for an unauthorized user to alter data.

Secure Sockets Layer (SSL) protocol uses a handshake to authenticate the endpoints and exchange information that is used to generate the session key that the endpoints for encryption and decryption use. SSL is a synchronous protocol and is suitable for point-to-point communication. SSL requires that the two endpoints maintain a connection with each other during the SSL session.

© Copyright IBM Corp. 2017

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

IBM Training

Elements of application level security: Data integrity and privacy (2 of 2)

- WS-Security
  - Standard that defines SOAP extensions for securing SOAP messages
  - Adds support for authentication, integrity, and privacy for a single SOAP message
  - Unlike SSL, no handshake to establish a session key between the two endpoints
  - Suitable for securing messages in an ***asynchronous*** environment (for example, SOAP or JMS)
- Propagation of identity (single sign-on):
  - When a client request needs to flow through several systems within the enterprise, the client is not forced to provide authentication data multiple times
  - Is used to propagate the authentication information to other systems, which can, in turn, apply access control

Security © Copyright IBM Corporation 2017

Figure 13-20. Elements of application level security: Data integrity and privacy (2 of 2)

The WS-Security standard defines SOAP extensions for securing SOAP messages. WS-Security adds support for authentication, integrity, and privacy for a single SOAP message. Unlike SSL, no handshake is required for establishing a session key. It makes WS-Security suitable for securing messages in an asynchronous environment, such as SOAP over Java Message Service (JMS) or SOAP over Service Integration Bus (SIB). WS-Security deployment descriptors can be set in your applications before deployment. See related information for more details.

The final element of security is one of the propagations of identity, which is achieved through single sign-on.

When a client request must flow through several systems within the enterprise, the client is not forced to provide authentication data multiple times. The single sign-on method is used to propagate the authentication information to downstream systems, which can, in turn, apply access control.

A client is asked to provide user name and password information only once. The provided identity propagates throughout the system.

When a client request flows through multiple systems within the enterprise, the client must authenticate only once. This concept of identity propagation is solved by using a single sign-on method.

© Copyright IBM Corp. 2017

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

IBM Training

Supported authentication mechanisms

WebSphere Application Server supports several authentication mechanisms:

- Lightweight Third Party Authentication (LTPA)
  - Single token mechanism
  - For application server environments distributed across multiple systems and environments
  - Provides single sign-on (SSO) feature
- Kerberos (KRB5)
  - Single token mechanism
  - Enables end-to-end interoperability with other applications that support Kerberos authentication
  - Java client can participate in the Kerberos SSO by using the Kerberos credential to authenticate to WebSphere Application Server
- RSA token authentication
  - Used for the administrative agent only
  - Used for processing administration requests

Security

© Copyright IBM Corporation 2017

Figure 13-21. Supported authentication mechanisms

WebSphere Application Server supports three means of authenticating clients:

- Lightweight Third Party Authentication (LTPA) allows single sign-on (SSO), which allows a user to authenticate only once for all WebSphere applications in a cell.
- Kerberos authentication: A Java client that participates in Kerberos single sign-on authentication can use the Kerberos credential to authenticate against the WebSphere Application Server, rather than a user ID and password.
- Rational Software Architect authentication is used only to authenticate administrative requests; it cannot be used for general application security.

© Copyright IBM Corp. 2017

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

IBM Training

Application security overview

- Application security can be declarative or programmatic
  - **Declarative**: Security policies are configured independent of the logic of the application code, by using security information that metadata specifies, either through annotations in the application or in the EJB deployment descriptor
  - **Programmatic**: Security policies are enforced in the application by using the Java Enterprise Edition security application programming interface (API)
- Deploying a secure application is similar to deploying a non-secured enterprise application
  - During deployment, administrators can do the role mapping for users and groups, and also the run-as mapping

Security

© Copyright IBM Corporation 2017

Figure 13-22. Application security overview

Application security provides application isolation and the requirements for authenticating users of the applications. Applications can be secured declaratively or programmatically.

When an application that is deployed in the application server uses the declarative security, application security must be enabled for the runtime environment. Application security does not have to be enabled for applications that rely only on programmatic security.

© Copyright IBM Corp. 2017

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

IBM Training

Authentication for web applications

- Web applications typically require both authentication (who the user is) and authorization (whether the user is authorized to access a valid resource)
- Using declarative security for web applications:
  - Security information (security roles, access constraints, authentication requirements) is contained in external XML web deployment descriptor file
  - No security API calls are hardcoded in the servlets or JSPs
  - The web container enforces security, and it is based on the requirements that the descriptor specifies
- Using programmatic security for web applications:
  - Application is security-aware
  - Contains calls to security APIs that allow the application developer to make more security decisions at run time

Security

© Copyright IBM Corporation 2017

Figure 13-23. Authentication for web applications

A typical web application consists of both web and EJB modules. Web components build the user interface to the application business logic. A web application can be accessed over the network by many users who do different roles and require access to different functions.

WebSphere Application Server supports the authentication mechanisms that the Java Platform, Enterprise Edition specification requires, including HTTP basic and form-based or client certificates. In addition, it offers Kerberos support and support for third-party authentication proxies.

© Copyright IBM Corp. 2017

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

IBM Training

Authentication for Enterprise JavaBeans applications

- Java EE applications usually consist of both web and Enterprise JavaBeans (EJB) modules
- EJB components implement business logic and can have access to sensitive data
  - They need to be protected since remote clients can invoke them
- Using declarative security for EJB applications:
  - Allows security policies to be configured without being apparent to the application code
  - Security information is specified through metadata by using annotations or EJB deployment descriptors
  - No security rules are hardcoded in the application code
  - The application server infrastructure enforces security
- Using programmatic security for EJB applications:
  - The developer can obtain details about the authenticated user, including the user's identity and security role
  - Useful when declarative security is not enough to fulfill the application security requirements

Security © Copyright IBM Corporation 2017

Figure 13-24. Authentication for Enterprise JavaBeans applications

Two types of Enterprise JavaBeans are in EJB 3.0:

- Session beans that represent business functions and services
- Message-driven beans (MDBs) that can process asynchronous messages that are sent to the Java Platform, Enterprise Edition application

The server infrastructure calls the message-driven beans and has no client credentials. For this reason, the explanation of EJB application security throughout the rest of the unit applies only to session beans. The client calls only the session beans. To call a secured EJB from a message-driven bean, you use delegation or the JAAS API.

With Java EE applications that use declarative security, the application server infrastructure enforces this security. The EJB container retrieves the security policy from the descriptor to get the security roles that are allowed to run bean methods. The container uses the security context that is associated with the method call to determine whether the user has the required role.

- If you must allow only a certain class of user (security role) access to specific application functions (methods of the enterprise bean), you might have to use only declarative security.
- If declarative security alone cannot fulfill the security needs of an application, you can use programmatic security to make access decisions.

© Copyright IBM Corp. 2017

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

- If you must switch caller identity to call an EJB, or you must call a secured bean from an unsecured one (such as an MDB), you can use delegation.
- If your security requirements are complicated, Java Platform, Enterprise Edition security might not be sufficient. Consider a third-party security framework or an external Java Authorization Contract for Containers (JACC) provider.

© Copyright IBM Corp. 2017

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

IBM Training

Role-based resource security

- Role-based resource security provides authorization to the IBM Process Server resources
- Business process management (BPM) roles govern the operations of an administrator
  - BPMadmin
  - BPMrolemanager
- Resource roles that govern the resource: owner, writer, or reader
- Security manager can be used to assign one of three resource roles to members of an organization
- Roles determine level of access to resources
- Role-based access control is available for the business calendar manager

Security

© Copyright IBM Corporation 2017

Figure 13-25. Role-based resource security

When a general user is authenticated to IBM Process Server, it is important for security that not every possible operation is available to that user. Allowing some users to do certain tasks, while denying these tasks to other users, is termed *access control*.

Access control can be configured for components that you develop to make them secure. You provide access control for components by using service component architecture qualifiers at development time.

Some IBM Process Server components that are packaged as enterprise archive (EAR) files secure their operation by using Java EE role-based security. In contrast to code-based security, which secures the operation of components, role-based access control secures *resources*. For example, in the Business Calendars widget, you can specify the type of access that users have for the individual timetables.

© Copyright IBM Corp. 2017

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

IBM Training

System and resource roles

System roles that are created after installation of IBM Process Server:

- **BPMadmin** has the authority to add and remove members from the **BPMrolemanager** role
  - Initially set to primary administrative user
- **BPMrolemanager** has authority to add and remove members from resource roles: owner, writer, and reader
  - Initially set to primary administrative user

Resource roles:

- Resource roles control access to a specific instance of a resource
- Access control at role level instead of individual user level
- Three resource-level roles: owner, writer, and reader
- Permission for all roles is set initially to **All Authenticated**

Security

© Copyright IBM Corporation 2017

Figure 13-26. System and resource roles

© Copyright IBM Corp. 2017

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

IBM Training

Owner, writer, and reader resource roles

- Owner role can read and write to the resource and can assign other members to the writer and reader roles
- Writer role can read and write to the resource
- Reader role can read the resource
- You can change the membership of a role
- You cannot change the name of a role
- You cannot add or delete a role
- You cannot change the permissions that are associated with a role

Security

© Copyright IBM Corporation 2017

Figure 13-27. Owner, writer, and reader resource roles

© Copyright IBM Corp. 2017

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

13.3. SCA security qualifiers

© Copyright IBM Corp. 2017

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

IBM Training

SCA security qualifiers

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 13-28. SCA security qualifiers

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

IBM Training

Using security-related quality of service qualifiers

- Two quality of service qualifiers that are related to security can be used to declare security requirements in SCA applications:
  - **Security permission**: Specifies a role that a specified type of users must possess to use an operation in an interface
  - **Security identity**: Specifies an identity under which an implementation runs
- These qualifiers are not provided by default
  - Must be added by using IBM Integration Designer or added to the SCDL component file

Security

© Copyright IBM Corporation 2017

Figure 13-29. Using security-related quality of service qualifiers

The qualities of service qualifiers are used to define a set of communication characteristics that an application requires, such as transmission priority, level of route reliability, transaction management, and security level. Two QoS qualifiers can be used for security purposes: security permission and security identity.

© Copyright IBM Corp. 2017

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

IBM Training

Security permission qualifier (1 of 3)

- The **security permission** qualifier is used in processes that involve human tasks
- The identity of the caller must have the designated role to call the interface or operation
- If no security permission is specified, no permissions are checked and all callers can call the interface or operation
- Can apply the qualifier at three levels of a component or import:
  - For all of its interfaces
  - For an individual interface
  - For an individual operation of an interface

Security

© Copyright IBM Corporation 2017

Figure 13-30. Security permission qualifier (1 of 3)

© Copyright IBM Corp. 2017

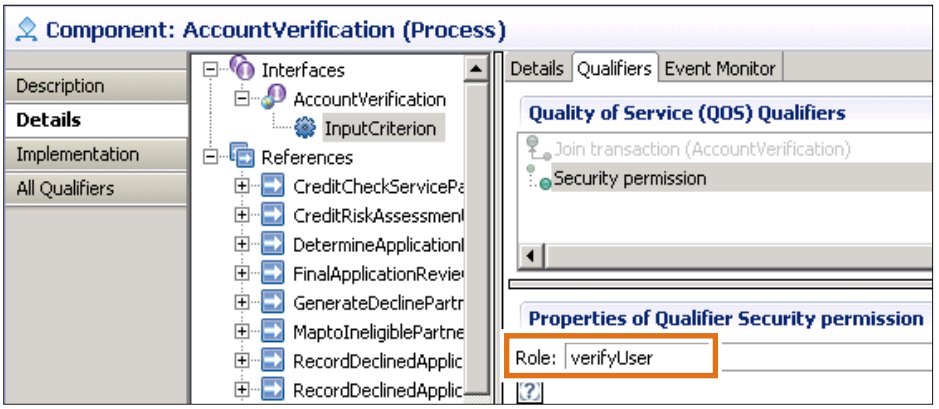
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Security permission qualifier (2 of 3)

- In this example, a user must have the *verifyUser* role to be able to invoke the InputCriterion operation of the AccountVerification interface

```
"<interface xsi:type="wsdl:WSDLPortType"
          portType="ns1:CreditRequest">
   <method name="calulateCreditScore">
   <scal:interfaceQualifier xsi:type="scdl:SecurityPermission"
          role="operator"/>
   </method>
</interface>"
```



© Copyright IBM Corporation 2017

Figure 13-31. Security permission qualifier (2 of 3)

You can use the **security permission** qualifier at the interface level. You use it to specify a role (a semantic grouping of permissions) that a particular type of users must have for starting an operation in an interface. When this qualifier is specified, the identity of the caller must have this role to be able to call the interface or operation. If no security permission is specified, no permissions are checked and all callers are permitted to call the interface or operation.

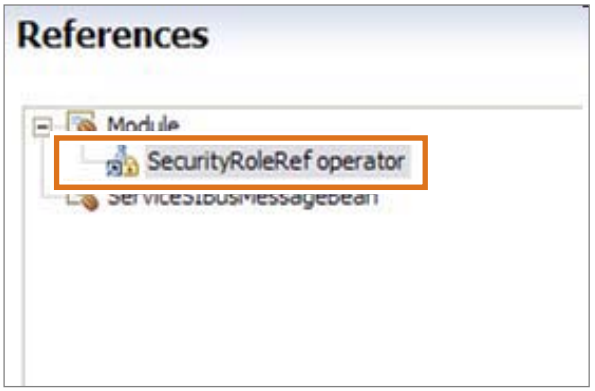
© Copyright IBM Corp. 2017

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

IBM Training

Security permission qualifier (3 of 3)

- When the security permission qualifier is specified, a security role reference is added to the Module bean in the SCA EJB project



© Copyright IBM Corporation 2017

Figure 13-32. Security permission qualifier (3 of 3)

© Copyright IBM Corp. 2017

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

IBM Training

Security identity qualifier (1 of 4)

- The **security identity** qualifier specifies the identity under which the implementation runs
- Equivalent to the **RunAs** role used for delegations in WebSphere Application Server
- An implementation qualifier
- When the component implementation is invoked, the implementation logic runs with the privileges specifics for that particular role

Security

© Copyright IBM Corporation 2017

Figure 13-33. Security identity qualifier (1 of 4)

You can use the **security identity** qualifier at the implementation level. You use this qualifier to provide a logical name (alias) for the identity under which the implementation runs at run time. If this qualifier is not specified, then the implementation runs under the identity of its caller, or under the hosting identity of the container if no caller identity is present. Roles are associated with the identity, and the roles dictate whether the implementation is authorized to start other components.

© Copyright IBM Corp. 2017

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

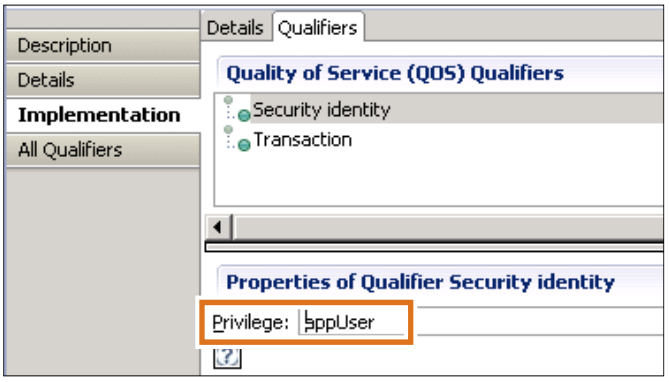
IBM Training

Security identity qualifier (2 of 4)

- In this example, the component runs under the logical name “appUser”

```
"<implementation
xsi:type="java:JavaImplementation"
class="sca.component.java.impl.CreditApprovalImpl">

    <scdl:implementationQualifier
         xsi:type="scdl:SecurityIdentity"
         privilege="admin"/>
 </implementation>"
```



© Copyright IBM Corporation 2017

Figure 13-34. Security identity qualifier (2 of 4)

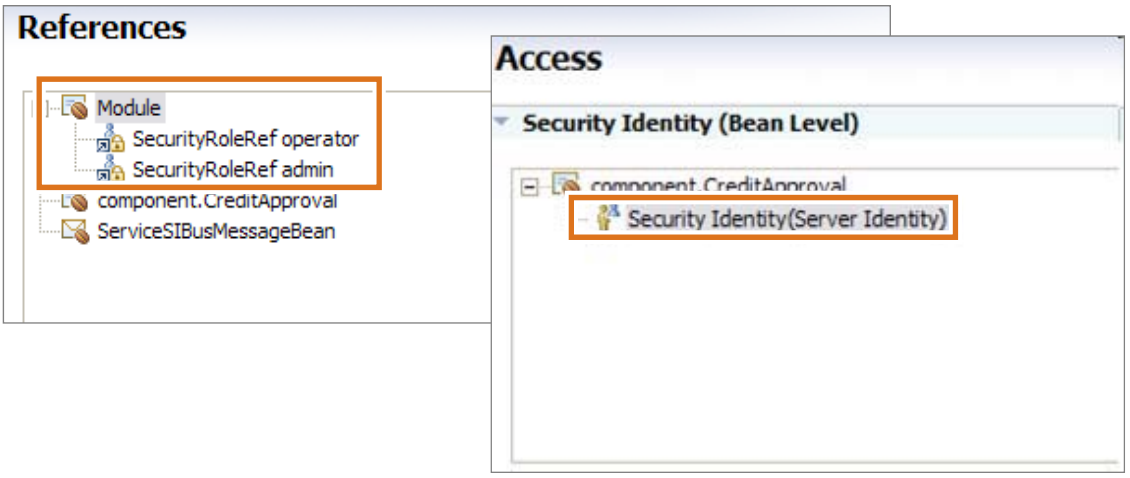
© Copyright IBM Corp. 2017

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

IBM Training

Security identity qualifier (3 of 4)

- When the security identity qualifier is specified:
  - A second module bean is added to the EJB project named **`component.<component name>`**
  - Another security role reference is added to the module bean in the SCA EJB project
  - In the EJB project, an entry for the Security Identity can be found



© Copyright IBM Corporation 2017

Figure 13-35. Security identity qualifier (3 of 4)

© Copyright IBM Corp. 2017

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

IBM Training

Security identity qualifier (4 of 4)

Two subject holders on the thread of execution can be checked for an authenticated subject at any point:

- Caller subject
  - Is the inbound subject and an application does not set it
  - Is used to authorize the running code
  - The running code might make downstream calls
- Invocation subject
  - Downstream calls are invoked by using the invocation subject
  - Is also known as the RunAs subject
  - Is set based on the value of the security identity qualifier

Security

© Copyright IBM Corporation 2017

Figure 13-36. Security identity qualifier (4 of 4)

© Copyright IBM Corp. 2017

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

IBM Training

Security context propagation

- SCA can propagate security credentials across asynchronous invocations
- Two SCA handlers to accomplish this task:
  - SourceSecurityContextHandler: Responsible for retrieving the authorization and authentication information from the subject and serializing the values to the SCA message in the form of a byte array
  - TargetSecurityContextHandler: Responsible for retrieving the authorization and authentication byte arrays from the SCA message and restoring the subject

Security

© Copyright IBM Corporation 2017

Figure 13-37. Security context propagation

© Copyright IBM Corp. 2017

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

13.4. Securing business processes

© Copyright IBM Corp. 2017

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

IBM Training

Securing business processes

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 13-38. Securing business processes

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

IBM Training

Securing business processes: Roles

- Java Enterprise Edition roles:
  - **BPESystemAdministrator**: Users that are assigned to this role have all privileges
  - **BPESystemMonitor**: Users that are assigned to this role can view the properties of all business process objects
  - **JMSAPIUser**: Business Flow Manager JMS API requests are run on behalf of the user ID; this role is mapped to, regardless of who the caller is
- Instance-based roles:
  - **Process starter**: View the properties of the process instance
  - **Process reader**: View the properties of the process instance, also becomes the reader
  - **Scope reader**: View properties and variables
  - **Administrator**: Administer process instance
  - **Process activity administrator**: Repair activities in the process
  - **Scope administrator**: Administer the activities in the scope

Security

© Copyright IBM Corporation 2017

Figure 13-39. Securing business processes: Roles

A role is a set of people who share a level of authorization. Actions that you can take on business processes depend on your authorization role. This role can be a Java EE role or an instance-based role.

Java EE roles are set up when Business Process Choreographer is configured. For Java EE role-based authorization, you must have a configured user registry and application security enabled.

The following Java Platform, Enterprise Edition (Java EE) roles are supported for processes:

- BPESystemAdministrator: Users that are assigned to this role have all privileges. This role is also known as the system administrator for business processes.
- BPESystemMonitor: Users that are assigned to this role can view the properties of all business process objects. This role is also known as the system monitor for business processes.
- JMSAPIUser: Business Flow Manager JMS API requests are run on behalf of the user ID, which this role is mapped to, regardless of who the caller is.

You can use the administrative console to change the assignment of users and groups to these roles.

© Copyright IBM Corp. 2017

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

A set of predefined authorization roles is provided for business processes and activities. You can assign these roles when you model the process. The association of users to instance-based roles is determined at run time by using people resolution.

If process administration is restricted to system administrators, then instance-based administration is disabled. The result is that administrative actions on processes, scopes, and activities are limited to users in the BPESystemAdministrator role. In addition, only the users in the BPESystemAdministrator or BPESystemMonitor roles can read, view, and monitor a process instance or parts of it.

© Copyright IBM Corp. 2017

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

IBM Training

Work items: Roles to operation mapping for processes and activities

	Comment
Administrator	Person who is allowed to administer a process
Reader	Person who is allowed to see process data
Starter	Person who creates and starts the process instance

	Human task activity (also known as staff activity)	Receive, pick, event handler	Comment
Potential owner	X	-	Person who can claim and work with tasks
Owner	X	-	The person who claimed the task
Potential starter	-	X	Person who is allowed to start an event
Administrator	X	X	Person who is allowed to administer a task
Editor	X	-	Person who is allowed to edit task data
Reader	X	X	Person who is allowed to see task data

Security

© Copyright IBM Corporation 2017

Figure 13-40. Work items: Roles to operation mapping for processes and activities

© Copyright IBM Corp. 2017

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

IBM Training

Securing business processes: Create and start processes

- The task that determines an authorization for the users:
  - The invocation task that is associated with the receive or pick (receive choice) activity is used to create and start a new process instance
  - The administration task that is associated with the process
- Use human tasks to create and start business processes:
  - Assign an inline invocation task to the initiating receive or pick (receive choice) activity of the process
  - Assign a stand-alone invocation task to the initiating receive or pick (receive choice) activity of the process

Security

© Copyright IBM Corporation 2017

Figure 13-41. Securing business processes: Create and start processes

The invocation task that is associated with the receive or pick (receive choice) activity determines the group of users who are allowed to create and start a process. These activities are used to create and start a new process instance, and also by the administration task that is associated with the process. The business process inherits the roles that you assign to these tasks.

In addition, when the Service Component Architecture (SCA) client calls the process, you can restrict the set of users who are allowed to start the process. You do this restriction by setting specific SCA security qualifiers when you install the process.

You can use human tasks in the following ways to create and start business processes:

- Assign an inline invocation task to the initiating receive or pick (receive choice) activity of the process.

  Some business processes might change sensitive business data; therefore, only authorized personnel can be authorized to create and start these processes. For this type of business process, you can assign a human task to the initiating receive activity of the process by specifying an inline invocation task for the process template. The potential starters that are defined for the inline invocation task become the potential starters of the process.

  The process can be started either by using the Human Task Manager API to create and start the invocation, or by using the Business Flow Manager API to initiate the process. Both

© Copyright IBM Corp. 2017

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

methods result in the same authorization checks. If an inline task is not specified, anyone can start the process.

- Assign a stand-alone invocation task to the initiating receive or pick (receive choice) activity of the process.

  You can also use a stand-alone invocation task that is wired to the business process to do authorization checks when a process is started. However, consider the following points if you use a stand-alone invocation task:

  - The authorization check is done only if the invocation task starts the process. The result is that the check is omitted when the process is started by using the Business Flow Manager API or an SCA client that is directly wired to the process component.
  - It uses the SCA infrastructure to start the process while the Business Flow Manager directly calls an inline task.
  - You have no access to the process context in the people assignment criteria definition. The result is that stand-alone tasks do not support dynamic people assignments that are based on the process context.

If an administration task is assigned to the process, the process inherits the administrator role of the administration task. A process administrator can do various actions on the process, including creating and starting a process instance.

© Copyright IBM Corp. 2017

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

IBM Training

Securing business processes: Interact with processes

- The tasks that determine an authorization for the users:
  - The invocation task that is associated with the receive or pick activity or event handlers
  - By the administration task that is associated with the process
- Use human tasks to interact with a process instance:
  - Assign an inline invocation task to the receive activity, pick activity, or event handler
  - Use a stand-alone human task to secure inbound operations of a business process
  - Assign an administration task to the process

Security

© Copyright IBM Corporation 2017

Figure 13-42. Securing business processes: Interact with processes

A long-running process can have multiple receive activities, pick (receive choice) activities, and event handlers. These activities are served by submitting a request to the appropriate operation of the corresponding process instance. The process instance is identified implicitly by providing a unique correlation set instance in the request according to the correlation set defined in the process model.

A receive or pick activity can be used to create a process instance. So, interacting with an existing process instance by submitting a request to a process is similar to starting a new process instance.

The following tasks determine the set of users that are authorized to submit a request to a process instance:

- Invocation task that is associated with the receive or pick activity, or the event handlers
- Administration task that is associated with the process

You can use human tasks in the following ways to interact with a process instance:

- Assign an inline invocation task to the receive activity, pick activity, or event handler.

  The potential starters that are defined for the inline invocation task submit a request to the corresponding operation of the process. The invocation task is optional. If an invocation task is not defined, everyone is authorized to submit a request.

© Copyright IBM Corp. 2017

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

You can also use a stand-alone human task to secure inbound operations of a business process. The same rules and restrictions apply as for stand-alone invocation tasks for process-creating operations.

- Assign an administration task to the process.

  The process inherits the administrator role of the administration task. A process administrator can interact with a process by using its operations.

  If an administration task is not specified for the process, the starter of the process becomes the process administrator. In this case, the process starter is allowed to submit requests to operations of the process instance.

Consider a process that uses the same operation in different receive or pick (receive choice) activities, or event handlers, and the receiving process instance is not expecting a request. The corresponding receive or pick (receive choice) activity is not yet waiting or the event handler is not yet active. As a result, the user who is sending the request must be authorized to send a request to all of these activities and event handlers; otherwise, the request is rejected.

© Copyright IBM Corp. 2017

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

IBM Training

Securing business processes: Administer processes

- Use administration tasks to authorize a user, or group of users, to do administrative actions on business processes and their associated activities
- Process administration
  - Specify an administration task as part of a long-running business process
  - The administrator and reader roles for the administration task determine who the process administrator and the process reader are
- Scope administration
  - Model an administration task for the scope that defines scope readers and scope administrators
- Activity administration
  - Reader and administrator roles of the process, and reader and administrator roles of the enclosing scopes, are automatically propagated to the activities

Security

© Copyright IBM Corporation 2017

Figure 13-43. Securing business processes: Administer processes

You can use administration tasks to authorize a user, or group of users, to do administrative actions on business processes and their associated activities.

Note: If you are using one of the alternative process administration modes, administration tasks might not be created. In addition, administrative actions on processes, scopes, and activities might be limited to users in the BPESystemAdministrator role.

- **Process administration**

  To define which users are allowed to do administrative actions and to read the process data, you can specify an administration task as part of a long-running business process. The administrator and reader roles for the administration task determine who the process administrator and the process reader are. The process administrator can, for example, terminate the process instance.

  An administration task is associated with every business process. If an administration task is not modeled for the process, a default administration task is created at run time. This default task defines the process starter as the process administrator, and does not assign any readers to the process.

- **Scope administration**

© Copyright IBM Corp. 2017

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

You can model an administration task for the scope that defines scope readers and scope administrators. A scope reader is allowed to view local variables. Scope administrators are allowed to repair activity instances in the scope, and to view and update local variables. If the scope is enclosed in another scope, the enclosing scope inherits the scope reader and administration rights. Scope readers and administrators also become readers and administrators of the activities in the scope.

- **Activity administration**

  The administrator role for an activity administration task determines who is allowed to administer the corresponding activity. The activity administrator can, for example, restart the activity. The administration task is created as soon as administrative actions, that is, restart or complete, can be done on the activity instance. Reader and administrator roles of the process, and reader and administrator roles of the enclosing scopes, are automatically propagated to the activities.

  In addition, you can model administration tasks for activities in the following ways:

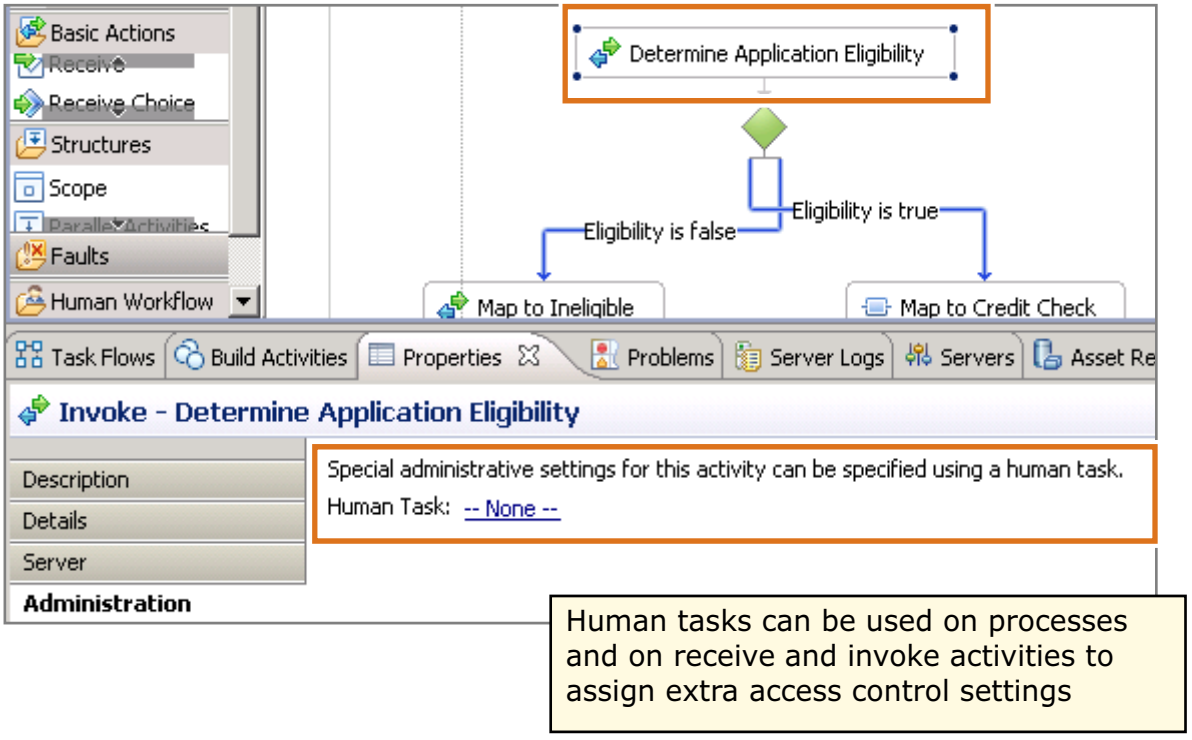
  - For each start or snippet activity, the administration task determines who is allowed to administer the activity in addition to the process administrators.
  - Use a default administration task for activities on the process level that applies to any activity that does not have an assigned administration task.

© Copyright IBM Corp. 2017

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

IBM Training

Assigning administrative tasks to processes and activities



Security

© Copyright IBM Corporation 2017

Figure 13-44. Assigning administrative tasks to processes and activities

A human task can be used to set permissions for any start or receive activity. By defining a human task on a receive activity, you are restricting to only certain users or services the capability of sending a message to this BPEL. Similarly, using a human task, you can restrict the communication between the BPEL and a reference partner.

© Copyright IBM Corp. 2017

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

IBM Training

Change process ownership

When creating and starting a long-running process instance, the security context of the process starter is stored with the instance and used for subsequent service invocations

- Repair scenario
  - Allows changing the ownership of a particular process instance; for example, in cases where the process starter is not with the company any more
- The new Business Flow Manager EJB API method *claimProcessOwnership* (PIID processInstanceID) allows a person with process administrator authorization to take on the ownership of a specific process instance
  - The current security context (security context on the thread) replaces the security context that is stored with the process instance
  - Subsequent service invocations are executed by using the new security context
- Is also exposed in the Business Process Choreographer Explorer

Security

© Copyright IBM Corporation 2017

Figure 13-45. Change process ownership

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain the role of security quality of service qualifiers in SCA applications
- Explain the basic security requirements for IBM Process Server
- Describe the methods that are used to secure SCA applications
- Identify the role-based security elements of processes and human tasks

Security

© Copyright IBM Corporation 2017

Figure 13-46. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: Authentication aliases are used to authenticate runtime code for access to system resources such as messaging engines and connection factories.
2. True or False: WS-Security is the preferred security protocol for web services.
3. True or False: The security permission qualifier specifies an identity under which an implementation runs.

Security

© Copyright IBM Corporation 2017

Figure 13-47. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: Authentication aliases are used to authenticate runtime code for access to system resources such as messaging engines and connection factories.
   The answer is True.
2. True or True: WS-Security is the preferred security protocol for web services.
   The answer is True.
3. True or False: The security permission qualifier specifies an identity under which an implementation runs.
   The answer is False. It specifies a role that a specified type of user must possess to use an operation in an interface.

© Copyright IBM Corporation 2017

Figure 13-48. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Implementing security

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 13-49. Exercise 12

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

IBM Training

Exercise objectives

- Configure security qualifiers for an SCA application
- Configure role-based security for human tasks
- Configure authorization for business process components
- Use Business Process Choreographer Explorer to test human task and process permission settings
- Configure users and groups with Process Center internal security
- Configure participant groups
- Test human service security with Process Portal

Security

© Copyright IBM Corporation 2016

Figure 13-50. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

Components that are required for Exercise 12

Prebuilt components that are imported in this lab:

- You do not import any object in this exercise
- You work with the existing infrastructure, and the tools that are available in IBM Business Process Manager
- You work with:
  - **Administrative console**
  - **Process Portal**
  - **Process Admin console**
  - **Process Designer**
  - **Process Center**
  - **IBM Integration Designer**
  - **Business Process Choreographer Explorer**

New components that you create in this lab:

1. Create users and groups

Security

© Copyright IBM Corporation 2017

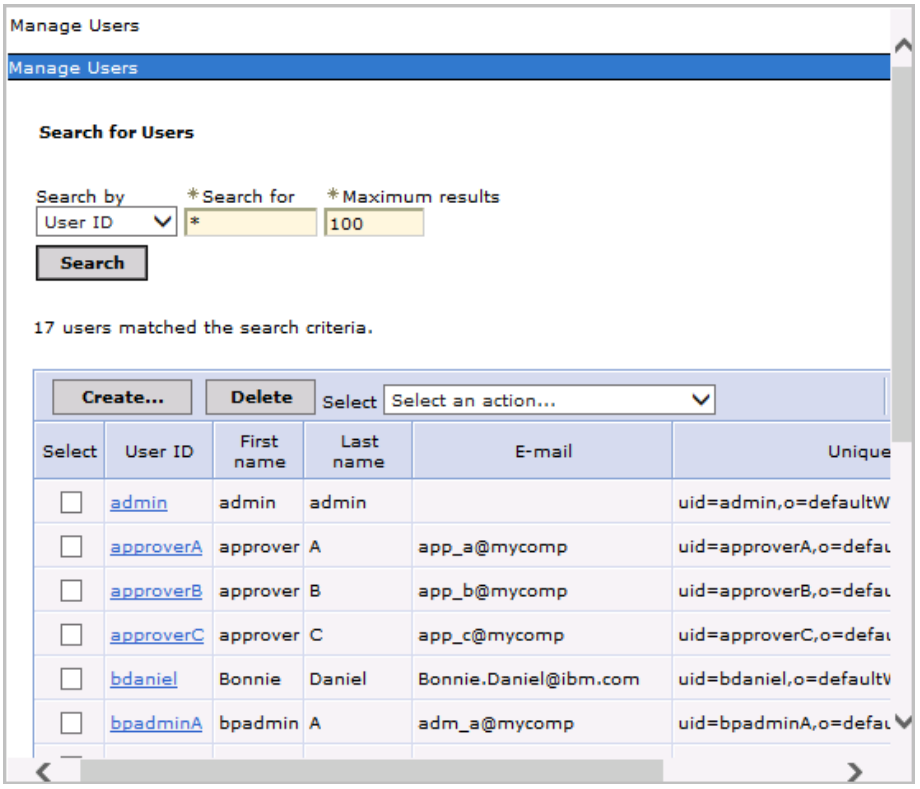
Figure 13-51. Components that are required for Exercise 12

© Copyright IBM Corp. 2017

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

IBM Training

Create users and groups in administrative console in Ex 12



© Copyright IBM Corporation 2017

Figure 13-52. Create users and groups in administrative console in Exercise 12

In this part of the exercise, you run a customized script that creates several users and groups that you use to test the components in this exercise.

© Copyright IBM Corp. 2017

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

IBM Training

Manage application security using security qualifier in Ex 12

	Security	
Component	Security permission	Security identity
AccountVerificationExport		
AccountVerification		
Implementation		
Interfaces		
AccountVerification		
InputCriterion	approvers	
References		
CreditCheckServicePartner		
FinalApplicationReviewPartner		

Security

© Copyright IBM Corporation 2017

Figure 13-53. Manage application security using security qualifier in Ex 12

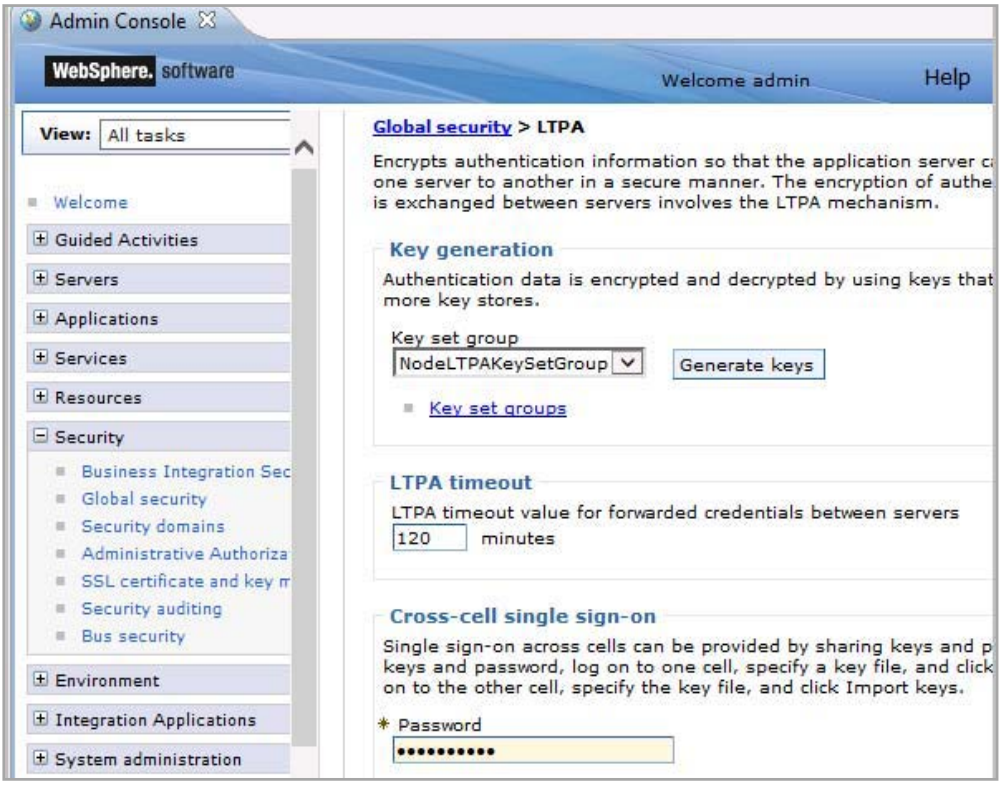
You add the group **approvers** to the **Security permission** qualifier for the **InputCriterion** operation on the **AccountVerification** interface. This addition means that only members of the approvers group have access to start a process instance by starting the InputCriterion operation on the process interface. These steps are about enabling application security.

© Copyright IBM Corp. 2017

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

IBM Training

Create LTPA encryption using global security in Ex 12 (1 of 2)



Security © Copyright IBM Corporation 2017

Figure 13-54. Create LTPA encryption using global security in Ex 12 (1 of 2)

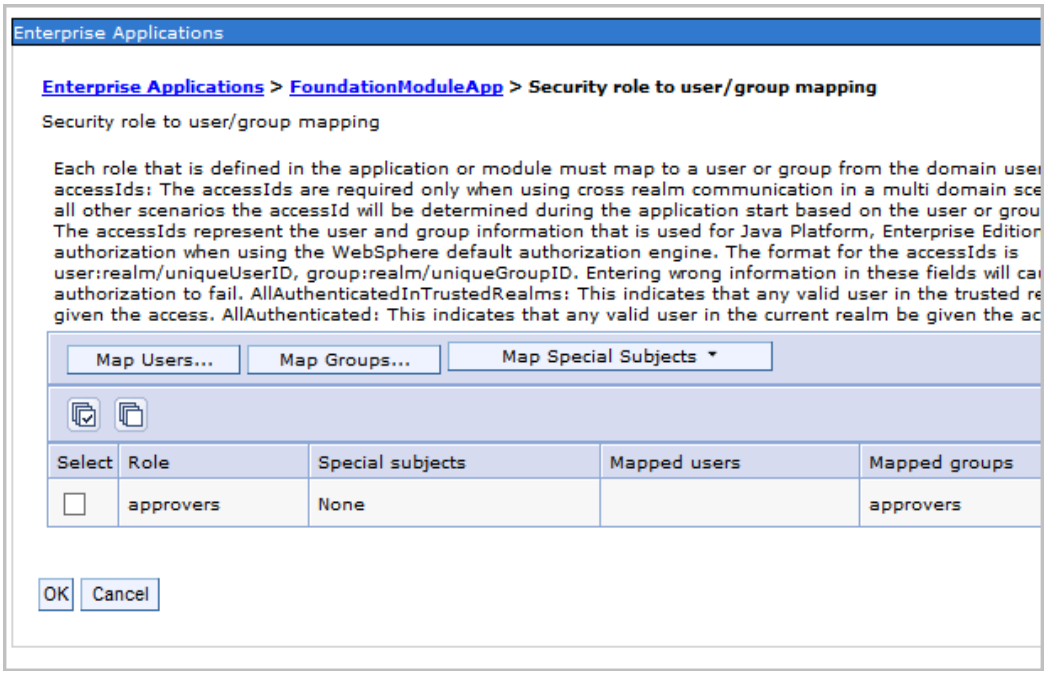
To enforce the permissions for tasks, you must enable security on IBM Process Server. The steps necessary for enabling security on IBM Process Server are identical to the steps for enabling security on WebSphere Application Server. These steps are also identical for turning on security in the WebSphere test environment or a separate installation of IBM Process Server. Here, you create the LTPA encryption key, which is used to establish the authentication token.

© Copyright IBM Corp. 2017

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

IBM Training

Create LTPA encryption using global security in Ex 12 (2 of 2)



© Copyright IBM Corporation 2017

Figure 13-55. Create LTPA encryption using global security in Ex 12 (2 of 2)

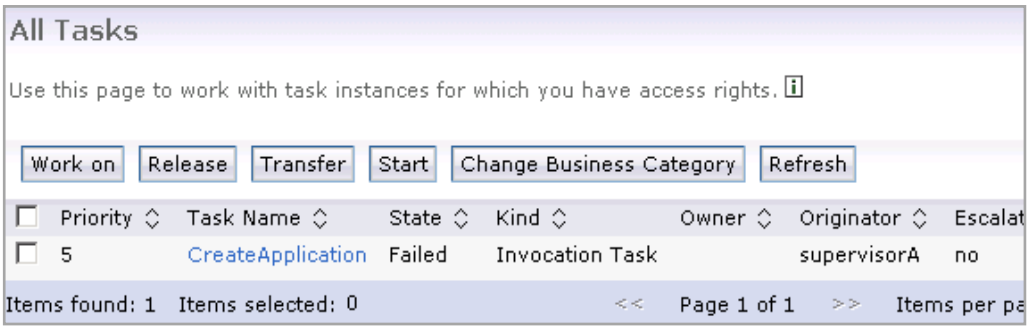
You assign the roles to the **AccountVerification** business process. Those assignments are role names, not the actual user or group names. For security to work, all of the applications that contain roles must map the roles to a user or group. You do that in this portion of the exercise.

© Copyright IBM Corp. 2017

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

IBM Training

Test task security in Business Process Choreographer Explorer in Ex 12 (1 of 2)



Security

© Copyright IBM Corporation 2017

Figure 13-56. Test task security in Business Process Choreographer Explorer in Exercise 12 (1 of 2)

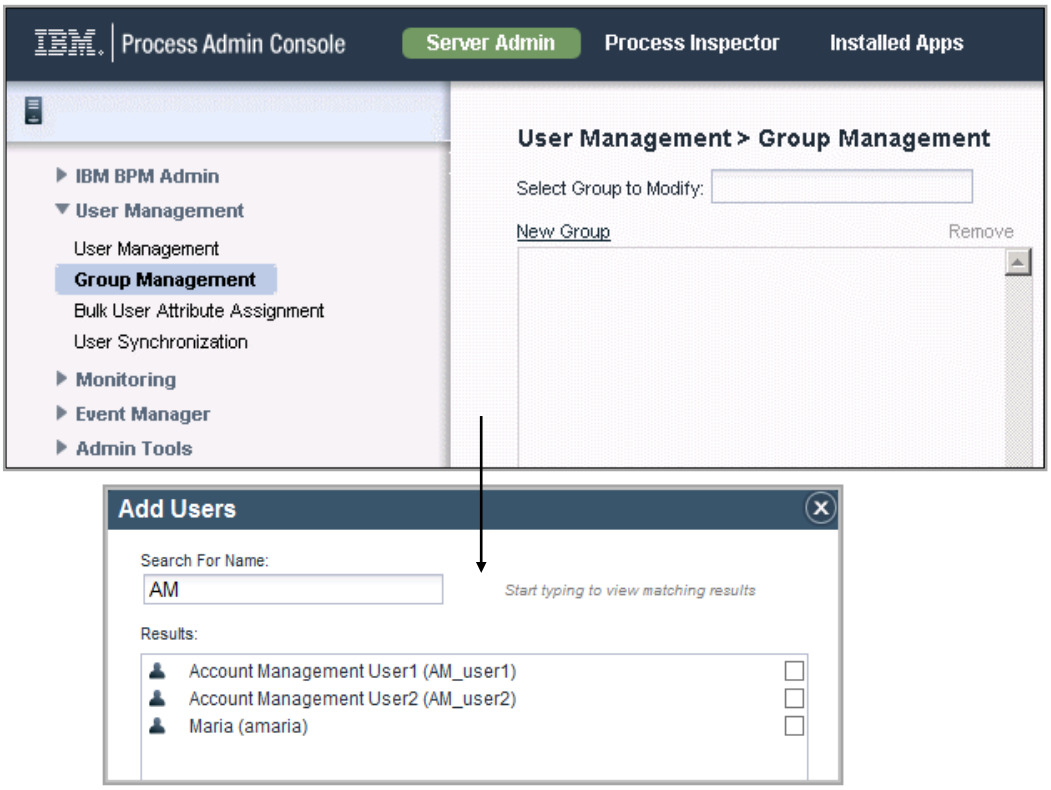
You create an instance of the **AccountVerification** process through the AccountOpeningUI human task user interface. You log in to the Business Process Choreographer Explorer client with the superVisorA user. Since this user is not from an approvers role for which security permission QoS was defined, it does not allow you to create an instance.

© Copyright IBM Corp. 2017

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

IBM Training

Test task security in Business Process Choreographer Explorer in Ex 12 (2 of 2)



Security © Copyright IBM Corporation 2017

Figure 13-57. Test task security in Business Process Choreographer Explorer in Exercise 12 (2 of 2)

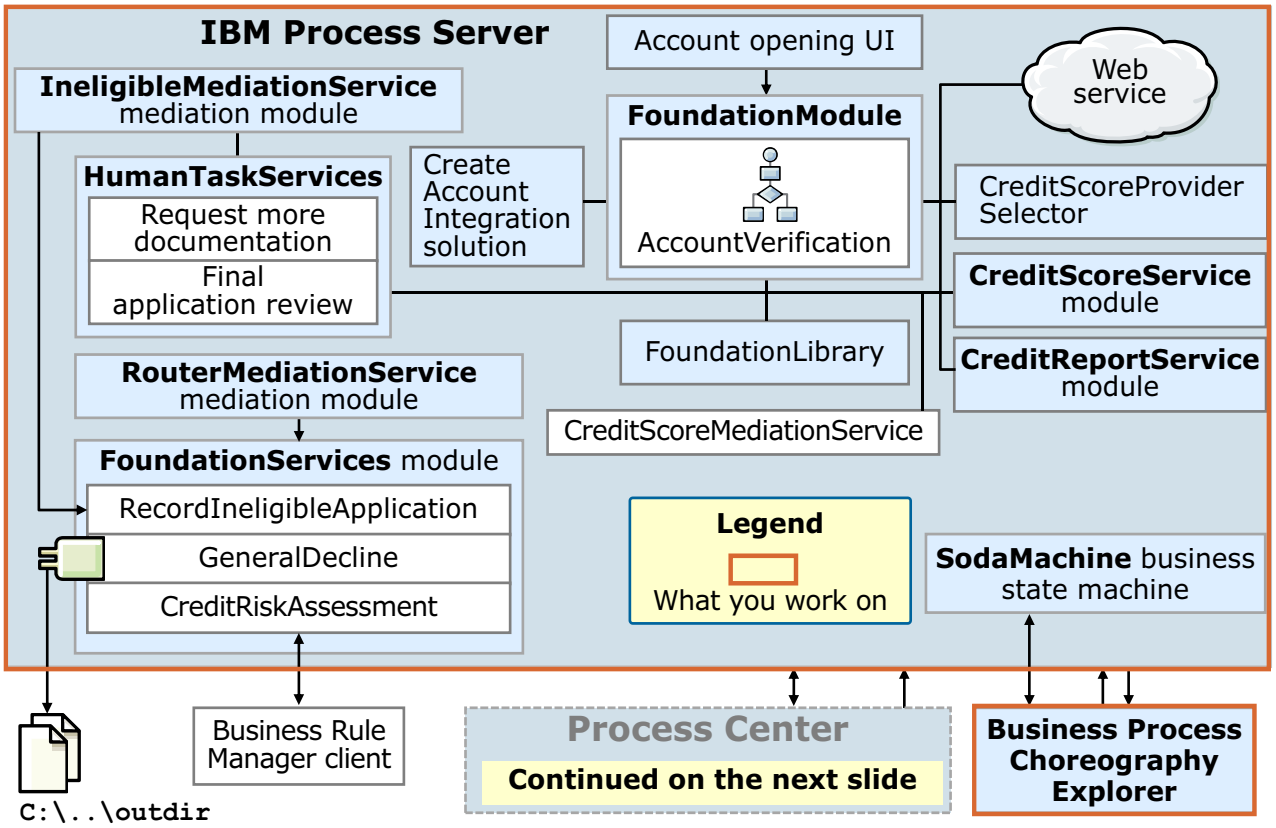
In this portion of the exercise, you experience how to create users and groups, and assign them to a participant group. You further learn how to configure a participant lane to use the created participant group. With this configuration, you enable security to all the activities in the participant lane so that only users of that participant group can participate in the activities of that lane in the business process definition.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 12 (1 of 2)



Security

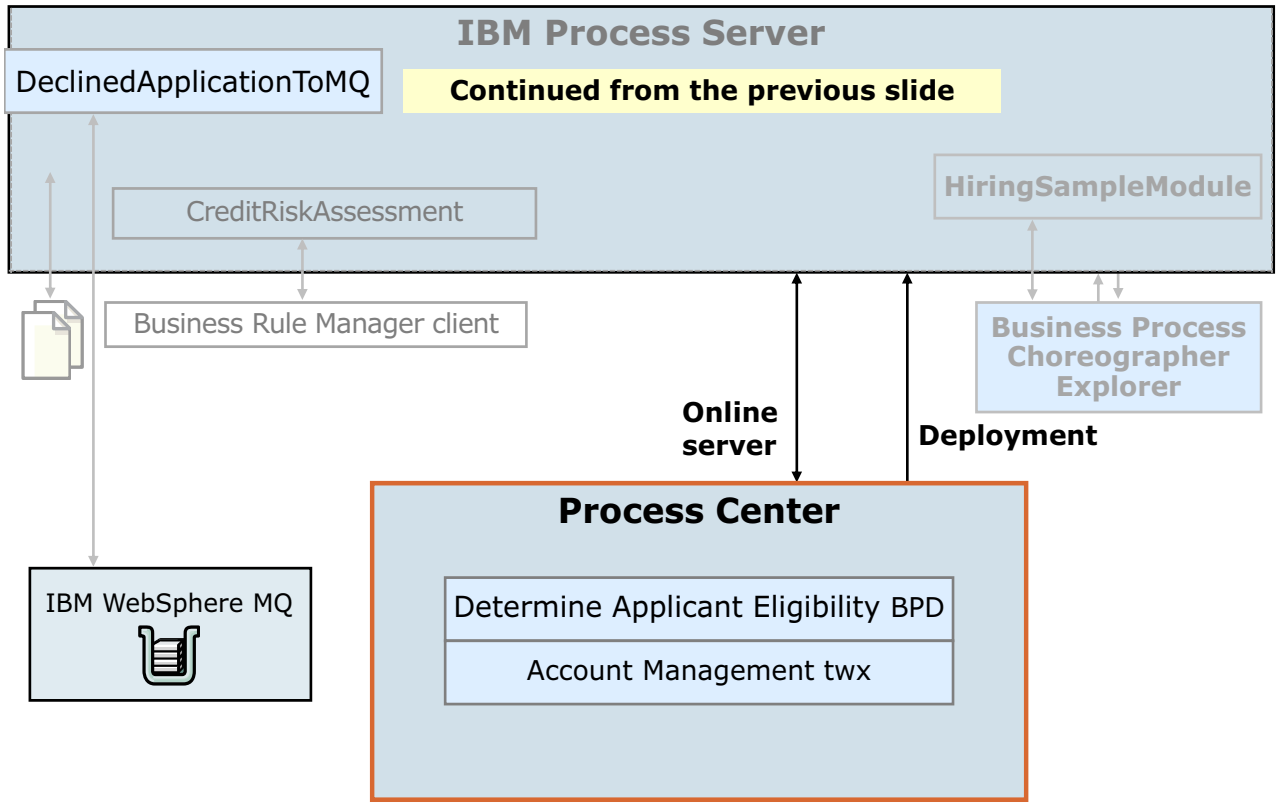
© Copyright IBM Corporation 2017

Figure 13-58. Course business scenario: What you work with in Ex 12 (1 of 2)

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

IBM Training

Course business scenario: What you work with in Ex 12 (2 of 2)



Security

© Copyright IBM Corporation 2017

Figure 13-59. Course business scenario: What you work with in Ex 12 (2 of 2)

© Copyright IBM Corp. 2017

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

Unit 14. Governance of process applications

Estimated time

00:45

Overview

This unit describes the governance of process applications.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain the purpose of applying governance to process applications
- Identify the integration services in the System Governance toolkit to create a governance process for a snapshot
- Describe how to enable governance in process applications

Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Governance of process applications: Overview

- The governance feature provides you control over the installation of snapshots
- When governance is enabled, snapshot installation requests made from Process Center pass through the governance process
- Snapshot is installed, only if the approvals defined in governance process are completed
- You use the System Governance toolkit to define custom governance processes
- Governance feature provides you a way to ensure that complete testing is done and approvals are secured before installation

Governance of process applications © Copyright IBM Corporation 2017

Figure 14-2. Governance of process applications: Overview

When governance is enabled on any process application, it controls the installation of the snapshot of that process application from IBM Process Center Server to the connected IBM Process Server.

Therefore, when the IBM Process Center requests the installation of the snapshot to the IBM Process Server, the request passes through the governance process.

When all approvals that are defined in the governance process are completed, only then does the snapshot get installed on the IBM Process Server.

The key benefit of applying such governance to the process application is to: 1) ensure that all the testing of the process application is done and 2) all the necessary approvals are in place before installation on the IBM Process Server.

The System Governance toolkit is used to build the custom governance services that can be used as a governance process.

© Copyright IBM Corp. 2017

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

IBM Training

What is a governance process?

- Governance process is a process application that defines the approvals that are required before the process application snapshot can be installed on a process server
- The System Governance toolkit provides the integration services that are required to build a governance process
- List of integration services included in toolkit:
  - Cancel Snapshot Installation
  - Get All Process Servers
  - Get Installation Status
  - Get Snapshot Status
  - Install Snapshot
  - Set Installation Status
  - Set Snapshot Status
- You cannot edit or change the artifacts in the System Governance toolkit

Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-3. What is a governance process?

The governance process is a process application where a process is defined to get approvals before the snapshot of a process application is installed to the connected IBM Process Server.

This process should be available before you enable governance to any process application. The machinery that is required to build the governance process application is provided in the System Governance toolkit as integration services. This toolkit is imported automatically when the IBM Business Process Manager product in installed. Before starting development of a governance process application, the dependency on the System Governance toolkit should be set. Only someone in the administrative group can set this dependency. The integration services that the toolkit provides are read-only, so you cannot edit them.

The integration services that the System Governance toolkit provides are:

- Cancel Snapshot Installation: Is used to cancel the request to install a snapshot.
- Get All Process Servers: Returns an array of ProcessServers registered with Process Center.
- Get Installation Status: Gets the installation status during long-running installations.
- Get Snapshot Status: Gets the snapshot.
- Install Snapshot: Is used to install the snapshot to the specified process server.

© Copyright IBM Corp. 2017

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

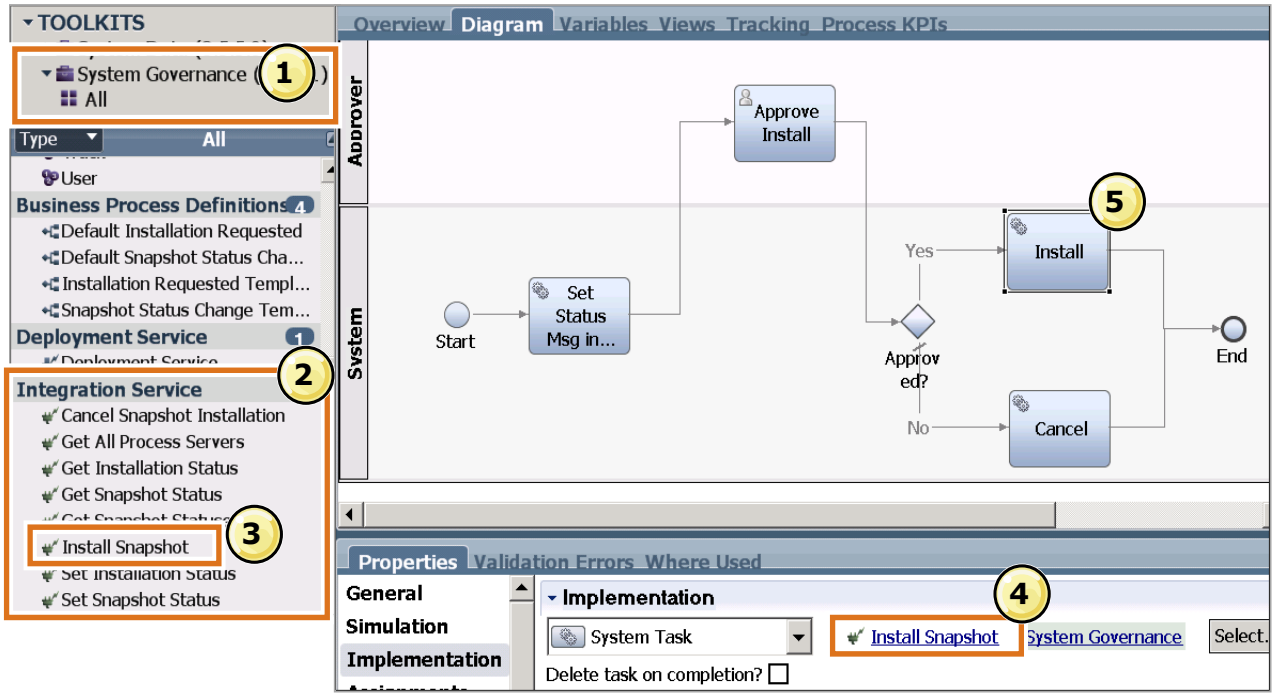
- Set Installation Status: Sets the installation status during long-running installations. Define the installation status messages that you want to use in the governance process while an installation request is waiting for approval.
- Set Snapshot Status: Sets the snapshot status.

© Copyright IBM Corp. 2017

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

IBM Training

Sample governance process application



Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-4. Sample governance process application

1. To develop a governance process application, the first step is to set the dependency on the System Governance toolkit.
2. All integration services available in the toolkit can be seen in the Implementation tab.
3. Use the Install Snapshot integration service to install the snapshot.
4. Configure the **Install** activity to use the **Install Snapshot** integration service from the System Governance toolkit as implementation.
5. The next step is to develop a process logic. This sample has a human service, Approve Install. When the authorized user approves the installation of the snapshot, it gets installed; otherwise, it does not. The **Install** service basically calls the **Install Snapshot** integration service, and the **Cancel** service calls the **Cancel Snapshot Installation** integration service from the System Governance toolkit.

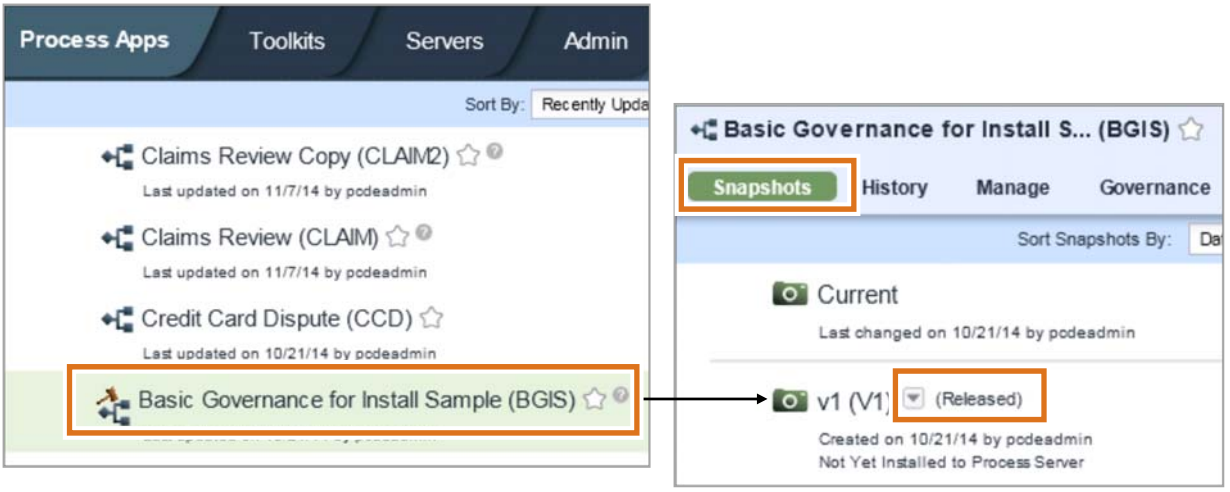
© Copyright IBM Corp. 2017

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

IBM Training

Release the governance process

- Make sure that a governance process (**Basic Governance for Install Snapshot**) is defined in the Process Center server
- Set snapshot status of the governance process to **Released** before enabling the process application to use the governance



© Copyright IBM Corporation 2017

Figure 14-5. Release the governance process

1. Make sure that a governance process (**Basic Governance for Install Sample**) is defined in the Process Center server.
2. Set the snapshot status of the governance process to **Released** before enabling the process application to use the governance process.

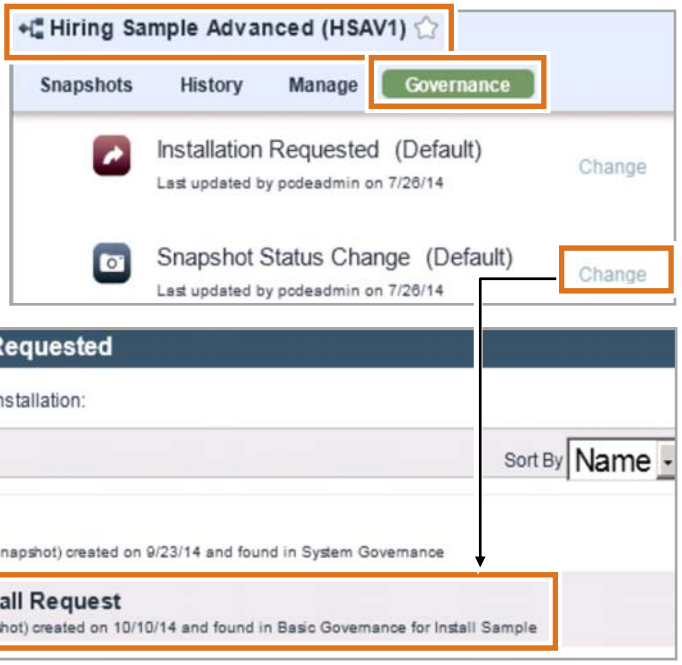
© Copyright IBM Corp. 2017

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

IBM Training

Enable governance for a process application

- Ensure that the snapshot status of the governance process is set to **Released** in the Process Center server before governance for a process application is enabled
- Administrative authority is required to enable or disable governance for a process application



- **Change** the Installation Requested governance from Default to **Review Install Request** to enable governance for the process application

Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-6. Enable governance for a process application

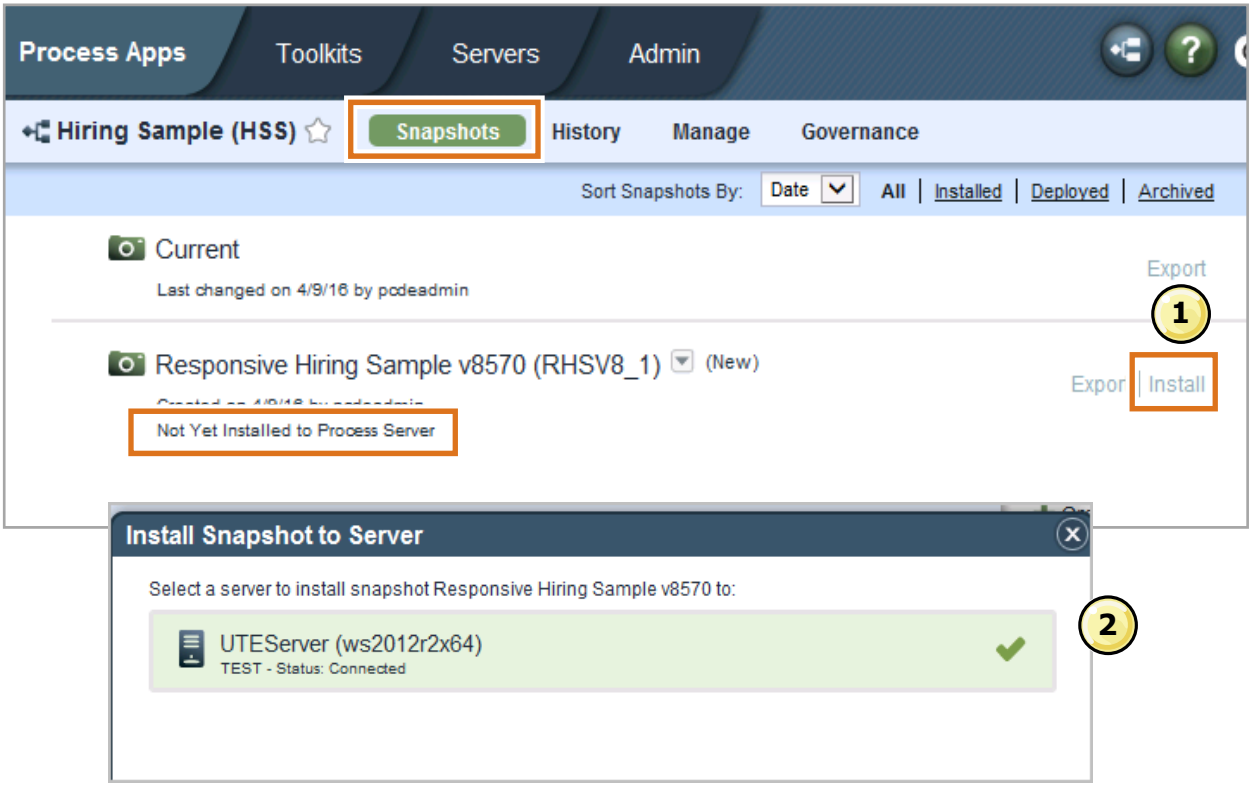
1. Make sure that snapshot status of the governance process is set to **Released** in the Process Center server before governance for a process application (**Hiring Sample Advanced**) is enabled.
2. Administrative authority is required to enable or disable governance for a process application.
3. **Change** the Installation Requested governance from Default to **Review Install Request** to enable governance for the process application.

© Copyright IBM Corp. 2017

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

IBM Training

Installing snapshots on IBM Process Server



Governance of process applications © Copyright IBM Corporation 2017

Figure 14-7. Installing snapshots on IBM Process Server

1. In the Snapshots tab of a process application, notice that the status is **Not Yet Installed to Process Server**. Click **Install**.
2. You see the window to select on which server to install the snapshot. Select the server, and the installation process begins. It first goes to the governance process to take the necessary approvals, described in next slide.

© Copyright IBM Corp. 2017

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

IBM Training

Status messages

- Status message below the snapshot changes to **"Installation in progress"**

- After few seconds the status message updates to "**Approval for Installation**"

© Copyright IBM Corporation 2017

Figure 14-8. Status messages

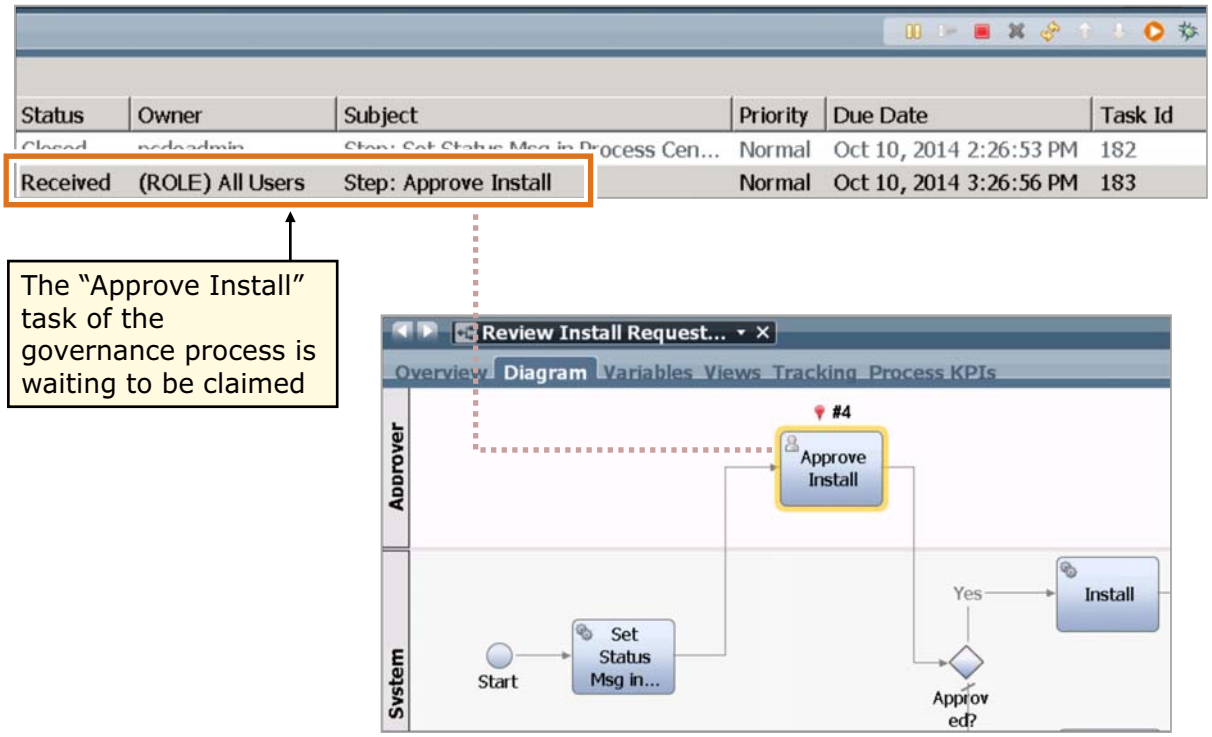
These status messages can be set by using the "**Set installation status**" integration service of System Governance toolkit.

© Copyright IBM Corp. 2017

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

IBM Training

Triggered governance process (1 of 2)



© Copyright IBM Corporation 2017

Figure 14-9. Triggered governance process (1 of 2)

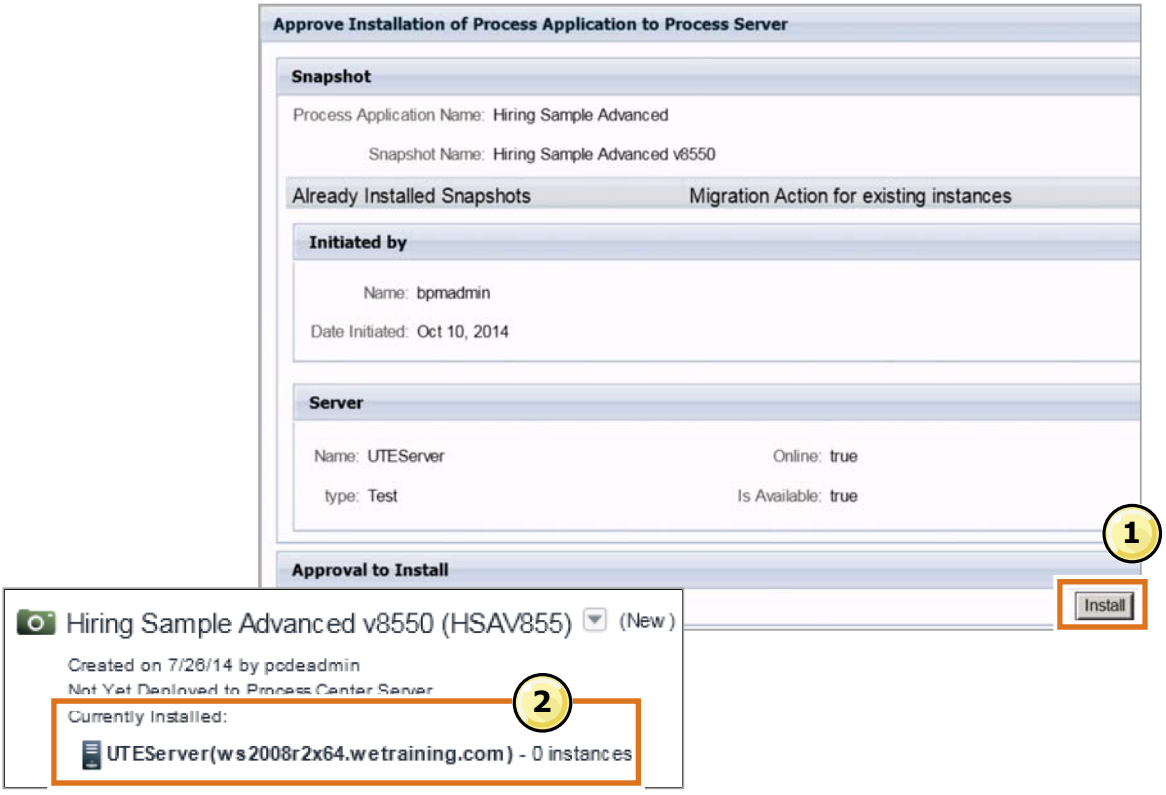
When you click **Install** in the snapshot, it triggers the governance process, and it then waits for the authorized user to claim the task of "Approve Install" shown in the console.

© Copyright IBM Corp. 2017

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

IBM Training

Triggered governance process (2 of 2)



© Copyright IBM Corporation 2017

Figure 14-10. Triggered governance process (2 of 2)

1. The authorized user claims the “Approve Install” task, views the details about the snapshot that is requested to be installed, and takes the decision to either **Install** or **Cancel** the request.
2. If the person selects to **Install,** the snapshot gets installed on the selected, IBM Process Server and the status is updated as “Currently Installed”.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain the purpose of applying governance to process applications
- Identify the integration services in the System Governance toolkit to create a governance process for a snapshot
- Describe how to enable governance in process applications

© Copyright IBM Corporation 2017

Figure 14-11. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or False: Governance process should be defined before enabling or disabling the governance for a process application.
2. True or False: Users with any rights can enable or disable the governance for a process application.
3. True or False: Integration services in the governance toolkit are editable.

Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-12. Review questions

Write your answers here:

1.

2.

3.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: Governance process should be defined before enabling or disabling the governance for a process application.
   The answer is True.
2. True or False: Users with any rights can enable or disable the governance for a process application.
   The answer is False. Administrative authority is required to enable or disable governance for a process application.
3. True or False: Integration services in the governance toolkit are editable.
   The answer is False. Integration services in the governance toolkit are not editable.

Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-13. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Applying governance to process applications

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 14-14. Exercise 13

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

IBM Training

Exercise objectives

- Create a governance process application
- Enable governance in a process application
- Test the governance of a snapshot installation

© Copyright IBM Corporation 2016

Figure 14-15. Exercise objectives

© Copyright IBM Corp. 2017

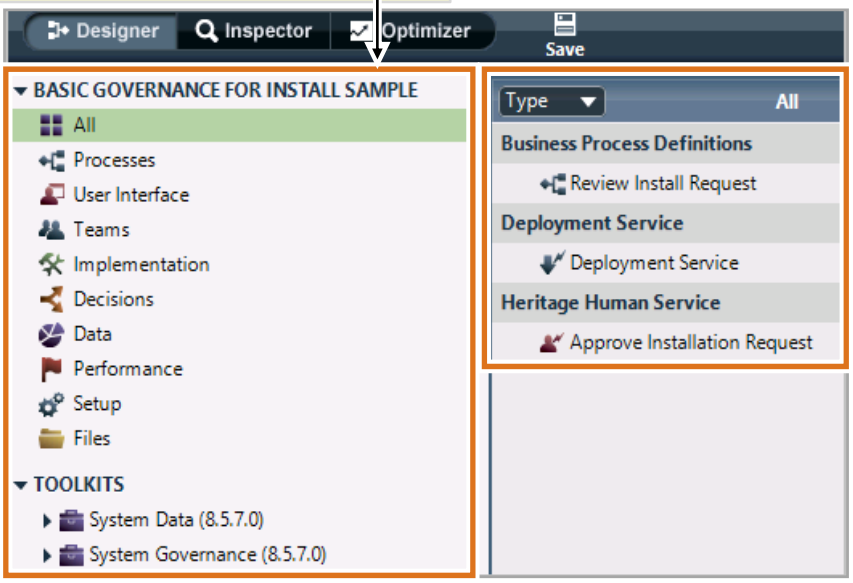
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Components that are required for Exercise 13

Prebuilt components that are imported in the lab:

1. **Basic Governance for Install Sample (BGIS)**
2. **System Governance toolkit (TWSYSG)**
3. **Basic_Governance_for_Install_Sample - v1**



New components that you create in the lab:

None

© Copyright IBM Corporation 2017

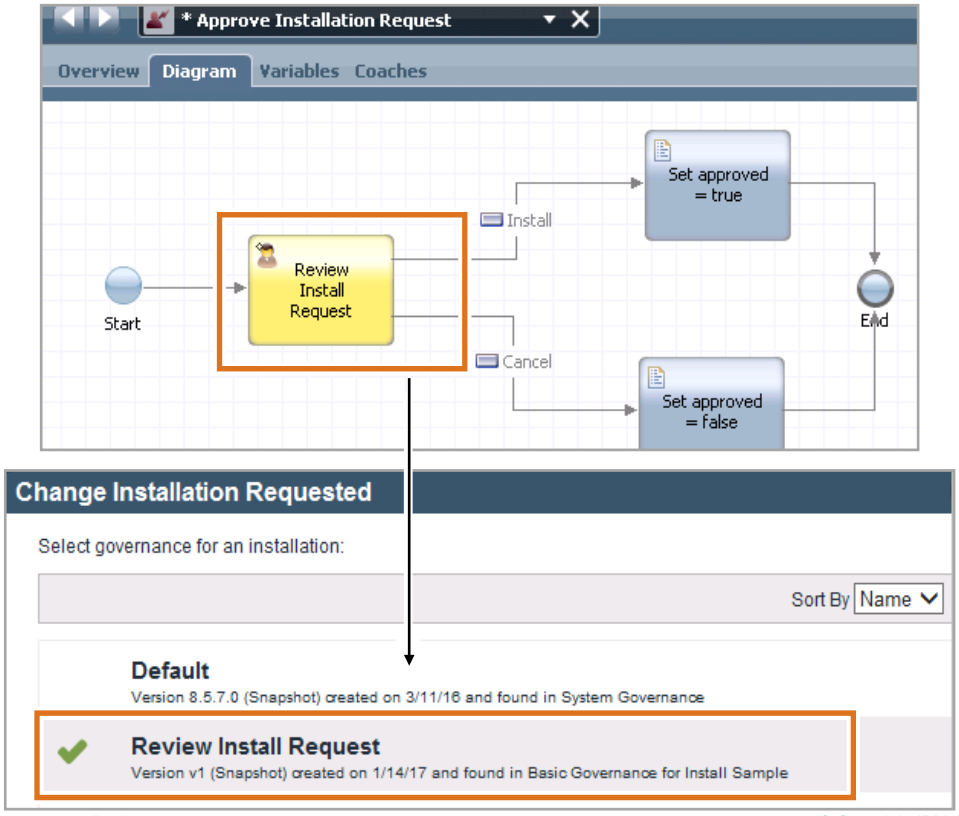
Figure 14-16. Components that are required for Exercise 13

© Copyright IBM Corp. 2017

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

IBM Training

Test governance template in Process Center in Exercise 13



© Copyright IBM Corporation 2017

Figure 14-17. Test governance template in Process Center in Exercise 13

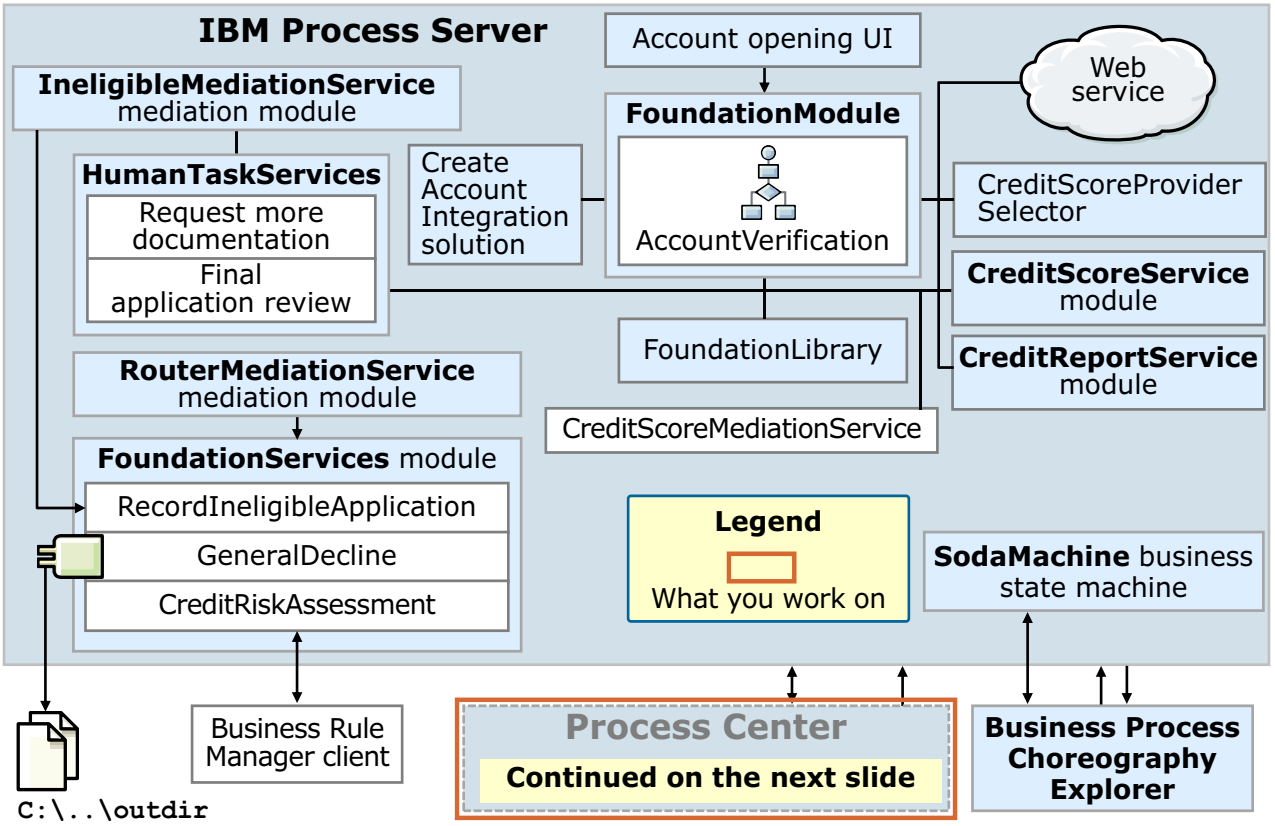
The BPD describes different activities that are involved in the installation approval process. You go through the implementation of each activity in the rest of the lab material.

© Copyright IBM Corp. 2017

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

IBM Training

Course business scenario: What you work with in Ex 13 (1 of 2)



Governance of process applications

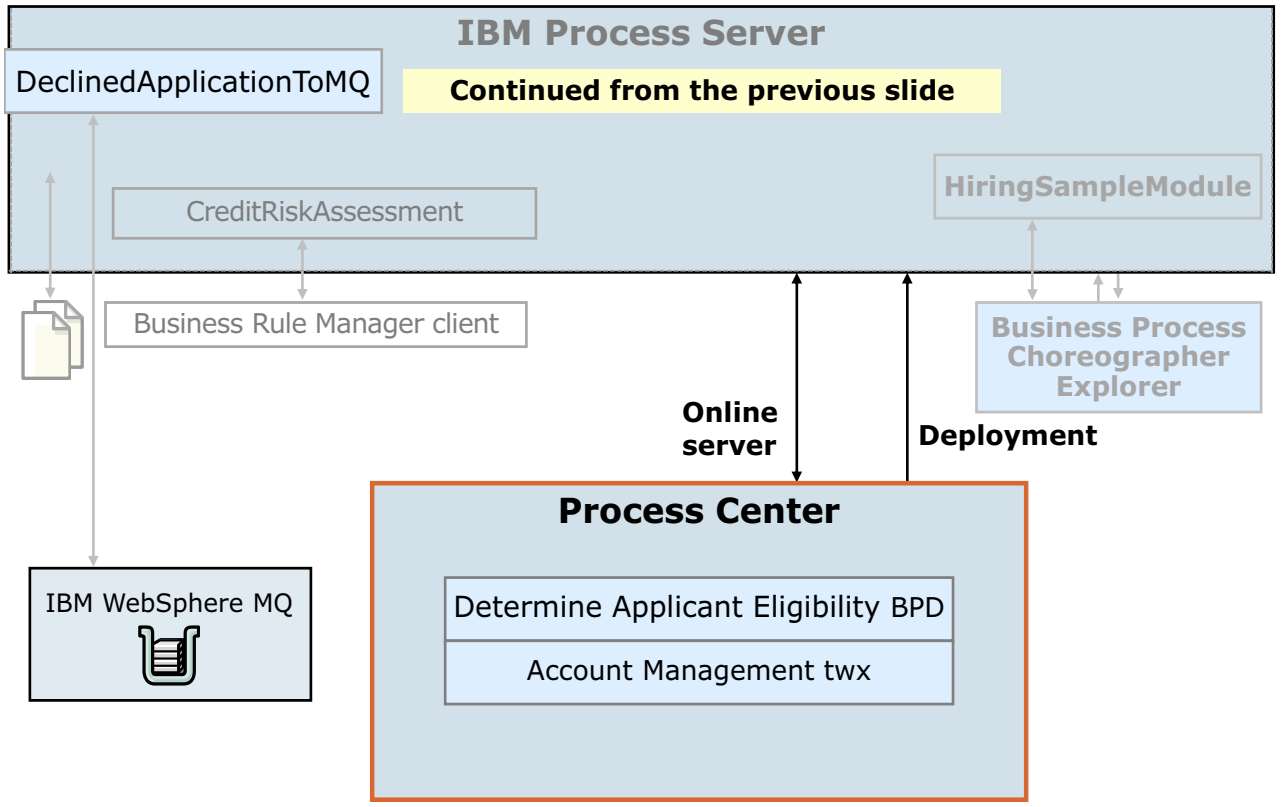
© Copyright IBM Corporation 2017

Figure 14-18. Course business scenario: What you work with in Ex 13 (1 of 2)

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

IBM Training

Course business scenario: What you work with in Ex 13 (2 of 2)



Governance of process applications

© Copyright IBM Corporation 2017

Figure 14-19. Course business scenario: What you work with in Ex 13 (2 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

References

- More information about the topic can be found at IBM Process Designer:
  - http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/govern_install_intro.html
    (for Version 8.5.5)

© Copyright IBM Corporation 2017

Figure 14-20. References

© Copyright IBM Corp. 2017

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

Unit 15. Integrating with IBM Business Process Manager

Estimated time

01:00

Overview

This unit describes the integration of IBM Business Process Manager with other products, such as IBM Business Monitor, IBM Case Manager, and WebSphere Service Registry and Repository.

How you will check your progress

- Checkpoint
- Machine exercise

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Describe integration strategies for IBM Process Designer and IBM Enterprise Content Management
- Describe integration strategies for IBM Integration Designer and IBM MobileFirst
- Describe integration strategies for IBM Process Server and IBM Operational Decision Manager
- Describe integration strategies for IBM Integration Designer and IBM Business Monitor

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

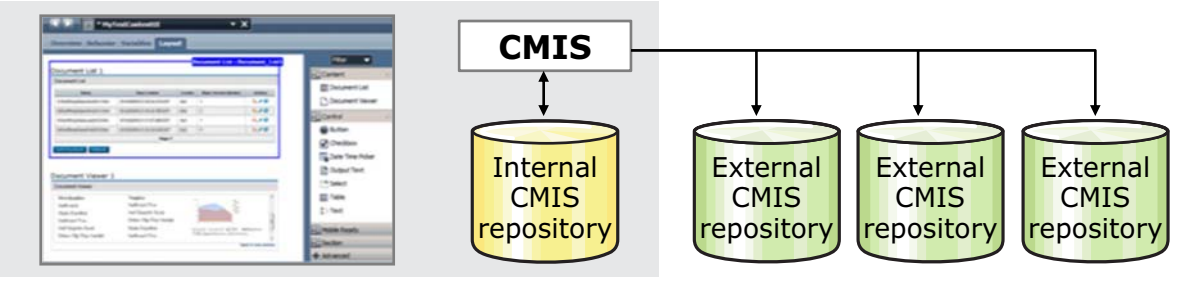
Figure 15-1. Unit objectives

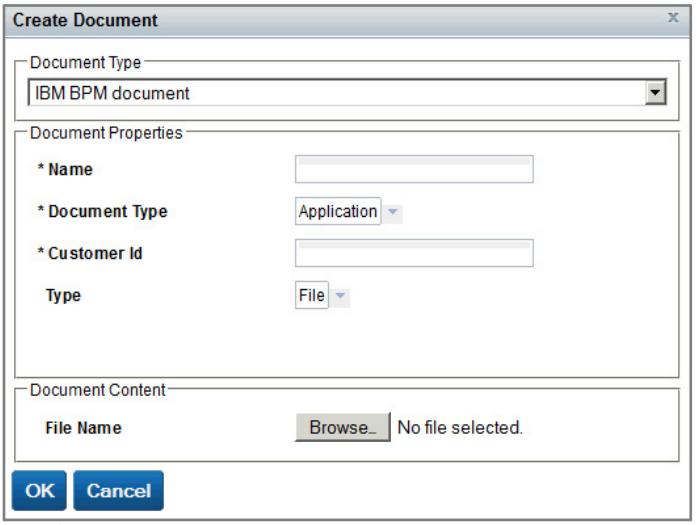
© Copyright IBM Corp. 2017

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

IBM Training

Integrating IBM BPM and Enterprise Content Management





Integrating with IBM Business Process Manager © Copyright IBM Corporation 2017

Figure 15-2. Integrating IBM BPM and Enterprise Content Management

Enterprise Content Manager systems manage documents of different types, such as records, images, and web pages, throughout their lifecycle. You can access and update such documents from a business process inside IBM Business Process Manager by using special Enterprise Content Manager operations in services such as an integration service or a client-side human service.

The integration with Enterprise Content Manager systems is based on Content Management Interoperability Services (CMIS). CMIS is an open standard that the Organization for the Advancement of Structured Information Standards (OASIS) administers. CMIS defines an abstraction layer for controlling diverse document management systems and repositories by using web protocols. As a result, you can have seamless integration between IBM Business Process Manager and Documentum, IBM Content Manager, FileNet Content Manager, SharePoint, or any other JCR-compliant repository.

An internal CMIS repository, which is known as the IBM BPM document store, supports an embedded Enterprise Content Manager and uses consistent access mechanisms for integrating to it. As a result, documents, images, or any other files can be associated with process instances even if your organization does not currently use an Enterprise Content Manager system.

© Copyright IBM Corp. 2017

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

Enterprise Content Manager events can trigger content start and intermediate events. For Enterprise Content Management integration, predefined types, services, and events are available. These types and services are contained in the Content Management (SYSCM) toolkit.

For more information on IBM Business Process Manager and Enterprise Content Management: http://www.ibm.com/developerworks/bpm/bpmjournal/1312_mcelroy/1312_mcelroy.html

© Copyright IBM Corp. 2017

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

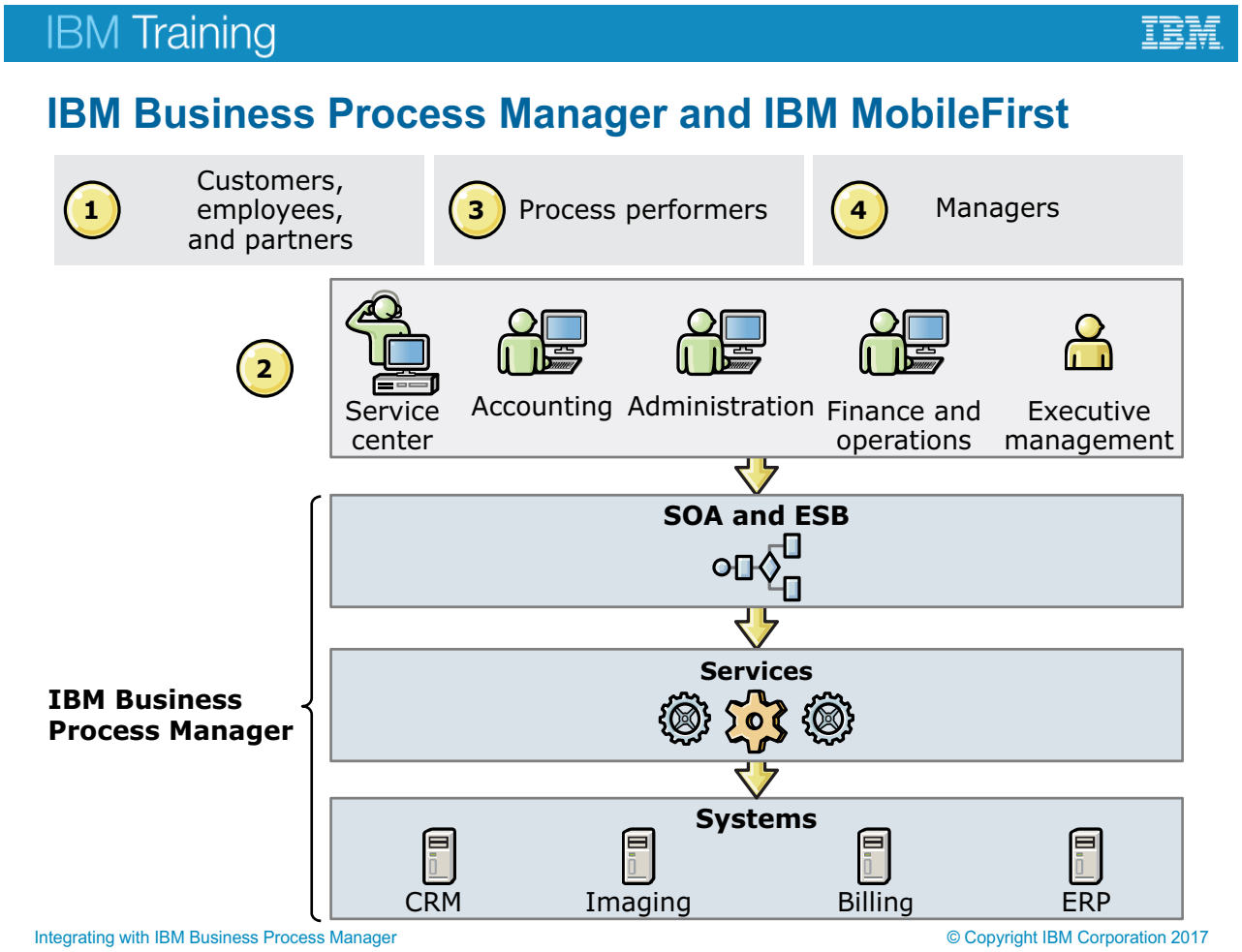


Figure 15-3. IBM Business Process Manager and IBM MobileFirst

Mobile Smarter Process helps organizations to reinvent how business is done by using Mobile technologies. As stated in the first part of the course, MobileFirst integration provides new context, new ways of engagement, and new methods of gathering input for processes. Now process participants can use mobile capabilities to interact with the process wherever they are. All of these new capabilities create more interactions than ever before. Although these interactions add value, they must be harnessed to accomplish an end goal. IBM Mobile Smarter Process combines market-leading offerings from IBM Smarter Process suites and IBM MobileFirst Platform offerings.

The illustrated process provides these advantages:

1. Customers, employees, and business partners want to interact with the organization by using their device of choice to instantiate a process or check on the status of a process they previously started. For example, in an insurance company scenario, by instantiating a claim, customers start a process that can use their device for various purposes: geolocation information available on their devices, device services such as the camera to take a picture of a damaged car and upload it, and the information that is stored in multiple back-end systems.
2. A process can involve many different roles to complete it. Smarter Process uses coaches or forms that streamline the activities of the process performers and work across any mobile devices with different form factors. A Mobile Smarter Process allows for process performers to use social collaboration to deliver a better customer experience. Integration with multiple

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

back-end systems might be necessary to complete the process that uses service-oriented architecture (SOA).

3. If the process requires human input, it can send a notification to the appropriate people and they can always be reached through their mobile devices. The business process is not going to be hindered because of someone not having access to a website or not being logged in.
4. Processes do not always occur as planned. As the global economy, the global workforce, and mobility expand, it is even more important to give managers process visibility and agility to make quick decisions. It is good to unleash the benefits of IBM Mobile Smarter Process with IBM Business Process Manager and IBM MobileFirst Platform changes. Otherwise, organizations risk losing customers. A Mobile Smarter Process gives managers the visibility and agility they need to use their device of choice for making fast adjustments to improve performance.

In summary, this simple example shows these advantages:

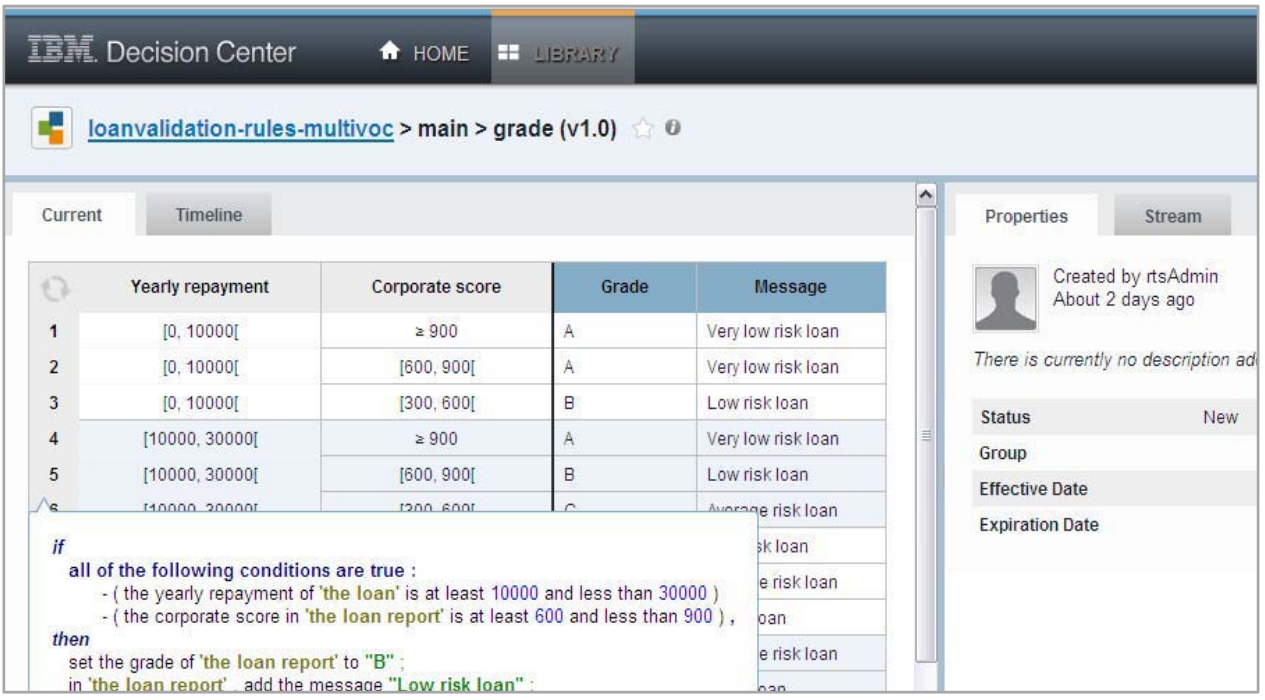
- Mobile extends the reach of a business process beyond the traditional channels to where the knowledge workers (customers, employees, business partners, managers) are. Knowledge workers always have access to the business process, and the business process can reach them when required. For example, if the process requires an approval, a notification is sent to the approver’s mobile device to alert the approver that the process is waiting for the approval.
- By adding mobility to the smarter process, Mobile Smarter Process allows managers to gain instant visibility. In this way, they can adapt to changes in the market by making faster adjustments and quicker decisions on a device of their choice. The Mobile Smarter Process has constant access to decision makers, and decision makers can always interact with the business process.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager and IBM Operational Decision Manager (1 of 2)



 © Copyright IBM Corporation 2017

Figure 15-4. IBM Business Process Manager and IBM Operational Decision Manager (1 of 2)

IBM Operational Decision Manager (ODM) complements IBM Business Process Manager products to empower businesses to automate, manage, and improve the decision cycle in the face of growing complexity and change. IBM Process Designer supports rich rule authoring and execution features that are based on paradigms that are familiar to Operational Decision Manager users. Authoring features include the Business Action Language (BAL) for authoring if-then rules in the Intellirule Editor.

Although IBM Business Process Manager provides limited rule creation capabilities, IBM Operational Decision Manager might be used for other systems outside of IBM Business Process Manager, and can provide a dedicated rules engine to the organization.

IBM Process Designer users can export a rule project that contains the rules that implement a process task. The exported rules thus move from a process-centric lifecycle and governance model to the Operational Decision Manager methodology and lifecycle. IBM Process Designer and Integration Designer also include wizards to easily integrate existing business rule applications into business processes.

IBM Operational Decision Manager (ODM) complements the IBM Business Process Manager products to empower businesses to automate, manage, and improve the decision cycle in the face of growing complexity and change.

© Copyright IBM Corp. 2017

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

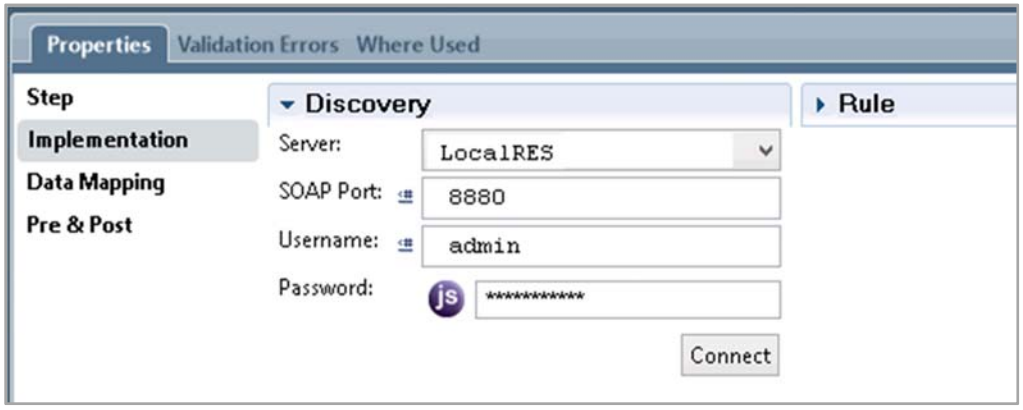
For more information about IBM Business Process Manager and IBM Operational Decision Manager, see “Best practices for designing and implementing decision services, Part 2: Integrating IBM Business Process Manager and IBM Operational Decision Management,” at: http://www.ibm.com/developerworks/bpm/bpmjournal/1212_boyer2/1212_boyer2.html

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager and IBM Operational Decision Manager (2 of 2)



© Copyright IBM Corporation 2017

Figure 15-5. IBM Business Process Manager and IBM Operational Decision Manager (2 of 2)

© Copyright IBM Corp. 2017

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

IBM Training

Building a decision service (1 of 2)

- Business rules are included in a process by adding a decision service to the process
- Add a decision service to a process application when the actions that should take place in your process depend upon one or more conditions
- Build your rule hierarchy so that rule conditions are ordered from most complex to least complex
- Create a final condition that is a catch-all rule
  - This rule is necessary if you cannot verify that the variable you want to modify in the rule is going to be set before running the process that triggers the decision service
- Consider encapsulating your rules in a single-function decision service
  - Encapsulating your rules makes the service available to any other part of the process application that needs the same rule logic

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 15-6. Building a decision service (1 of 2)

Business rules are included in a process by adding a decision service to the business process definition. Add a decision service to a process application when the actions that take place in the process depend upon one or more conditions. For example, if an employee holds the position of director and submits a meal expense for more than $250, then a rule is created and a variable is set in the rule. A rule such as approvalRequired routes the process sequence flow into a specific approval activity.

When building a decision service, follow these guidelines:

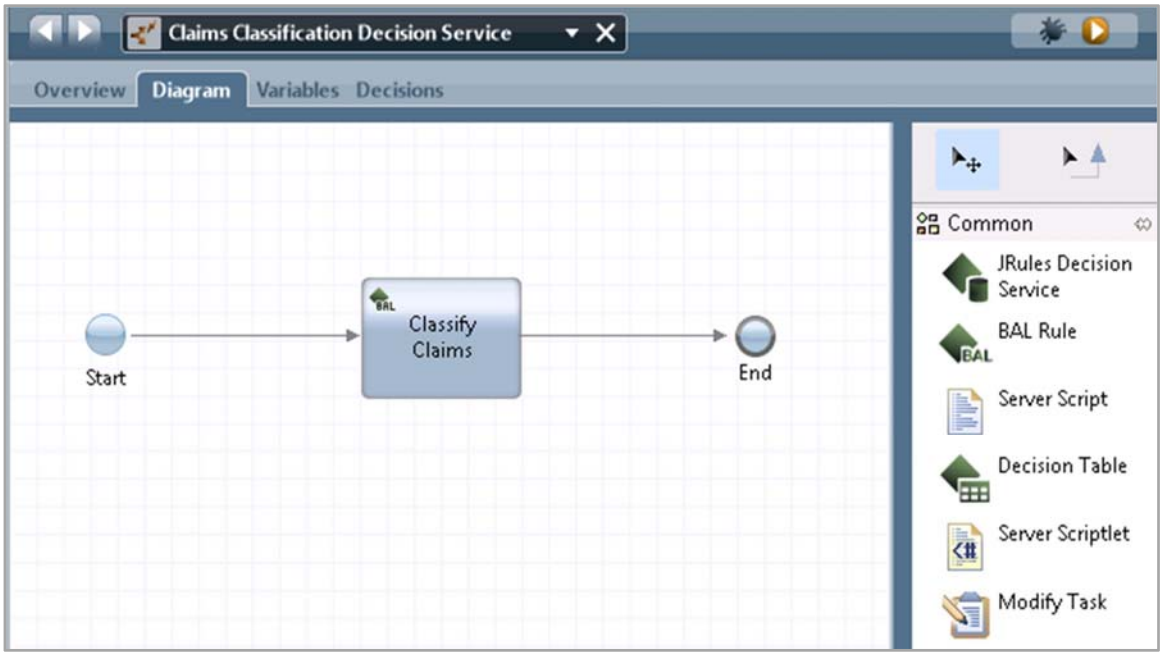
- Build the rule hierarchy so that the rule conditions are ordered from most complex to least complex.
- Create a final condition that is a catch-all rule. This rule is necessary if the team cannot verify that the variable that they want to modify in the rule is set before running the process that triggers the decision service.
- Consider encapsulating the rules in a single-function decision service that allows the service to be available to any other part of the process application that needs the same rule logic.

© Copyright IBM Corp. 2017

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

IBM Training

Building a decision service (2 of 2)



© Copyright IBM Corporation 2017

Figure 15-7. Building a decision service (2 of 2)

In the library pane, click the (**+**) plus sign next to the Decisions category in the library to create a decision service. A decision service contains one or more components. Components have three types:

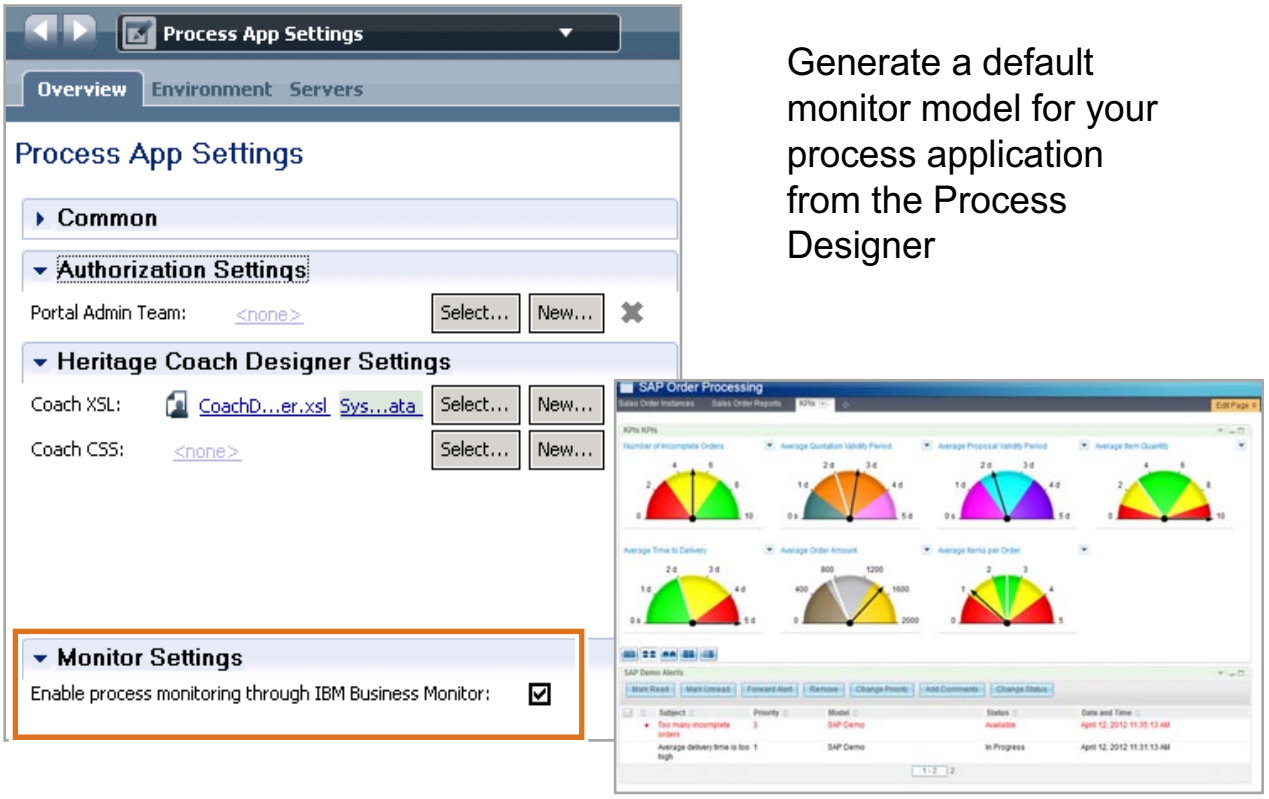
- **BAL Rule**: Use the rule editor in this component to author business rules with Business Action Language (BAL), a natural language technology.
- **JRules Decision Service**: IBM Business Process Manager integrates the IBM Operational Decision Manager with the JRules Decision Service component. You can use this rule component to connect to and implement rule applications that are available on a JRules Rule Execution Server.
- **Decision Table**: The Decision Table component contains a rule table. Each row in the rule table represents a Boolean condition that evaluates to true or false at run time. When a rule evaluates to true, the JavaScript expression that you provide as the rule action is run.

© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager and IBM Business Monitor



Generate a default monitor model for your process application from the Process Designer

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 15-8. IBM Business Process Manager and IBM Business Monitor

IBM Business Monitor is comprehensive business activity monitoring software that:

- Provides an up-to-date view of your business performance
- Provides predictions for you to act on before problems occur
- Processes business events and data with personalized business dashboards
- Calculates key performance indicators (KPIs) and metrics
- Can collect events and data from a wide variety of sources

IBM Business Process Manager contains native capabilities to visualize process performance, but Business Monitor provides a robust tool that can monitor IBM Business Process Manager and other sources.

You can use the IBM Process Designer to generate a default monitor model for your process application. A dashboard is automatically generated for the monitor model, and it shows up in the Business Space Manager page in a web browser. You can also create your own dashboard for viewing monitored data.

You can integrate IBM Business Monitor with IBM Business Process Manager to provide business monitoring capability for your process applications. Business authors who use IBM Process Designer to define a BPMN process application can specify IBM Business Monitor tracking

© Copyright IBM Corp. 2017

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

definitions for the process application without having to use a separate tool. If IBM Business Monitor is installed, its tracking definitions (part of a monitor model) are updated when you update tracking definitions inside the Process Designer or when process applications are deployed to the runtime environments.

For more information about IBM Business Process Manager and IBM Business Monitor: http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbpm.main.doc/kc-homepage-bpm.html

© Copyright IBM Corp. 2017

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

IBM Training

About IBM Business Monitor

- Captures specified business-related data from business applications that are based on the monitor model
- Extracts the measurement variables from the data
- Transforms the variables into metric and key performance indicator (KPI) values
- Shows the measurement values on the monitor dashboard
- Provides business intelligence insight through dimensional analysis and reporting
- Allows users to define actions when specified situations occur
- Identifies and notifies users of operation failures for inspection and analysis

Integrating with IBM Business Process Manager © Copyright IBM Corporation 2017

Figure 15-9. About IBM Business Monitor

Instance diagrams show content in annotated Scalable Vector Graphics (SVG). You can define elements to set colors, show and hide, associate metric and KPI values, and more.

A KPI is a key performance indicator; that is, any key business measure that a customer must measure to monitor the health of a business.

Metrics and KPIs are business performance measurements, and are covered in IBM Business Monitor courses.

© Copyright IBM Corp. 2017

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

IBM Training

Advantages of IBM Business Monitor

- Manage and optimize deployed SOA solutions
  - Continuous process improvement
  - View real-time performance
  - Intervene in deployed processes
  - Modify dashboards without coding
- Business activity monitoring (BAM)
  - The presentation of real-time information that is related to activities inside organizations and across relationships with external partners and customers
- Monitor process performance
  - View key performance indicators (KPIs) calculated from live process data
  - Detect situations, view alerts from active processes
  - Send notifications or start automated responses

Integrating with IBM Business Process Manager © Copyright IBM Corporation 2017

Figure 15-10. Advantages of IBM Business Monitor

IBM Business Monitor

BAM information is role-based, targeted, and focused to the status and results of various operations, processes, and transactions. It is used to make informed, quick business decisions, and to act to address problem areas. BAM systems collect and process business events that are fed from applications, integration software, or from business process management software.

A KPI is any key business measure that a customer must measure to monitor the health of a business.

IBM Business Monitor analyzes many event messages and captures a tremendous amount of information. However, one of its primary objectives is to distill this mass of data into relevant, consumable, business-level information pertinent to the role of the user. In the business measures model, key performance indicators (KPIs) were defined for the process. Now, as the process is running, these key performance measurements are updated in real time and are shown in various forms for the user.

This chart shows available views that show KPIs:

- A KPI table view
- A gauge view that shows KPIs as automotive gauges with defined targets and limits

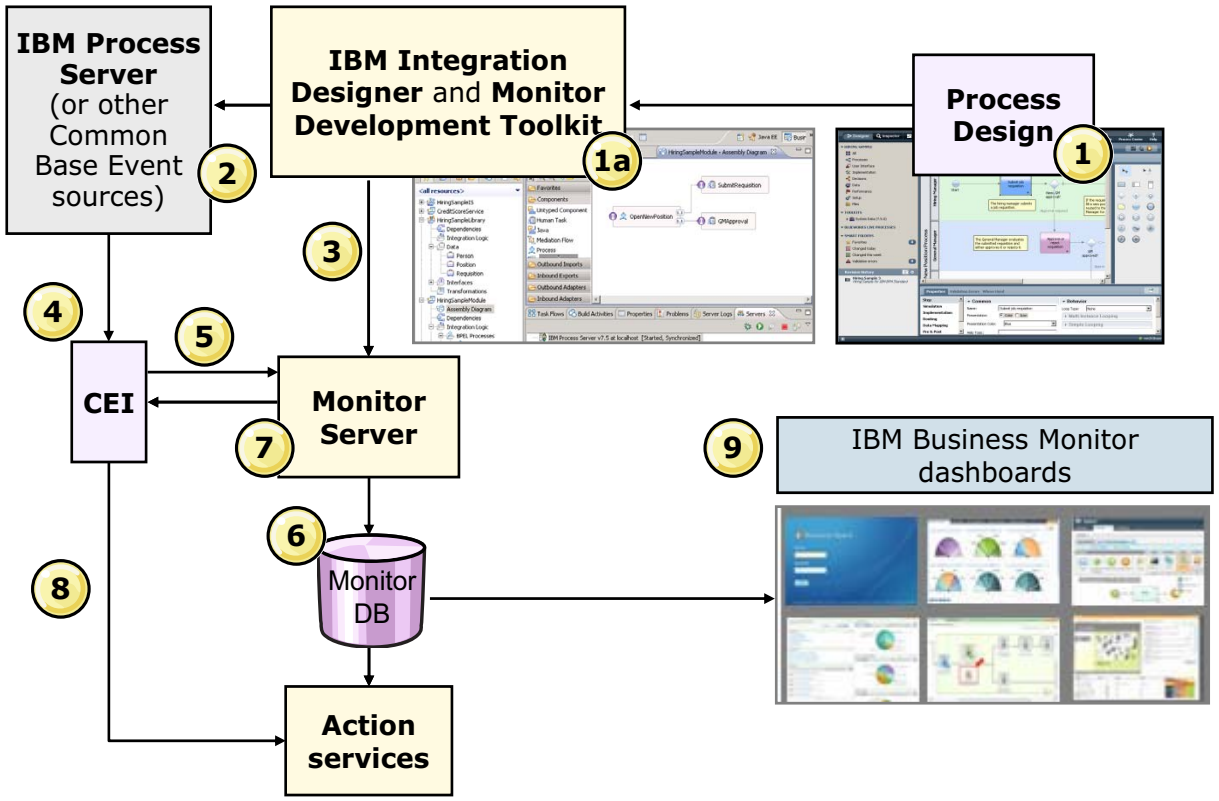
Combinations of these and other views can be used together, depending on the role of the user.

© Copyright IBM Corp. 2017

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

IBM Training

Logical architecture for business activity monitoring



© Copyright IBM Corporation 2017

Figure 15-11. Logical architecture for business activity monitoring

A process design tool can be used to create the process model, which is then exported to IBM Integration Designer.

1. Process Design: Use IBM Process Designer to create the business process model.
   a. IBM Integration Designer is used to import the process model files that were created in Process Design stage. The BPEL implementation is assembled for deployment on IBM Process Server and exported as a BPEL-compliant process model for installation on IBM Process Server. The IBM Business Monitor development toolkit is used to create the Business Monitor model for IBM Business Monitor execution.
2. The BPEL application is deployed to IBM Process Server.
3. The monitor model is deployed to the IBM Business Monitor Server.
4. Events are generated from IBM Process Server and sent to the Common Event Infrastructure (CEI).
5. CEI sends the events to Monitor Server, where they are analyzed against the monitor model to determine whether the events must be monitored.
6. The server persists metric data to the state tables in the monitor database.
7. The server emits situation events to CEI, as needed.

© Copyright IBM Corp. 2017

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

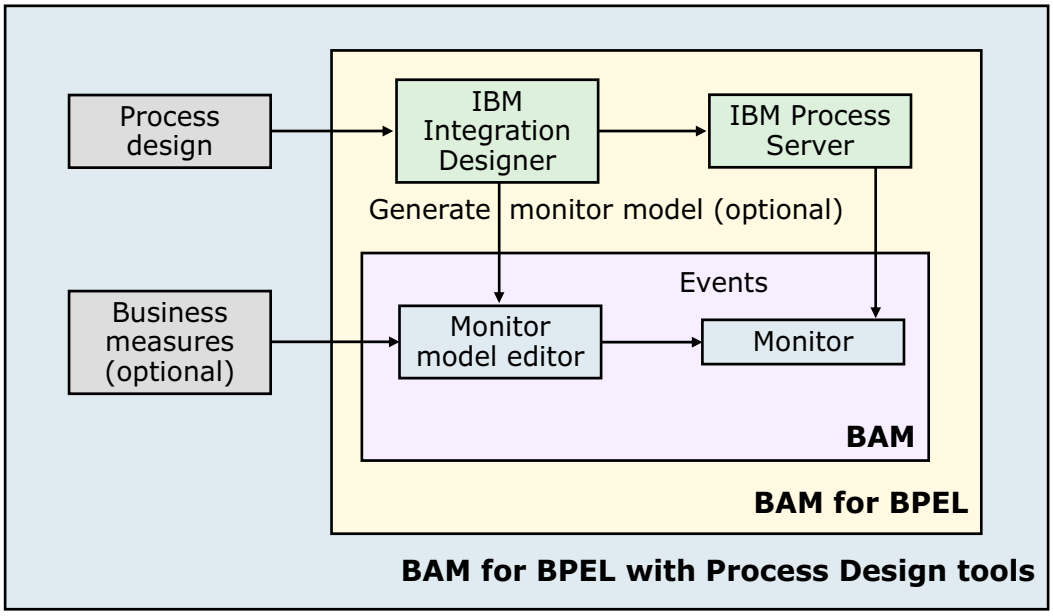
8. CEI sends situation events to the action manager, which runs the defined action.
9. Dashboards query the data warehouse for reporting and, when needed, export actual data from completed process instances.

© Copyright IBM Corp. 2017

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

IBM Training

Creating monitor models in IBM Integration Designer



© Copyright IBM Corporation 2017

Figure 15-12. Creating monitor models in IBM Integration Designer

When creating monitor models in IBM Integration Designer, keep in mind:

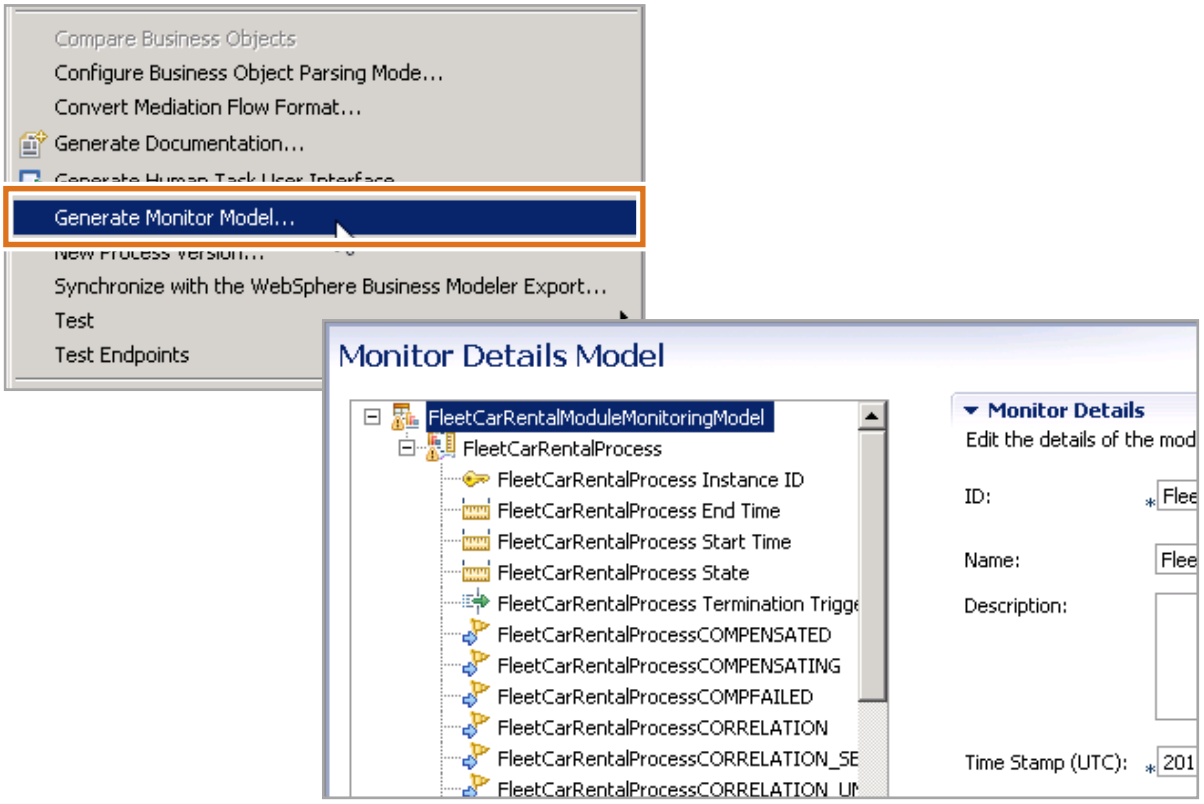
- Process design: Use IBM Process Designer to create the business process model, and optionally, to create business measures
- BAM: Business activity monitoring
- BPEL: Business Process Execution Language

© Copyright IBM Corp. 2017

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

IBM Training

Generating a monitor model in IBM Integration Designer



Integrating with IBM Business Process Manager © Copyright IBM Corporation 2017

Figure 15-13. Generating a monitor model in IBM Integration Designer

IBM Integration Designer provides many useful tools for integration with IBM Business Monitor. Developers can generate a monitor model from a module with a simple “Generate Monitor Model” tool. This tool starts a wizard, which walks the developer through a series of configuration screens. The developer chooses the module, the interfaces, and artifacts from which to generate the KPIs, and a set of predefined, specific events upon which the monitor model is built. IBM Integration Designer creates a special module to contain the monitor model, and opens a special Business Monitoring perspective so that the developer can add, edit, or further define KPIs, targets, or stopwatches in the model.

© Copyright IBM Corp. 2017

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

IBM Training

Good practices when integrating IBM Business Monitor

Optimize performance

- Use the Deployment Environment wizard when configuring WebSphere Application Server ND or clustering
- Add IBM Business Process Manager and IBM Business Monitor to the same cell
  - Simplifies configuration
  - Supports the monitor model in a process app

Add the monitor model to a process app in IBM Process Center

- Create a separate process app that contains only the monitor model
  - Limits the number of snapshots with the same monitor model
- Always update the monitor model version time stamp before publishing to IBM Process Center
  - With IBM Business Monitor, you cannot deploy the same monitor model version if it is already deployed

© Copyright IBM Corporation 2017

Figure 15-14. Good practices when integrating IBM Business Monitor

It is a good idea to author a monitor model in its own separate process app. If the monitor model is in the same process app, then every time the process app is updated, you end up with multiple process app snapshots, which contain the same monitor model. It becomes a problem at run time when you deploy all these snapshots to the server because IBM Business Monitor does not support deployment of the same monitor model version when that version is already deployed. A warning message is provided when associating a monitoring project to a process app.

On the topic of controlling a monitor model version, be sure to update the monitor model version time stamp. The monitor model version time stamp is used to set the version of the monitor model. If you do not update the time stamp, you can get an error when deploying the updated model because the version of the monitor model is the same as the deployed version.

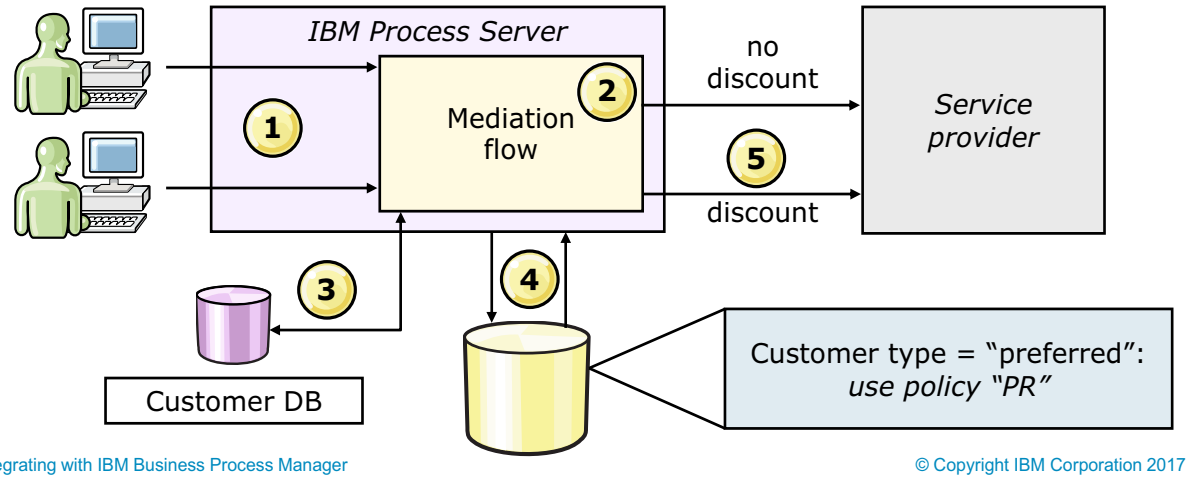
© Copyright IBM Corp. 2017

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

IBM Training

IBM Business Process Manager and WebSphere Service Registry and Repository

- Integration is supported in IBM Process Server
  - Supporting primitives includes endpoint lookup (ELP), SLA check, and policy resolution
- Configure WebSphere Service Registry and Repository connection in IBM Process Server administration console



Integrating with IBM Business Process Manager © Copyright IBM Corporation 2017

Figure 15-15. IBM Business Process Manager and WebSphere Service Registry and Repository

Mediation policies can be used to dynamically alter message processing. In this example, a preferred customer earns a discount when placing an order. The properties of mediation primitives within the mediation flow can be dynamically modified with values that are obtained from WebSphere Service Registry and Repository, as follows:

1. The customer places an order.
2. The mediation flow in IBM Process Server processes the message.
3. The mediation flow determines the type of customer by looking up the customer ID in a database. The retrieved customer type is stored in the SMO.
4. A policy resolution primitive queries the WebSphere Service Registry and Repository by using the customer type. If the customer type is "preferred", a policy match is found; otherwise, no policy match is found.
5. If policy match was found, promoted properties of primitives in the mediation flow are overridden (for example, message element setter) to set the "discount" element. The service provider applies the discount as the result of the "discount" element, which is set to a nonzero value (promoted properties whose values are derived from policies are called "dynamic properties").

© Copyright IBM Corp. 2017

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

A policy is not needed for “regular” (nonpreferred) customers; since no policy match is found in WebSphere Service Registry and Repository, no overrides are applied to the subsequent mediation primitives. This scenario does not select a different service provider, only a change in the processing within the mediation flow. Its purpose is to associate a mediation policy with a target service rather than a module, and add an endpoint lookup mediation primitive before the policy resolution mediation primitive.

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Describe integration strategies for IBM Process Designer and IBM Enterprise Content Management
- Describe integration strategies for IBM Integration Designer and IBM MobileFirst
- Describe integration strategies for IBM Process Server and IBM Operational Decision Manager
- Describe integration strategies for IBM Integration Designer and IBM Business Monitor

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 15-16. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Review questions

1. True or false: Developers can create and edit monitor models in IBM Integration Designer.
2. Which one of the following mediation primitives is not typically used to integrate with WebSphere Service Registry and Repository?
   A. SLA check
   B. Database lookup
   C. UDDI endpoint lookup
   D. Policy resolution

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 15-17. Review questions

Write your answers here:

1.

2.

© Copyright IBM Corp. 2017

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

IBM Training

Review answers

1. True or False: Developers can create and edit monitor models in IBM Integration Designer.
   The answer is False. Monitor models are generated and edited in the Business Monitor perspective.
2. Which one of the following mediation primitives is not typically used to integrate with WebSphere Service Registry and Repository?
   A. SLA check
   B. Database lookup
   C. UDDI endpoint lookup
   D. Policy resolution
   The answer is B.

© Copyright IBM Corporation 2017

Figure 15-18. Review answers

© Copyright IBM Corp. 2017

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

IBM Training

Exercise: Integrating other applications with IBM Integration Designer

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Figure 15-19. Exercise 14

© Copyright IBM Corp. 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Training

Exercise objectives

- Create a monitor model
- Import and explore a monitor model
- Associate a monitor model with a process application

© Copyright IBM Corporation 2016

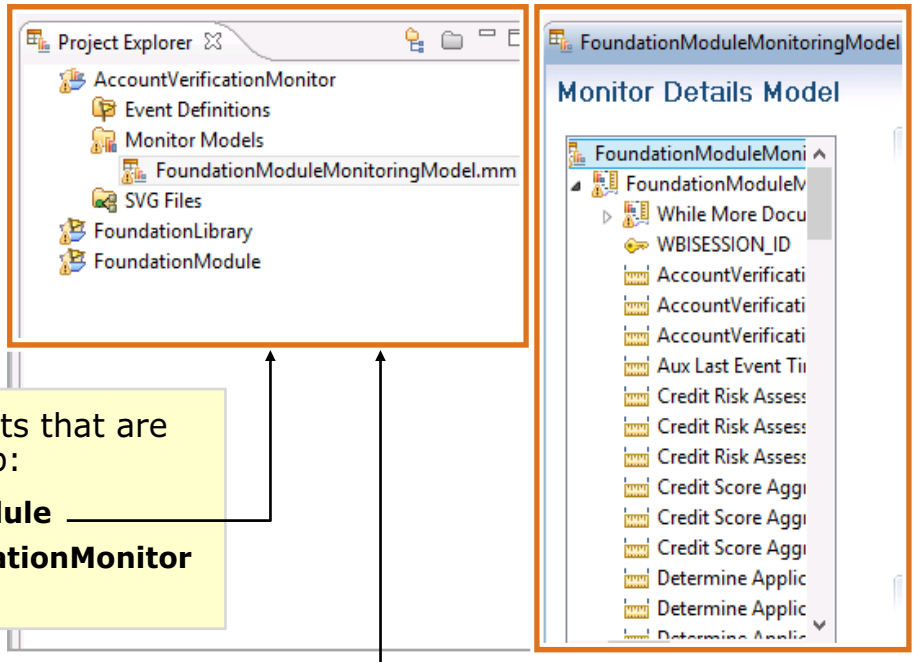
Figure 15-20. Exercise objectives

© Copyright IBM Corp. 2017

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

IBM Training

Components that are required for Exercise 14



Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **AccountVerificationMonitor** monitor model

New components that you create in the lab:

AccountVerificationMonitor monitor project

Integrating with IBM Business Process Manager

© Copyright IBM Corporation 2017

Figure 15-21. Components that are required for Exercise 14

This exercise is not intended to teach you how to build a complete monitor model with key performance indicators. It is intended to demonstrate how tools in IBM Integration Designer can be used to generate monitor models. It is also intended to demonstrate good practices when you integrate IBM Integration Designer with IBM Business Monitor.

© Copyright IBM Corp. 2017

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

Unit 16. Course summary

Estimated time

00:15

Overview

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

© Copyright IBM Corp. 2017

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

IBM Training

Unit objectives

- Explain how the course met its learning objectives
- Identify other IBM Training courses that are related to this topic
- Access the IBM Training website
- Locate appropriate resources for further study

Course summary

© Copyright IBM Corporation 2017

Figure 16-1. Unit objectives

© Copyright IBM Corp. 2017

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

IBM Training

Course objectives

- Create and modify an application by using iterative, model-driven development
- Develop a business process diagram in IBM Process Designer
- Work with a process application snapshot in IBM Integration Designer
- Associate IBM Integration Designer artifacts with a process application snapshot
- Use artifacts from an IBM Integration Designer library in a business process diagram
- Implement and deploy SCA modules and libraries that contain version information
- Use the serviceDeploy tool to install versioned modules in an IBM Process Server profile

Course summary

© Copyright IBM Corporation 2017

Figure 16-2. Course objectives

© Copyright IBM Corp. 2017

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

IBM Training

Course objectives

- Enable cross-component trace to follow the invocation sequences between SCA applications to diagnose and repair an application failure
- Implement the event sequencing quality of service qualifier in an application
- Implement a WebSphere MQ import in an application
- Implement a fault handler and compensation handler in a business process
- Examine and test a business state machine
- Examine the transaction propagation settings and transaction quality of service qualifier settings for an application
- Implement a selector component and dynamically navigate a business process instance at run time

Course summary

© Copyright IBM Corporation 2017

Figure 16-3. Course objectives

© Copyright IBM Corp. 2017

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

IBM Training

Course objectives

- Implement a static (lookup) relationship
- Create a service aggregation mediation flow by using several prebuilt mediation primitives
- Create a reusable mediation subflow that contains error handling primitives
- Configure security quality of service qualifiers and implement role-based security for human tasks and business processes
- Apply governance to process applications
- Configure IBM Business Process Manager Advanced tools for integration with other applications, such as IBM Business Monitor

Course summary

© Copyright IBM Corporation 2017

Figure 16-4. Course objectives

© Copyright IBM Corp. 2017

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

IBM Training

To learn more on the subject

- IBM Training website:
  www.ibm.com/training
- IBM Redbooks:
  www.redbooks.ibm.com
- To stay informed about IBM training, go to the following sites:
  - YouTube: youtube.com/IBMTraining
  - Facebook: facebook.com/ibmtraining

Course summary

© Copyright IBM Corporation 2017

Figure 16-5. To learn more on the subject

© Copyright IBM Corp. 2017

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

IBM Training

Identify other IBM Training courses

- Training paths: http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&c=a0003096
- For administration and deployment of IBM Business Process Manager, see:
  - WB871: *Administration of IBM Business Process Manager Advanced V8.5.7*
- For IBM Business Process Manager troubleshooting and performance, see:
  - WB869: *IBM Business Process Manager V8.5 Problem Determination*
- For IBM BPM Standard courses, see:
  - WB823/ZB823: *Process Implementing with IBM Business Process Manager Standard V8.5.7 – I*
  - WB824/ZB824: *Process Implementing with IBM Business Process Manager Standard V8.5.7 – II*

Course summary

© Copyright IBM Corporation 2017

Figure 16-6. Identify other IBM Training courses

© Copyright IBM Corp. 2017

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

IBM Training

Earn an IBM Badge

- After completing this course, you are ready to take the IBM Badge test *IBM Business Process Manager Advanced V8.5.7 Developer*
- Use IBM Badges to share verified proof of your IBM credentials
- Find your Badge test on this site:
  - https://www.ibm.com/services/learning/ites.wss/zzen?pageType=badgesearch
- The *IBM Business Process Manager Advanced V8.5.7 Developer* Badge test requires these courses:
  - **WB860/ZB860**: *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - I*
  - **WB861/ZB861**: *Developing Applications in IBM Business Process Manager Advanced V8.5.7 - II*
- After completing **both** of these courses, take the Badge test

Course summary

© Copyright IBM Corporation 2017

Figure 16-7. Earn an IBM Badge

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

IBM Training

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
  - Product information
  - Training and certification
  - Documentation
  - Support
  - Technical information

- To learn more, see the IBM Cloud Education Resource Guide:
  - www.ibm.biz/CloudEduResources

Course summary

© Copyright IBM Corporation 2017

Figure 16-8. Enhance your learning with IBM resources

© Copyright IBM Corp. 2017

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

IBM Training

Unit summary

- Explain how the course met its learning objectives
- Identify other IBM Training courses that are related to this topic
- Access the IBM Training website
- Locate appropriate resources for further study

Course summary

© Copyright IBM Corporation 2017

Figure 16-9. Unit summary

© Copyright IBM Corp. 2017

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

IBM Training

Course completion

You have completed this course:

Developing Applications in IBM Business Process Manager Advanced V8.5.7 - II

Do you have any questions?

Course summary

© Copyright IBM Corporation 2017

Figure 16-10. Course completion

© Copyright IBM Corp. 2017

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

Appendix A. List of abbreviations

2PC 2-phase commit

A

ACID atomicity, consistency, isolation, and durability
ACORD Agent-Company Organization for Research and Development
ACORD Association for Cooperative Operations Research and Development
AIS Advanced Integration services
Ant Another Neat Tool
API application programming interface
APQC American Productivity and Quality Center
ASBO application-specific business object

B

B2B business-to-business
BAL Business Action Language
BAM business activity monitoring
BDD business-driven development
BFM business flow management
BFM Business Flow Manager
BG business graph
BI business integration
BO business object
BPD business process definition
BPD business process diagram
BPEL Business Process Execution Language
BPM business process management
BPMN Business Process Model and Notation
BRG business rule group
BRMS business rule management system
BSM business state machine

C

CEI Common Event Infrastructure
CICS Customer Information Control System
CMIS Content Management Interoperability Services
COBOL Common Business Oriented Language
CRON Chronograph
CSP communications service provider
CSV comma-separated values
CVS Concurrent Versions System

© Copyright IBM Corp. 2017

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

D

DB database
DB2 Database 2
DN distinguished name
DOS disk operating system

E

EAI enterprise application integration
EAR enterprise archive
ECM Enterprise Content Management
EDI electronic data interchange
EE Enterprise Edition
EIS enterprise information system
EJB Enterprise JavaBeans
ELP endpoint lookup primitive
EPR endpoint reference
ES event sequencing
ESB enterprise service bus

F

FFDC first-failure data capture
FIFO first-in first-out
FK foreign key
FP fix pack

G

GB gigabyte
GBO generic business object
GPS global positioning system

H

HATS Host Access Transformation Services
HL7 Health Level 7
HR human resources
HT human task
HTTP Hypertext Transfer Protocol

I

IA industry accelerator
IBM International Business Machines Corporation
ID identification data
IE Internet Explorer
IID instance ID
IIOP Internet Inter-ORB Protocol
IMS Information Management System
I/O input/output

© Copyright IBM Corp. 2017

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

IP Internet Protocol
ISC Integrated Solutions Console
ISO International Organization for Standardization
IT information technology

J

J2C Java 2 Connector
JAAS Java Authentication and Authorization Service
JACC Java Authorization Contract for Containers
Jacl Java Command Language
JAR Java archive
JAX Java API for XML
JAXB Java Architecture for XML Binding
JAX-RPC Java API for XML-based RPC
JAX-WS Java API for XML Web Services
JCA Java Connector architecture
JDBC Java Database Connectivity
JEE Java Enterprise Edition
JMS Java Message Service
JNDI Java Naming and Directory Interface
JSF JavaServer Faces
JSON JavaScript Object Notification
JSP JavaServer Pages
JVM Java virtual machine

K

KPI key performance indicator

L

LDAP Lightweight Directory Access Protocol
LOB line of business
LTPA Lightweight Third Party Authentication

M

MDA model-driven architecture
MDB message-driven bean
MDD model-driven development
MQ Message Queue
MQI Message Queue Interface

N

ND Network Deployment

O

OAGIS Open Applications Group Integration Specification
OASIS Organization for the Advancement of Structured Information Standards

© Copyright IBM Corp. 2017

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

ODM Object Database Manager
ORB Object Request Broker
OS operating system
OSOA open service-oriented architecture

P

PDF Portable Document Format
PI process instance
PIID process instance ID
PLM Product Life Cycle Management
POJO plain old Java object

Q

QoS quality of service

R

RAR resource adapter archive
RAS reusable asset specification
RIID relationship instance ID
RMI Remote Method Invocation
RMI/IIOP Remote Method Invocation over Internet Inter-ORB Protocol
RPC Remote Procedure Call

S

SACL State Adaptive Choreography Language
SAP Systems Applications and Products (data processing)
SCA Service Component Architecture
SCDL Service Component Definition Language
SCM software configuration management
SCM source control management
SDK software development kit
SDO Service Data Object
SIB service integration bus
SIBus service integration bus
SLA service level agreement
SME subject matter expert
SMO service message object
SOA service-oriented architecture
SOAP A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. Usage note: SOAP is not an acronym; it is a word (formerly an acronym for Simple Object Access Protocol)
SOI service-oriented integration
SQL Structured Query Language
SSL Secure Sockets Layer
SSO single sign-on
STP straight through processing
SVG Scalable Vector Graphics
SWAM Simple WebSphere Authentication Mechanism

© Copyright IBM Corp. 2017

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

T

TEL Task Execution Language
TIBCO The Information Bus Company

U

UCM Unified Change Management
UDDI Universal Description, Discovery, and Integration
UI user interface
UML Unified Modeling Language
URI Uniform Resource Identifier
URL Uniform Resource Locator
UTC Coordinated Universal Time
UTE unit test environment
UUID Universally Unique Identifier

V

VMM virtual member manager
VOB versioned object database

W

WS web services
WS-BPEL Web Services Business Process Execution Language
WSDL Web Services Description Language
WSRR WebSphere Service Registry and Repository
WS-Security Web Services Security

X

XA Extended Architecture
XCT cross-component trace
XML Extensible Markup Language
XPath XML Path Language
XSD XML Schema Definition
XSL Extensible Stylesheet Language
XSLT Extensible Stylesheet Language Transformation

© Copyright IBM Corp. 2017

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

Appendix B. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: http://www.ibm.com/training
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: http://www.ibm.com/blogs/ibm-training
- **IBM Certification**
  - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
  - For more information, see: http://www.ibm.com/certify
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
  - For more information, see: https://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&c=a0003096

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the latest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
  - Receive concise updates from Middleware Education a few times each week.
  - Follow Middleware Education at: twitter.com/IBMCloudEdu
- **Facebook**:

© Copyright IBM Corp. 2017

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

  - Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
  - Find IBM Training at: facebook.com/ibmtraining
- **YouTube**:
  - See the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: youtube.com/IBMTraining

Support

- **Middleware Support portal**
  - The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARS). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
  - For more information, see: http://www.ibm.com/software/websphere/support
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: http://www.ibm.com/software/support/isa
- **IBM Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.
  - For more information, see: http://www.ibm.com/software/info/education/assistant/

Middleware documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.

© Copyright IBM Corp. 2017

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

  - For more information, see: http://www.ibm.com/redbooks
- **IBM documentation and libraries**
  - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
  - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
  - For more information, see: http://www.ibm.com/developerworks

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see: http://www.ibm.com/services/us/en/it-services/systems/middleware-services/

© Copyright IBM Corp. 2017

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

IBM Training

© Copyright International Business Machines Corporation 2017.