

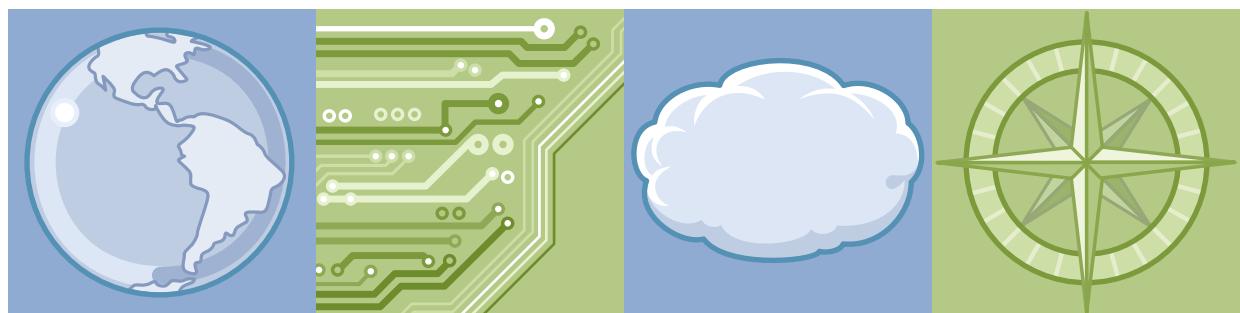


IBM Training

Student Notebook

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

Course code WB860 ERC 1.2



IBM Cloud
Middleware

Trademarks

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

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

AIX®	CICS®	ClearCase®
DataPower®	DB™	DB2®
developerWorks®	Express®	IMS™
Initiate®	Lombardi®	Notes®
PartnerWorld®	Power®	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
Sametime®	WebSphere®	z/OS®

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

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

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

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

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

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.

June 2016 edition

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

Contents

Trademarks	xxi
Course description	xxv
Agenda	xxvii
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	1-29
IBM Process Center	1-30
Process Portal	1-31
IBM Process Server (1 of 2)	1-33
IBM Process Server (2 of 2)	1-34
IBM Process Server components	1-35
User roles and tools in a typical deployment scenario	1-36
Introduction to IBM BPM on Cloud	1-37
BPM on Cloud customer focus: Manage and automate decisions	1-38
IBM BPM on Cloud: Three runtime environments	1-39
IBM BPM on Cloud user portal	1-40
Unit summary	1-41
Checkpoint questions	1-42
Checkpoint answers	1-43
References	1-44

Unit 2. Overview of IBM Business Process Manager Advanced V8.5.7	2-1
Unit objectives	2-2
Topics	2-3
2.1. BPD and BPEL	2-5
BPD and BPEL	2-6
Two kinds of business processes: BPD and BPEL	2-7
BPMN	2-8
BPMN process in Process Designer (desktop interface)	2-10
BPMN process in Process Designer (web-based interface)	2-12
BPEL	2-13
BPEL process in IBM Integration Designer	2-14
When to use BPMN?	2-15
When to use BPEL?	2-16
2.2. IBM BPM Standard or IBM BPM Advanced?	2-19
IBM BPM Standard or IBM BPM Advanced?	2-20
IBM BPM Advanced contains all of IBM BPM Standard	2-21
When to use IBM BPM Standard?	2-22
When to use IBM BPM Advanced?	2-23
Use IBM Process Designer or IBM Integration Designer?	2-25
Artifacts in IBM Process Designer and IBM Integration Designer tool	2-26
2.3. Key components in IBM BPM Advanced	2-27
Key components in IBM BPM Advanced	2-28
Service Component Architecture (SCA)	2-29
Business value of Service Component Architecture (SCA)	2-30
Business rules	2-31
Business value of rules	2-32
Adapters	2-33
Business value of adapters	2-34
Mediations	2-35
Business value of mediations	2-36
Account verification business process that you build in this course	2-37
Account verification process: BPD or BPEL?	2-39
Course business scenario: What you build (1 of 2)	2-40
Course business scenario: What you build (2 of 2)	2-41
Unit summary	2-42
Checkpoint questions	2-43
Checkpoint answers	2-44
Exercise 1	2-45
Exercise objectives	2-46
Account verification process: Overview	2-47
Course business scenario: What you build	2-49
Course business scenario: What you work with	2-50
Components that are required for Exercise 1	2-51
Collaborate by using Process Portal in Exercise 1	2-52
Unit 3. Course business scenario: What you build	3-1
Unit objectives	3-2
Some important points	3-3
WB860 lab environment	3-4
Exercise 1	3-5
What are some key tasks that you did in Exercise 1?	3-6

Components that are required for Exercise 1	3-7
Account verification process: Overview	3-8
Collaborate by using Process Portal in Exercise 1	3-10
Course business scenario: What you worked with in Exercise 1 (1 of 2)	3-11
Course business scenario: What you worked with in Exercise 1 (2 of 2)	3-12
Exercise 2	3-13
What are some key tasks that you do in Exercise 2?	3-14
Components that are required for Exercise 2	3-15
Explore IBM Integration Designer in Exercise 2	3-16
Course business scenario: What you work with in Exercise 2	3-17
Exercise 3	3-18
What are some key tasks that you do in Exercise 3?	3-19
Components that are required for Exercise 3	3-20
Use cross-component tracing in Exercise 3	3-21
Test by using the Business Process Choreographer Explorer	3-22
Course business scenario: What you work with in Exercise 3	3-23
Exercise 4	3-24
What are some key tasks that you do in Exercise 4?	3-25
Components that are required for Exercise 4 (Section 1)	3-26
Invoke an external web service in Exercise 4 (Section 1)	3-27
Components that are required for Exercise 4 (Section 2)	3-28
Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)	3-29
Course business scenario: What you work with in Exercise 4	3-30
Exercise 5	3-31
What are some key tasks that you do in Exercise 5?	3-32
Components that are required for Exercise 5 (1 of 2)	3-33
Components that are required for Exercise 5 (2 of 2)	3-34
Course business scenario: What you work with in Exercise 5	3-35
Exercise 6	3-36
What are some key tasks that you do in Exercise 6?	3-37
Account verification process: Overview	3-38
Components that are required for Exercise 6 (1 of 7)	3-39
Components that are required for Exercise 6 (2 of 7)	3-40
Components that are required for Exercise 6 (3 of 7)	3-41
Components that are required for Exercise 6 (4 of 7)	3-42
Components that are required for Exercise 6 (5 of 7)	3-44
Components that are required for Exercise 6 (6 of 7)	3-45
Components that are required for Exercise 6 (7 of 7)	3-46
Course business scenario: What you work with in Exercise 6	3-47
Exercise 7	3-48
What are some key tasks that you do in Exercise 7?	3-49
Components that are required for Exercise 7 (1 of 2)	3-50
Components that are required for Exercise 7 (2 of 2)	3-51
Code snippets that are created in the AccountVerification process in Exercise 7	3-53
Complete Account verification process (1 of 4)	3-54
Complete Account verification process (2 of 4)	3-55
Complete Account verification process (3 of 4)	3-56
Complete Account verification process (4 of 4)	3-57
Course business scenario: What you work with in Exercise 7	3-58
Exercise 8	3-59

What are some key tasks that you do in Exercise 8?	3-60
Components that are required for Exercise 8	3-61
Create data map in Exercise 8 (1 of 2)	3-62
Create data map in Exercise 8 (2 of 2)	3-63
Assemble SCA application that contains the business process in Exercise 8	3-64
Test the created application by deploying to a Process Server runtime	3-65
Use the test client in Exercise 8	3-66
Course business scenario: What you work with in Exercise 8	3-67
Exercise 9	3-68
What are some key tasks that you do in Exercise 9?	3-69
Components that are required for Exercise 9	3-70
Create rule group in Exercise 9	3-72
Course business scenario: What you work with in Exercise 9	3-73
Exercise 10	3-74
What are some key tasks that you do in Exercise 10?	3-75
Components that are required for Exercise 10 (1 of 2)	3-76
Components that are required for Exercise 10 (2 of 2)	3-77
Assemble SCA components in Exercise 10	3-78
Course business scenario: What you work with in Exercise 10	3-80
Exercise 11	3-81
What are some key tasks that you do in Exercise 11?	3-82
Components that are required for Exercise 11	3-83
Assemble SCA components in Exercise 11	3-84
Course business scenario: What you work with in Exercise 11	3-86
Exercise 12	3-87
What are some key tasks that you do in Exercise 12?	3-88
Components that are required for Exercise 12 (1 of 2)	3-89
Components that are required for Exercise 12 (2 of 2)	3-90
Create the RouteRequest mediation flow in Exercise 12	3-91
Assemble SCA components in Exercise 12 (1 of 2)	3-92
Assemble SCA components in Exercise 12 (2 of 2)	3-93
Course business scenario: What you work with in Exercise 12	3-94
Exercise 13	3-95
What are some key tasks that you do in Exercise 13?	3-96
Account verification process: Overview	3-97
Create business space in Exercise 13	3-98
Course business scenario: What you work with in Exercise 13	3-99
Exercise 14	3-100
What are some key tasks that you do in Exercise 14?	3-101
Components that are required for Exercise 14	3-102
Course business scenario: What you work with in Exercise 14	3-103
Exercise 15	3-104
What are some key tasks that you do in Exercise 15?	3-105
Components that are required for Exercise 15	3-106
Course business scenario: What you work with in Exercise 15	3-107
Exercise 16	3-108
What are some key tasks that you do in Exercise 16?	3-109
Components that are required for Exercise 16	3-110
Course business scenario: What you work with in Exercise 16	3-111
Exercise 17	3-112
What are some key tasks that you do in Exercise 16?	3-113

Course business scenario: What you work with in Exercise 17	3-114
Course business scenario: What you work with in Exercise 17	3-115
Unit summary	3-116
Unit 4. IBM Integration Designer overview.....	4-1
Unit objectives	4-2
Topics	4-3
4.1. Introduction to IBM Integration Designer.....	4-5
Introduction to IBM Integration Designer	4-6
IBM Integration Designer: Create, assemble, and test applications	4-7
IBM Integration Designer roles	4-8
Installing IBM Business Process Manager Advanced V8.5.7	4-10
IBM Installation Manager	4-12
IBM Integration Designer: Platform architecture	4-14
Installing IBM Integration Designer	4-16
4.2. Application composition and deployment	4-19
Application composition and deployment	4-20
Getting Started	4-21
Application composition and deployment	4-22
Business Integration perspective	4-23
Features overview: Visual tools (1 of 4)	4-25
Features overview: Visual tools (2 of 4)	4-26
Features overview: Visual tools (3 of 4)	4-27
Features overview: Visual tools (4 of 4)	4-28
IBM Process Server unit test environment (1 of 3)	4-29
IBM Process Server unit test environment (2 of 3)	4-31
IBM Process Server unit test environment (3 of 3)	4-32
Module	4-33
Module components	4-34
Mediation Module	4-35
Mediation module components	4-36
Library components	4-38
Creating modules and libraries	4-39
Library deployment	4-40
Business Integration view	4-41
Dependency editor	4-42
Packaging modules	4-43
Module deployment	4-44
Application, module, and library relationship	4-46
Using the process center perspective	4-47
Building projects	4-48
Artifacts in IBM Integration Designer and Process Designer	4-49
4.3. Documentation and organization capabilities	4-51
Documentation and organization capabilities	4-52
Documentation and organization capabilities	4-53
Documentation: Sticky notes	4-54
Documentation: Java task tags	4-55
Organization: Integration solution (1 of 2)	4-56
Organization: Integration solution (2 of 2)	4-57
Unit summary	4-58
Checkpoint questions	4-59

Checkpoint answers	4-60
Exercise 2	4-61
Exercise objectives	4-62
Course business scenario: What you work with in Exercise 2	4-63
Components that are required for Exercise 2	4-64
Explore IBM Integration Designer in Exercise 2	4-65
References	4-66
Unit 5. Service Component Architecture overview	5-1
Unit objectives	5-2
Topics	5-3
5.1. Overview of SCA modules	5-5
Overview of SCA modules	5-6
Service Component Architecture	5-7
Overview and business value of SCA	5-8
SCA components	5-10
SCA component definition	5-12
SCA component interfaces	5-13
SCA component references	5-15
SCA module: Overview	5-17
References and wires	5-18
Exports	5-19
Imports	5-20
Example: Using imports and exports	5-22
SCA module components: Stand-alone references	5-23
Example: HelloWorld	5-25
Assembly Diagram: Components	5-26
Composing SCA modules: Assembly Diagram editor	5-27
5.2. Testing SCA applications	5-29
Testing SCA applications	5-30
Testing SCA components in the integration test client	5-31
Test configurations	5-33
Emulators	5-34
Monitors (1 of 2)	5-35
Monitors (2 of 2)	5-38
Entering data for testing	5-39
Using a test data table for testing	5-40
Using a stand-alone data pool for testing	5-41
Overview: Testing SCA components (1 of 2)	5-42
Overview: Testing SCA components (2 of 2)	5-44
Testing SCA modules in the integration test client	5-45
Overview: Testing SCA modules (1 of 2)	5-46
Overview: Testing SCA modules (2 of 2)	5-48
Using the Servers view when testing modules	5-49
Server Logs view	5-50
Cross-component tracing	5-51
Unit summary	5-53
Checkpoint questions	5-54
Checkpoint answers	5-55
Exercise 3	5-56
Exercise objectives	5-57

Course business scenario: What you work with in Exercise 3	5-58
Components that are required for Exercise 3	5-59
Use cross-component tracing in Exercise 3	5-60
Test by using the Business Process Choreographer Explorer	5-61
References	5-63
Unit 6. SCA bindings	6-1
Unit objectives	6-2
Topics	6-3
6.1. SCA bindings.....	6-5
SCA bindings	6-6
Incoming and outgoing interactions with external components	6-7
SCA invocation styles	6-9
Import and export resources	6-10
Quality of service qualifiers	6-11
Export binding types	6-12
Import binding types	6-13
JCA adapter EIS bindings (1 of 2)	6-14
JCA adapter EIS bindings (2 of 2)	6-16
Enterprise JavaBeans (EJB) binding (1 of 2)	6-17
Enterprise JavaBeans (EJB) binding (2 of 2)	6-18
WSDL to Java interface mapping for EJB beans	6-19
SCA bindings (1 of 2)	6-20
SCA bindings (2 of 2)	6-21
JMS bindings (1 of 2)	6-22
JMS bindings (2 of 2)	6-23
Generic JMS bindings	6-24
WebSphere MQ bindings (1 of 2)	6-25
WebSphere MQ bindings (2 of 2)	6-26
HTTP bindings	6-27
6.2. Web services	6-31
Web services	6-32
Overview: Web services	6-33
Discovering web services	6-35
Importing a web service WSDL interface	6-37
Web service export binding (1 of 2)	6-38
Web service export binding (2 of 2)	6-40
Testing web services	6-42
Unit summary	6-43
Checkpoint questions	6-44
Checkpoint answers	6-45
Exercise 4	6-46
Exercise objectives	6-47
Course business scenario: What you work with in Exercise 4	6-48
Components that are required for Exercise 4 (Section 1)	6-49
Invoke an external web service in Exercise 4 (Section 1)	6-51
Components that are required for Exercise 4 (Section 2)	6-52
Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)	6-53
References	6-54

Unit 7. Business objects	7-1
Unit objectives	7-2
Topics	7-3
7.1. Introduction to Service Data Objects	7-5
Introduction to Service Data Objects	7-6
Introduction to Service Data Objects (SDO)	7-7
Data objects	7-9
Data graphs	7-10
Example: SDO data tree	7-11
7.2. Introduction to business objects	7-13
Introduction to business objects	7-14
Business objects are an SOA core component	7-15
Introduction to the business object framework	7-16
Business objects that are represented as SDOs	7-17
Business objects: Data abstraction for SCA	7-18
Composition of business object framework	7-19
Business objects (1 of 2)	7-21
Business objects (2 of 2)	7-23
Creating business objects	7-24
Business graphs	7-26
Business graph: Change summary	7-27
Data objects in IBM Business Process Manager	7-29
Unit summary	7-30
Checkpoint questions	7-31
Checkpoint answers	7-32
Supplemental note: Business object framework runtime environment	7-33
Exercise 5	7-34
Exercise objectives	7-35
Course business scenario: What you work with in Exercise 5	7-36
Components that are required for Exercise 5 (1 of 2)	7-37
Components that are required for Exercise 5 (2 of 2)	7-38
References	7-39
 Unit 8. Business process choreography overview	 8-1
Unit objectives	8-2
Topics	8-3
8.1. Introduction to Web Services Business Process Execution Language (WS-BPEL)	8-5
Introduction to Web Services Business Process Execution Language (WS-BPEL)	8-6
Business processes are service components	8-7
Business processes in IBM Business Process Manager	8-8
Overview: Business Process Execution Language (1 of 2)	8-10
Overview: Business Process Execution Language (2 of 2)	8-11
BPEL interoperates with SCA and SDO	8-13
Microflows versus long-running processes	8-14
8.2. Elements of BPEL processes	8-17
Elements of BPEL processes	8-18
Main elements and concepts of a BPEL process	8-19
Elements of a business process defined	8-20
Process elements: Partners	8-21
Process elements: Variables	8-22
Process elements: Correlation sets (1 of 2)	8-24

Process elements: Correlation sets (2 of 2)	8-25
Process elements: Activities	8-27
Unit summary	8-28
Checkpoint questions	8-29
Checkpoint answers	8-30
Exercise 6	8-31
Exercise objectives	8-32
Course business scenario: What you work with in Exercise 6	8-33
Account verification process: Overview	8-34
Components that are required for Exercise 6 (1 of 7)	8-35
Components that are required for Exercise 6 (2 of 7)	8-36
Components that are required for Exercise 6 (3 of 7)	8-37
Components that are required for Exercise 6 (4 of 7)	8-38
Components that are required for Exercise 6 (5 of 7)	8-40
Components that are required for Exercise 6 (6 of 7)	8-41
Components that are required for Exercise 6 (7 of 7)	8-42
Unit 9. Business process basic and structured activities	9-1
Unit objectives	9-2
Topics	9-3
9.1. WS-BPEL basic activities	9-5
WS-BPEL basic activities	9-6
Overview of WS-BPEL activities	9-7
Basic activities: Receive, reply, and invoke	9-8
Basic activities: Assign	9-9
Basic activities: Receive choice (1 of 2)	9-10
Basic activities: Receive choice (2 of 2)	9-11
Basic activities: Wait	9-12
Basic activities: Human task	9-13
Basic activities: Snippets	9-14
Basic activities: Empty action	9-15
Basic activities: Data map	9-16
Creating a data map	9-18
Common XPath editor	9-20
Activity exit conditions	9-21
9.2. WS-BPEL structured activities	9-23
WS-BPEL structured activities	9-24
Structured activities: Sequences and parallel activities	9-25
Links and link conditions	9-26
Structured activities: Choice	9-27
Structured activities: While loop	9-28
Structured activities: Repeat until loop	9-29
Structured activities: For each	9-30
Structured activities: Generalized flow (1 of 2)	9-31
Structured activities: Generalized flow (2 of 2)	9-33
Structured activities: Scope	9-34
Structured activities: Collaboration scope (1 of 2)	9-35
Structured activities: Collaboration scope (2 of 2)	9-36
Unit summary	9-37
Checkpoint questions	9-38
Checkpoint answers	9-39

Exercise 7	9-40
Exercise objectives	9-41
Course business scenario: What you work with in Exercise 7	9-42
Components that are required for Exercise 7 (1 of 2)	9-43
Components that are required for Exercise 7 (2 of 2)	9-44
Code snippets that are created in the AccountVerification process in Exercise 7	9-46
Complete Account verification process (1 of 4)	9-47
Complete Account verification process (2 of 4)	9-48
Complete Account verification process (3 of 4)	9-49
Complete Account verification process (4 of 4)	9-50
References	9-51
Unit 10. Business process handlers, runtime behavior, and clients	10-1
Unit objectives	10-2
Topics	10-3
10.1. WS-BPEL handlers and error processing activities	10-5
WS-BPEL handlers and error processing activities	10-6
Handlers and error processing activities: Throw and rethrow	10-7
Handlers and error processing activities: Fault handler	10-9
When a fault is thrown	10-10
Error processing: Fault links	10-12
Handlers and error processing activities: Compensate	10-13
Handlers and error processing activities: Compensation handlers	10-15
Handlers and error processing activities: Event handlers	10-16
Handlers and error processing activities: Terminate	10-18
10.2. Runtime aspects of BPEL processes	10-19
Runtime aspects of BPEL processes	10-20
Business process runtime support	10-21
Business process templates and instances	10-23
Microflow and long-running processes at run time	10-24
Clients for business processes	10-26
Business Process Choreographer Explorer client	10-28
Runtime process dynamicity (1 of 2)	10-30
Runtime process dynamicity (2 of 2)	10-31
Installing and uninstalling business process applications	10-32
Deletion of completed instances	10-34
Unit summary	10-35
Checkpoint questions	10-36
Checkpoint answers	10-37
Exercise 8	10-38
Exercise objectives	10-39
Course business scenario: What you work with in Exercise 8	10-40
Components that are required for Exercise 8	10-41
Create data map in Exercise 8 (1 of 2)	10-42
Create data map in Exercise 8 (2 of 2)	10-43
Assemble the SCA application that contains the business process in Exercise 8	10-44
Test the created application by deploying to a Process Server runtime	10-45
Use the test client in Exercise 8	10-46
References	10-47

Unit 11. Business rules	11-1
Unit objectives	11-2
Topics	11-3
11.1. Overview of business rules	11-5
Overview of business rules	11-6
Business rules are a service component	11-7
What is a business rule?	11-8
Business rule actions	11-10
Business value of rules	11-11
Rule sets defined	11-13
Rule sets in IBM Integration Designer (1 of 2)	11-14
Rule sets in IBM Integration Designer (2 of 2)	11-15
Decision table	11-17
Decision tables in IBM Integration Designer (1 of 2)	11-18
Decision tables in IBM Integration Designer (2 of 2)	11-19
Rule sets versus decision tables	11-21
Rule groups	11-22
Rule group components	11-24
Business rule group properties	11-26
Display names and description fields for rule groups	11-27
Business rules: Tools (1 of 2)	11-28
Business rules: Tools (2 of 2)	11-29
Business rules in IBM Process Designer	11-30
11.2. Runtime aspects of rule groups	11-31
Runtime aspects of rule groups	11-32
Business rules at run time	11-33
More runtime aspects of business rules	11-34
Designer and developer roles in business rules	11-35
Business rules manager web client	11-36
Templates in the business rules manager	11-37
Exporting and importing rule changes	11-38
Business rule auditing	11-39
Public business rules API	11-40
Unit summary	11-41
Checkpoint questions	11-42
Checkpoint answers	11-43
Exercise 9	11-44
Exercise objectives	11-45
Course business scenario: What you work with in Exercise 9	11-46
Components that are required for Exercise 9	11-47
Create rule group in Exercise 9	11-49
References	11-50
Unit 12. Adapters	12-1
Unit objectives	12-2
Topics	12-3
12.1. Introduction to adapters	12-5
Introduction to adapters	12-6
Adapters are supporting services	12-7
Introduction to adapters	12-8
Adapter components	12-9

WebSphere (JCA) Adapters	12-11
Available WebSphere (JCA) application adapters	12-12
Architecture	12-13
Available WebSphere (JCA) technology adapters	12-14
Advantages of Java EE Connector Architecture (1 of 2)	12-15
Advantages of Java EE Connector Architecture (2 of 2)	12-17
12.2. Using WebSphere (JCA) Adapters	12-19
Using WebSphere (JCA) Adapters	12-20
Using WebSphere Adapters	12-21
Enterprise Metadata Discovery	12-22
Discovered interactions	12-24
Data Exchange Service Provider Interface	12-25
Application adapters and event polling	12-26
Unit summary	12-27
Checkpoint questions	12-28
Checkpoint answers	12-29
Exercise 10	12-30
Exercise objectives	12-31
Course business scenario: What you work with in Exercise 10	12-32
Components that are required for Exercise 10 (1 of 2)	12-33
Components that are required for Exercise 10 (2 of 2)	12-34
Assemble SCA components in Exercise 10	12-35
References	12-37
Unit 13. Developing mediation services	13-1
Unit objectives	13-2
Topics	13-3
13.1. Mediation services	13-5
Mediation services	13-6
Mediation flows are supporting services	13-7
What does an enterprise service bus do? (1 of 2)	13-8
What does an enterprise service bus do? (2 of 2)	13-9
Key concepts	13-10
Example: Mediation module and mediation flow	13-11
Typical task to implement a mediation flow	13-12
Mediation module and mediation flow: Tools (1 of 2)	13-13
Mediation module and mediation flow: Tools (2 of 2)	13-14
Mediation flow editor: Overview tab	13-15
Mediation flow editor: Request flow view	13-16
Mediation flow editor: Response flow view	13-17
Mediation flow editor: Correspondence between Overview and Flow views	13-18
Terminals in the mediation flow editor	13-19
Wiring of mediation primitive terminals	13-20
Example of wiring of mediation primitive terminals	13-21
Mediation flow editor: Properties view	13-22
Mediations subflow: Overview	13-23
Mediations subflow: Implementation	13-25
Promoted properties: Overview	13-26
Promoted properties: Administrative management	13-27
13.2. Service message objects	13-29
Service message objects	13-30

Overview: SDO, SMO, business objects	13-31
Service message object (SMO)	13-32
SMO structure: Headers	13-33
SMO structure: Correlation and transient contexts	13-34
SMO structure: FaultInfo context	13-35
SMO structure: Body	13-36
SMO in a mediation flow	13-37
Manipulating SMOs (1 of 2)	13-39
Manipulating SMOs (2 of 2)	13-40
Unit summary	13-42
Checkpoint questions	13-43
Checkpoint answers	13-44
Exercise 11	13-45
Exercise objectives	13-46
Course business scenario: What you work with in Exercise 11	13-47
Components that are required for Exercise 11	13-48
Assemble SCA components in Exercise 11	13-49
Unit 14. Mediation primitives.....	14-1
Unit objectives	14-2
Topics	14-3
14.1. Message transformation and enrichment primitives	14-5
Message transformation and enrichment primitives	14-6
What is a mediation primitive?	14-7
Common features of mediation primitives	14-9
Mediation primitives: Transformation (1 of 2)	14-10
Mediation primitives: Transformation (2 of 2)	14-11
Mapping mediation primitive	14-12
Message Element Setter primitive	14-14
Message header setter primitives	14-15
Custom Mediation primitive	14-16
14.2. Flow control primitives.....	14-17
Flow control primitives	14-18
Mediation primitives: Routing (1 of 2)	14-19
Mediation primitives: Routing (2 of 2)	14-20
Message Filter primitive	14-21
Message Filter primitive: Configuration	14-22
Service Invoke primitive: Overview	14-24
Service Invoke primitive compared to callout node	14-25
Type Filter primitive	14-26
14.3. Faults, tracing, and error handling primitives.....	14-27
Faults, tracing, and error handling primitives	14-28
Faults	14-29
WSDL faults: Modeled	14-30
WSDL faults: Unmodeled	14-31
Handling WSDL faults: Request flow	14-32
Handling WSDL faults: Response flow	14-33
Mediation primitives: Error handling, debugging, and event recording	14-34
Message Logger primitive	14-35
Trace primitive	14-37
Stop primitive	14-38

Fail primitive	14-39
Unit summary	14-40
Checkpoint questions	14-41
Checkpoint answers	14-42
Exercise 12	14-43
Exercise objectives	14-44
Course business scenario: What you work with in Exercise 12	14-45
Components that are required for Exercise 12 (1 of 2)	14-46
Components that are required for Exercise 12 (2 of 2)	14-47
Create the RouteRequest mediation flow in Exercise 12	14-48
Assemble SCA components in Exercise 12 (1 of 2)	14-49
Assemble SCA components in Exercise 12 (2 of 2)	14-50
Unit 15. Business Space	15-1
Unit objectives	15-2
15.1. Topics	15-3
Topics	15-4
15.2. Introduction to Business Space	15-5
Introduction to Business Space	15-6
Problem statement that Business Space addresses	15-7
What is Business Space?	15-8
Business Space: A unified environment	15-9
Key roles (participants) in business process management	15-11
Integration with the BPM portfolio	15-12
15.3. Business Space terminology	15-13
Business Space terminology	15-14
Business Space hierarchy	15-15
Pages (1 of 2)	15-16
Pages (2 of 2)	15-18
Templates (1 of 2)	15-19
Templates (2 of 2)	15-20
Templates available in IBM Process Manager Advanced	15-21
Business space	15-23
Widget	15-25
Problem determination widgets	15-27
Human Task Management widget	15-29
Tasks widget	15-30
WebSphere Service Registry and Repository widgets	15-31
IBM Business Monitor widgets	15-32
15.4. Configure a new space	15-33
Configure a new space	15-34
Create a space (1 of 3)	15-35
Create a space (2 of 3)	15-36
Create a space (3 of 3)	15-38
Customizing a business space	15-39
Working with business spaces	15-41
Unit summary	15-43
Checkpoint questions	15-44
Checkpoint answers	15-45
Exercise 13	15-46
Exercise objectives	15-47

Course business scenario: What you work with in Exercise 13	15-48
Account verification process: Overview	15-49
Create business space in Exercise 13	15-51
Unit 16. Advanced testing	16-1
Unit objectives	16-2
Topics	16-3
16.1. Advanced testing	16-5
Advanced testing	16-6
Testing modules	16-7
Unit testing: Integration test client	16-8
Integration test client: Events tab	16-10
Integration test client: Configurations tab	16-12
Component testing	16-14
Component test project	16-16
Test suite: Overview tab	16-18
Test suite: Test Cases tab	16-19
Test suite: Configurations tab	16-21
Running a test case	16-23
Cross-component trace (1 of 2)	16-24
Cross-component trace (2 of 2)	16-26
Fine-grained trace	16-28
Command-line test invocation	16-29
Stand-alone data pool	16-30
Component Test Explorer (1 of 2)	16-31
Component Test Explorer (2 of 2)	16-32
Server Logs view	16-33
The IBM Guided Activity Assistant	16-35
16.2. Deploying integration applications with serviceDeploy	16-37
Deploying integration applications with serviceDeploy	16-38
Overview of serviceDeploy	16-39
serviceDeploy: Single developer	16-42
Multiple developers and serviceDeploy	16-43
Unit summary	16-44
Checkpoint questions	16-45
Checkpoint answers	16-46
Exercise 14	16-47
Exercise objectives	16-48
Course business scenario: What you work with in Exercise 14	16-49
Components that are required for Exercise 14	16-50
Unit 17. IBM Process Center	17-1
Unit objectives	17-2
Topics	17-3
17.1. Introduction to IBM Process Center	17-5
Introduction to IBM Process Center	17-6
IBM Business Process Manager	17-7
IBM Process Center capabilities	17-8
17.2. Components of IBM Process Center	17-9
Components of IBM Process Center	17-10
IBM Process Center Console	17-11

Accessing IBM Process Center	17-12
IBM Process Center: Process Apps	17-13
IBM Process Center: Toolkits	17-14
IBM Process Center: Snapshots	17-15
IBM Process Center: Servers	17-16
IBM Process Center: Admin	17-17
17.3. Manage, test, and deploy IBM BPM applications	17-19
Manage, test, and deploy IBM BPM applications	17-20
Importing from the IBM Process Center repository	17-21
Associating a module or library	17-22
Publish and synchronize	17-24
IBM Process Designer artifacts in IBM Integration Designer	17-25
IBM Integration Designer artifacts in IBM Process Designer	17-26
Manage	17-27
Cleanup snapshots	17-28
Deploy	17-29
Typical development and deployment scenario	17-30
Unit summary	17-31
Checkpoint questions	17-32
Checkpoint answers	17-33
Exercise 15	17-34
Exercise objectives	17-35
Course business scenario: What you work with in Exercise 15 (1 of 2)	17-36
Course business scenario: What you work with in Exercise 15 (2 of 2)	17-37
Components that are required for Exercise 15	17-38
Exercise 16	17-39
Exercise objectives	17-40
Course business scenario: What you work with in Exercise 16	17-41
Components that are required for Exercise 16	17-42
Unit 18. Advanced Integration services	18-1
Unit objectives	18-2
Topics	18-3
18.1. AIS introduction	18-5
AIS introduction	18-6
What is an Advanced Integration service?	18-7
Business value of an Advanced Integration service	18-8
18.2. Approaches in creation of AIS	18-9
Approaches in creation of AIS	18-10
Two approaches for working with AIS	18-11
Top-down approach	18-12
Step 1: Create process application in Process Designer	18-13
Step 2: Define inputs and outputs in AIS	18-14
Step 3: Implement the AIS	18-15
Step 4: Refresh and Publish changes to Process Center	18-16
Bottom-up approach	18-17
Step 1: Create export in Integration Designer	18-18
Bottom-up approach	18-19
Unit summary	18-20
Checkpoint questions	18-21
Checkpoint answers	18-22

Exercise 17	18-23
Exercise objectives	18-24
Course business scenario: What you work with in Exercise 17 (1 of 2)	18-25
Course business scenario: What you work with in Exercise 17 (2 of 2)	18-26
Unit 19. Course summary	19-1
Unit objectives	19-2
Course learning objectives (1 of 3)	19-3
Course learning objectives (2 of 3)	19-4
Course learning objectives (3 of 3)	19-5
Earn an IBM Badge	19-6
To learn more on the subject	19-7
Unit summary	19-8
Appendix A. IBM BPM on Cloud	A-1
Appendix B. Supplemental information about mediations and mediation primitives	B-1
Appendix C. Human tasks and human task clients	C-1
C.1. Introduction to human tasks	C-3
C.2. Clients for human tasks	C-15
Appendix D. Human task roles, people queries, and escalations	D-1
D.1. Human task assignment	D-3
D.2. Escalation and notification	D-21
Appendix E. List of abbreviations	E-1
Appendix F. Resource guide	F-1

Trademarks

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

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

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

AIX®	CICS®	ClearCase®
DataPower®	DB™	DB2®
developerWorks®	Express®	IMS™
Initiate®	Lombardi®	Notes®
PartnerWorld®	Power®	Rational Team Concert™
Rational®	Redbooks®	RequisitePro®
Sametime®	WebSphere®	z/OS®

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

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

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

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

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

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.

For BPM on Cloud students

This course is designed for business analysts who work with IBM Business Process Manager Advanced.

This course was created with IBM Business Process Manager Advanced V8.5.7, but the tools and concepts that are covered in this class also apply to IBM BPM on Cloud. Several of the tools that you work with in this course (Process Center, Process Portal, and Process Server) are also available in ODM on Cloud. You download Process Designer and Integration Designer locally through a link that is provided in BPM on Cloud.

Relevant differences between the on-premises version of IBM BPM and BPM on Cloud are noted throughout the course.

Course description

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

Duration: 5 days

Purpose

This course teaches you how to use the tools in IBM Business Process Manager Advanced V8.5.7 to build and deploy process integration solutions. IBM Process Server supports a service-oriented architecture (SOA) by providing a platform for using diverse technologies to integrate business applications. In this course, you learn how IBM Process Server enables business integration applications to define business logic and processes based on Web Services Business Process Execution Language (WS-BPEL) and business rules. In the intensive hands-on lab exercises, you design, develop, and test a comprehensive business integration solution. You use the IBM Integration Designer tool to create business integration solutions by using the Service Component Architecture (SCA) programming model, the Service Data Objects (SDO) data model, and the mediation flow programming model. The lab environment for this course uses the Windows Server 2012 R2 64-bit platform.

Audience

This course is designed for system administrators, integration specialists, application developers, business process developers, support engineers, and technical sales and marketing professionals.

Prerequisites

Before taking this course, you should have:

- Basic Java and Java Platform, Enterprise Edition (Java EE) skills
- Basic Extensible Markup Language (XML) skills

Successful completion of course WB819, is highly advisable, but not required.

Objectives

After completing this course, you should be able to:

- Describe the purpose and business value of a service-oriented architecture

- Describe the purpose and business value of the tools included in IBM Business Process Manager Advanced V8.5.7: IBM Process Designer, IBM Integration Designer, IBM Process Server, IBM Process Center, IBM Process Portal, and Business Space
- Identify and describe the features that are available in IBM Process Designer, IBM Integration Designer, and the Service Component Architecture
- Describe the structure of modules and libraries
- Create business objects, shared interfaces, and business calendars
- Model a complex business process diagram in IBM Process Designer
- Create a complex business process that includes basic and structured WS-BPEL activities in IBM Integration Designer
- Use SCA bindings to communicate with web services
- Use WebSphere Java EE Connector Architecture (JCA) adapters in applications
- Implement data maps to transform business data
- Implement mediation modules to route and transform messages
- Implement business rules and use the Business Rules Manager web client to interact with business rules at run time
- Create a business space by using widgets and templates that are available for IBM Business Process Manager Advanced V8.5.7
- Create and run component test projects in IBM Integration Designer
- Bring the UTE Process Server online and connect with the Process Center
- Explore the IBM Process Center repository
- Create and implement an Advanced Integration service (AIS) in IBM Process Designer and IBM Integration Designer

Agenda

Day 1

- Course introduction
- Unit 1. Introduction to IBM Business Process Manager V8.5.7
- Unit 2. Overview of IBM Business Process Manager Advanced V8.5.7
- Exercise 1. Exploring IBM Process Designer and IBM Process Portal
- Unit 3. Course business scenario: What you build
- Unit 4. IBM Integration Designer overview
- Exercise 2. Exploring IBM Integration Designer, part I

Day 2

- Unit 5. Service Component Architecture overview
- Exercise 3. Exploring IBM Integration Designer, part II
- Unit 6. SCA bindings
- Exercise 4. Working with web services
- Unit 7. Business objects
- Exercise 5. Creating business objects and shared interfaces
- Unit 8. Business process choreography overview
- Exercise 6. Creating a business process, part I

Day 3

- Unit 9. Business process basic and structured activities
- Exercise 7. Creating a business process, part II
- Unit 10. Business process handlers, runtime behavior, and clients
- Exercise 8. Creating a business process, part III
- Unit 11. Business rules
- Exercise 9. Creating business rules
- Unit 12. Adapters
- Exercise 10. Implementing WebSphere (JCA) adapters

Day 4

- Unit 13. Developing mediation services
- Exercise 11. Creating mediation services, part I
- Unit 14. Mediation primitives
- Exercise 12. Creating mediation services, part II
- Unit 15. Business Space
- Exercise 13. Exploring Business Space

Day 5

- Unit 16. Advanced testing
- Exercise 14. Using component tests
- Unit 17. IBM Process Center
- Exercise 15. Bringing the UTE Process Server online
- Exercise 16. Exploring IBM Process Center
- Unit 18. Advanced Integration services
- Exercise 17. Implementing Advanced Integration services
- Unit 19. Course summary

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

What this unit is about

In this unit, you learn about the purpose, function, and business value of SOA, and apply SOA principles to business process management (BPM). The unit introduces the tools that are included with IBM Business Process Manager V8.5.7 Advanced edition.

What you should be able to do

After completing this unit, you should be able to:

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

How you will check your progress

- Checkpoint questions



The image shows the WebSphere Education logo on the left and the IBM logo on the right, both set against a blue decorative background.

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

[Comments \(0\)](#) [Versions \(1\)](#) [Attachments \(1\)](#) [About](#)

© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

Unit objectives

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

© Copyright IBM Corporation 2016

Figure 1-2. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to business process management
- IBM Business Process Manager

© Copyright IBM Corporation 2016

Figure 1-3. Topics

WB8601.2

Notes:

1.1. Introduction to business process management

Introduction to business process management



© Copyright IBM Corporation 2016

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

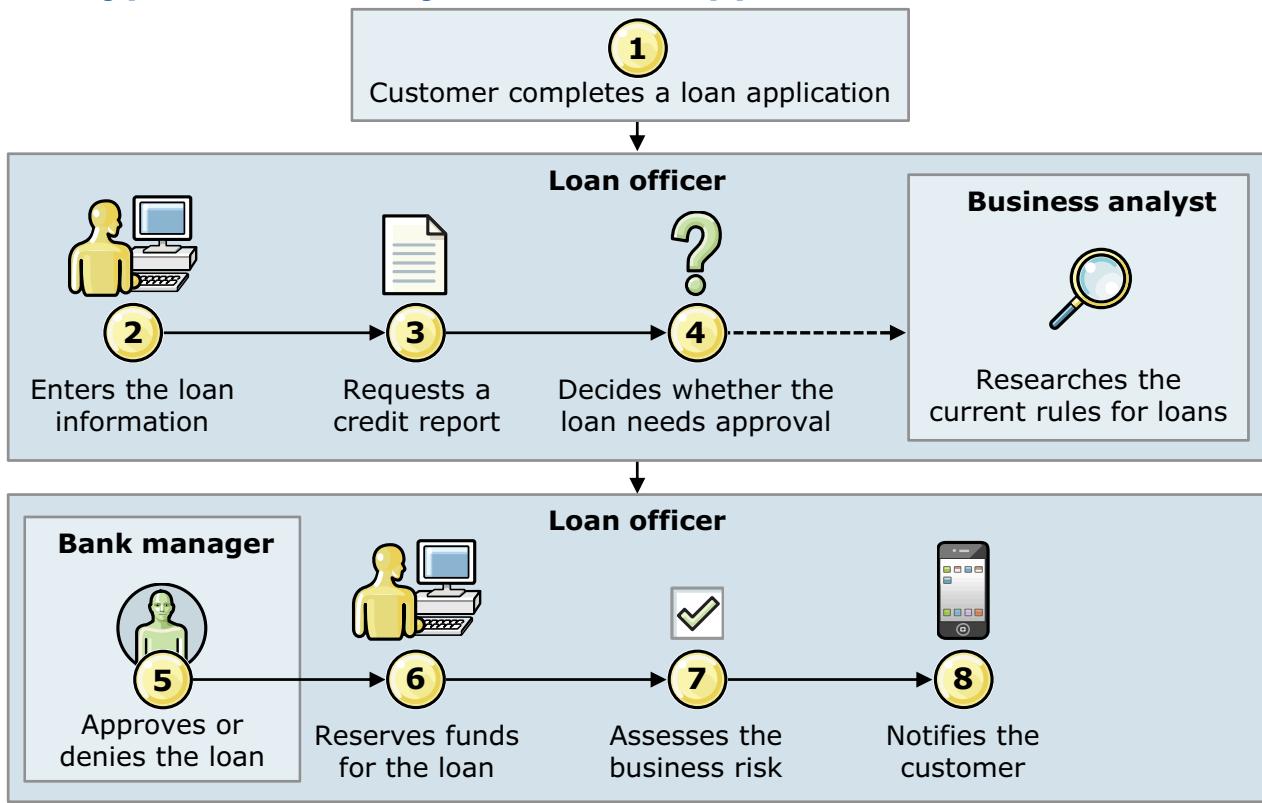
10.1

Figure 1-4. Introduction to business process management

WB8601.2

Notes:

A typical manual system: Loan application



© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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.



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 2016

Figure 1-6. Challenges with the old system

WB8601.2

Notes:

Loan application as an automated business process

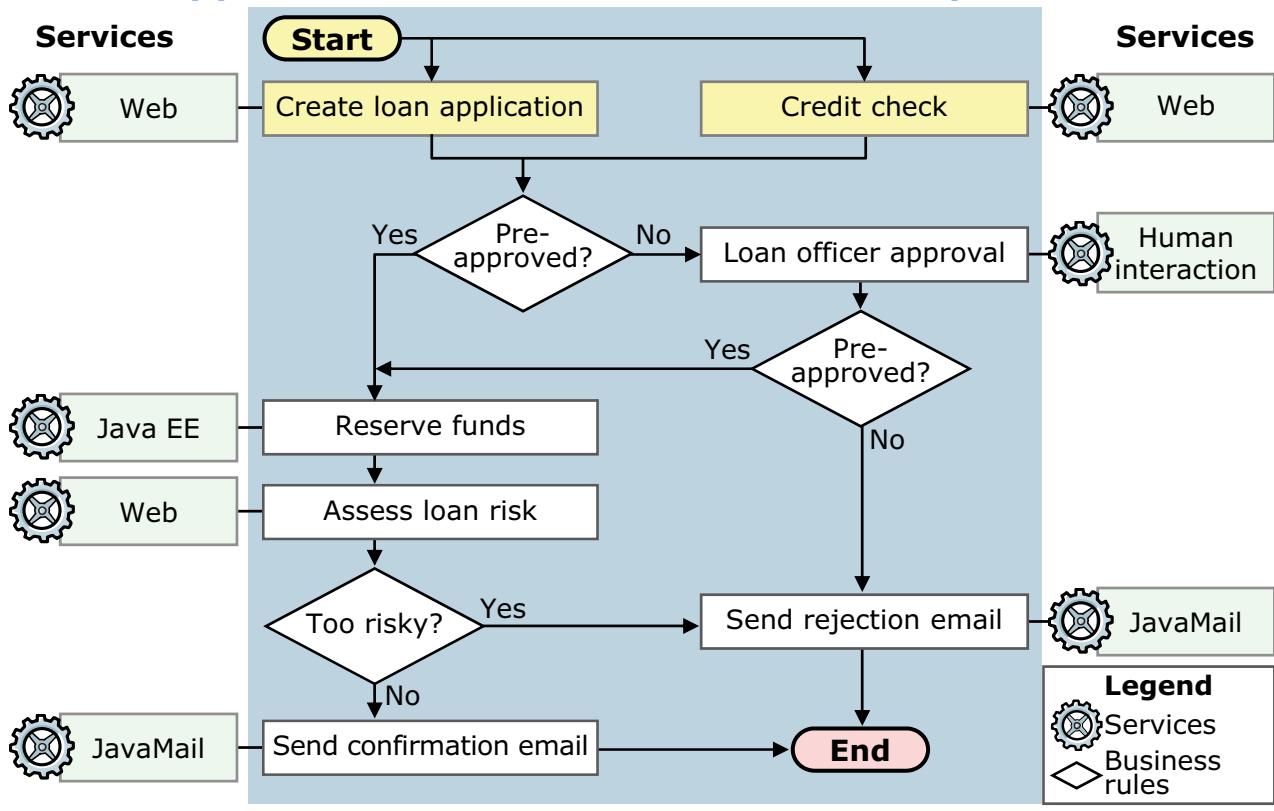


Figure 1-7. Loan application as an automated business process

WB8601.2

Notes:

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

© Copyright IBM Corporation 2016

Figure 1-8. Services and service-oriented architecture

WB8601.2

Notes:

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.

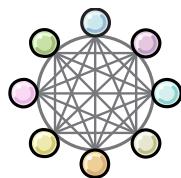
One way to provide an existing application as a service is to create an SOA gateway that converts the interfaces of the existing application into SOA-compliant interfaces. Creating this gateway involves no disruption to the application coding, so regression testing of the application is not necessary. 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.

Evolution of enterprise integration toward SOA

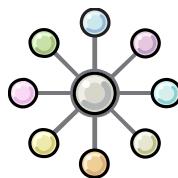
- Architectures enable flexibility and reuse
- Integrate applications by using **loosely coupled** and **well-defined** interactions
 - Hide the implementation details from the service-invoking application
 - Provide a common service definition independent of a specific technology

Point-to-point connectivity



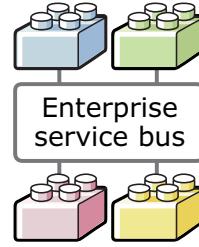
- Custom point-to-point integration between technology-aware bridges and application components
- **Point-to-point systems are brittle**

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**

© Copyright IBM Corporation 2016

Figure 1-9. Evolution of enterprise integration toward SOA

WB8601.2

Notes:

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

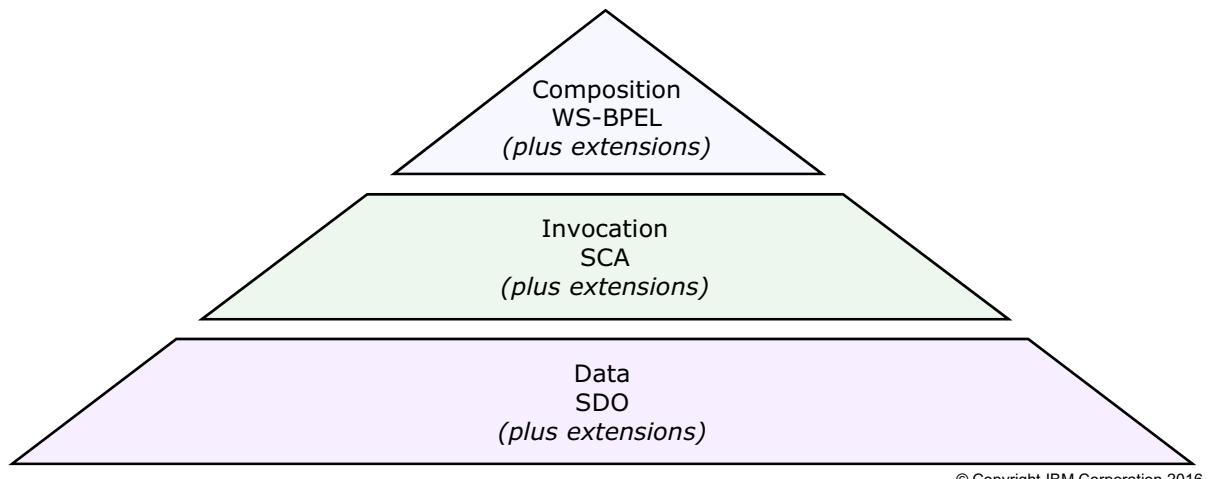
components are typically tightly coupled. Initial costs are higher than point-to-point, but maintenance efforts are much smaller.

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.

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 2016

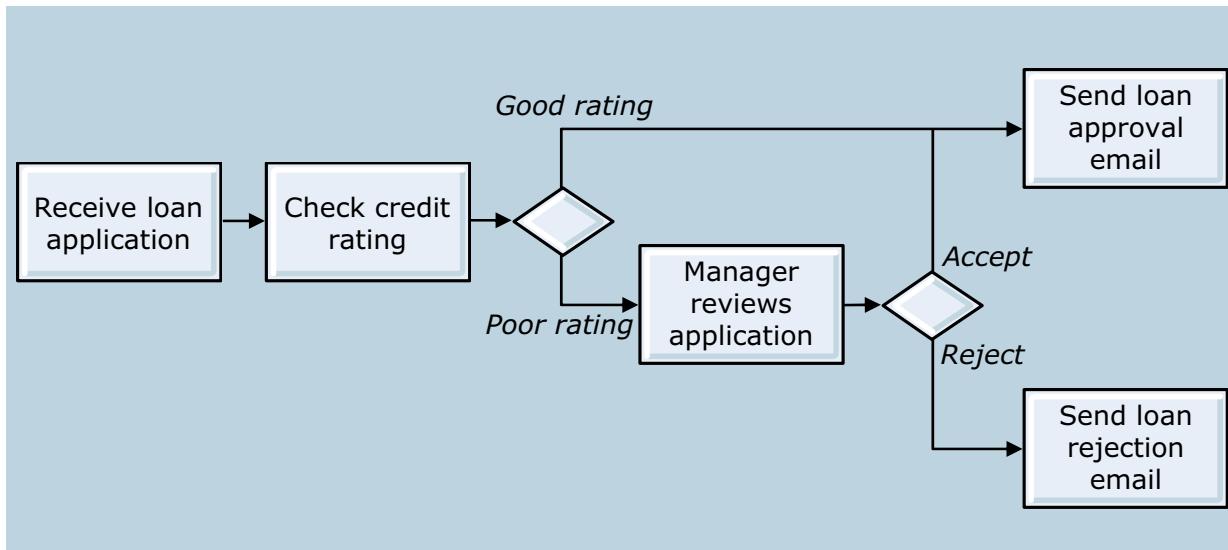
Figure 1-10. SOA model: Open standards with IBM extensions

WB8601.2

Notes:

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



© Copyright IBM Corporation 2016

Figure 1-11. Business processes support business goals

WB8601.2

Notes:

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.

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 2016

Figure 1-12. Business Process Management

WB8601.2

Notes:



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 2016

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

WB8601.2

Notes:

1.2. IBM Business Process Manager



IBM Business Process Manager



© Copyright IBM Corporation 2016

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

10.1

Figure 1-14. IBM Business Process Manager

WB8601.2

Notes:

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

© Copyright IBM Corporation 2016

Figure 1-15. IBM Business Process Manager

WB8601.2

Notes:

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

Business integration roles in IBM BPM development

User role	Role definition
Integration developer	<ul style="list-style-type: none"> Some basic programming experience (loops, conditionals, and string manipulation) Expects tools to simplify and abstract advanced IT implementation details
Application developer (IT developer)	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Focus is on administration, management, and maintenance of a deployed business solution

© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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.



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

© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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.



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

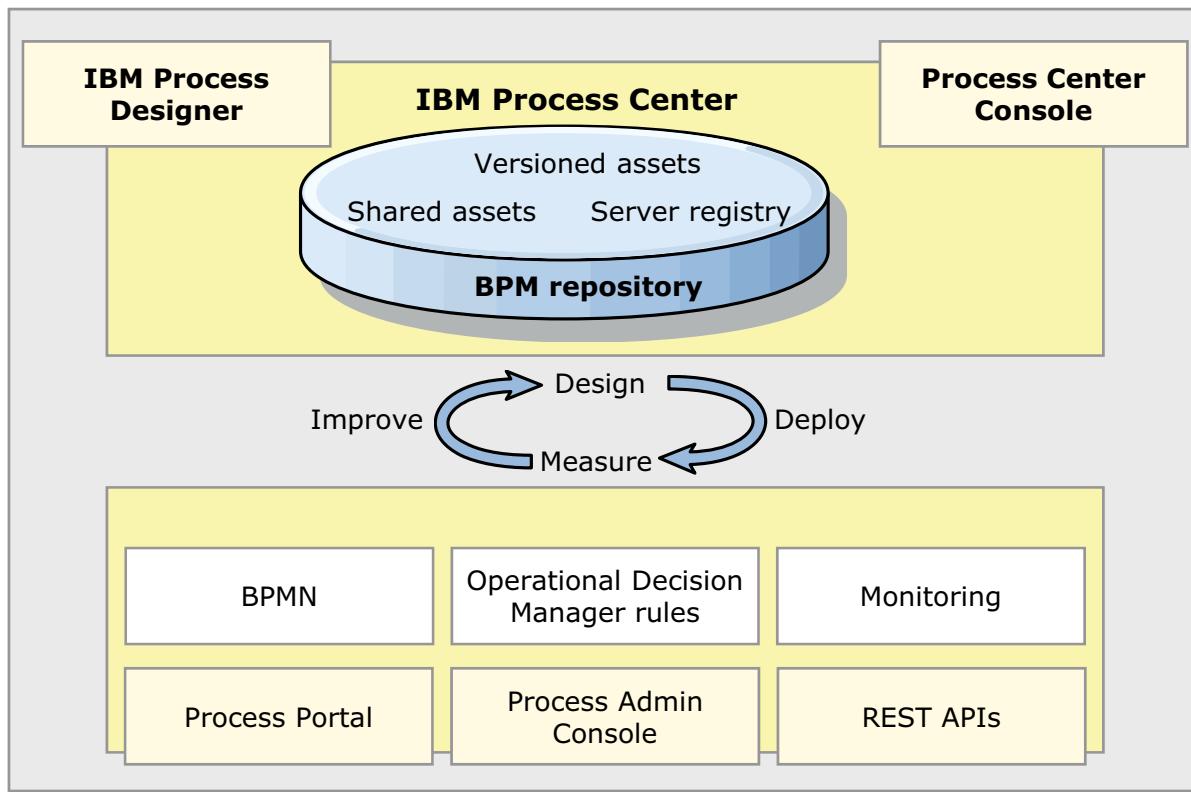
© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

IBM Business Process Manager Standard (2 of 2)



© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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

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).



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

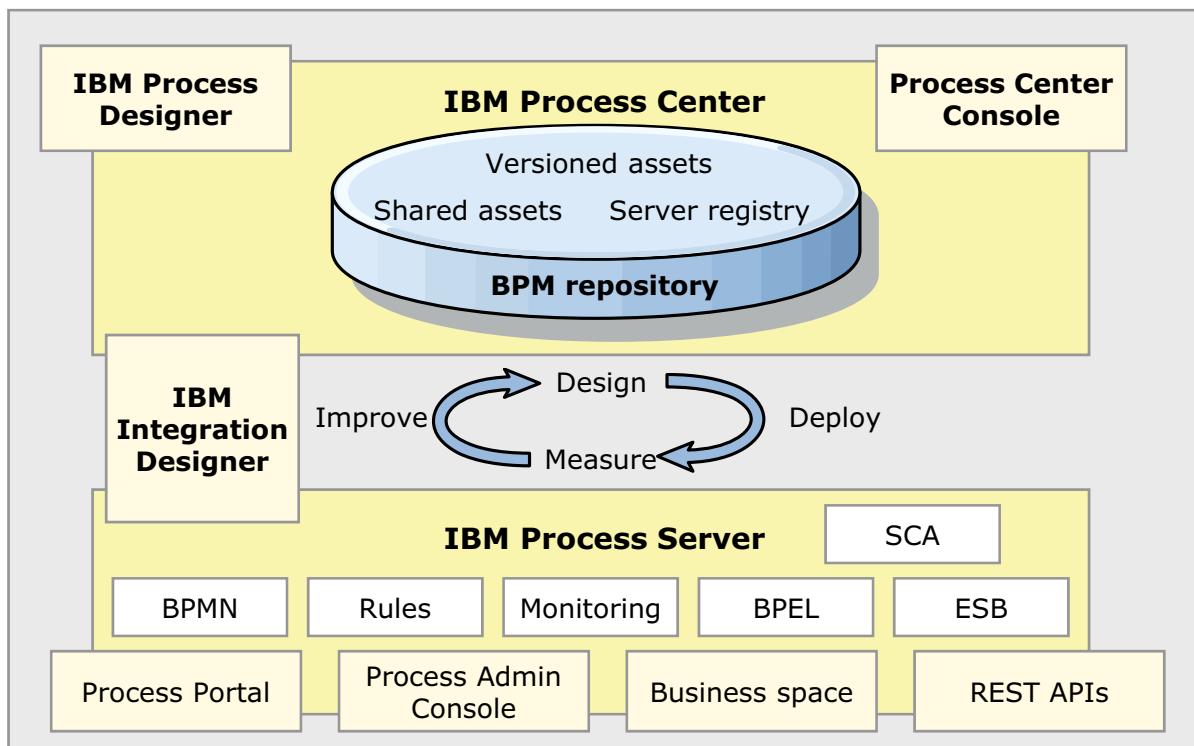
© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

IBM Business Process Manager Advanced (2 of 2)



© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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).

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 2016

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

WB8601.2

Notes:

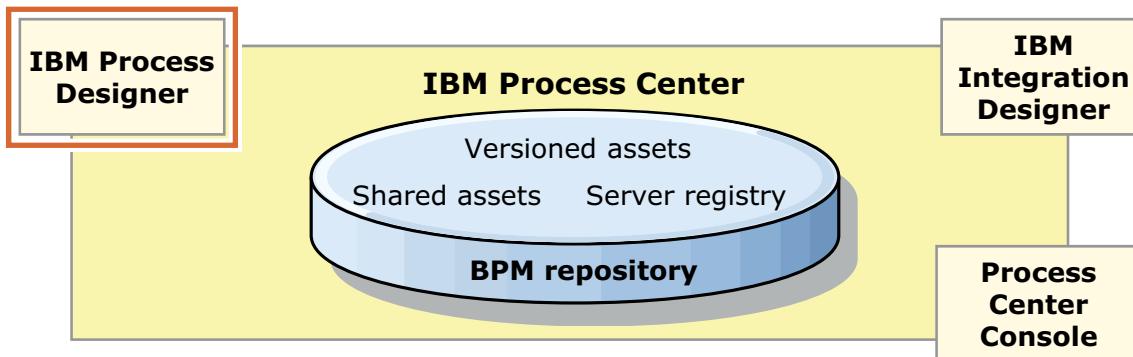
IBM Process Designer is an authoring environment that is used to create process models that contain automated and human tasks that are developed with the Business Process Model and Notation (BPMN) and Business Process Definition (BPD) formats.

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.

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)



© Copyright IBM Corporation 2016

Figure 1-23. IBM Process Designer

WB8601.2

Notes:

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; sequence lines, rules, and variables. When you model a process, you create a reusable business process definition (BPD).

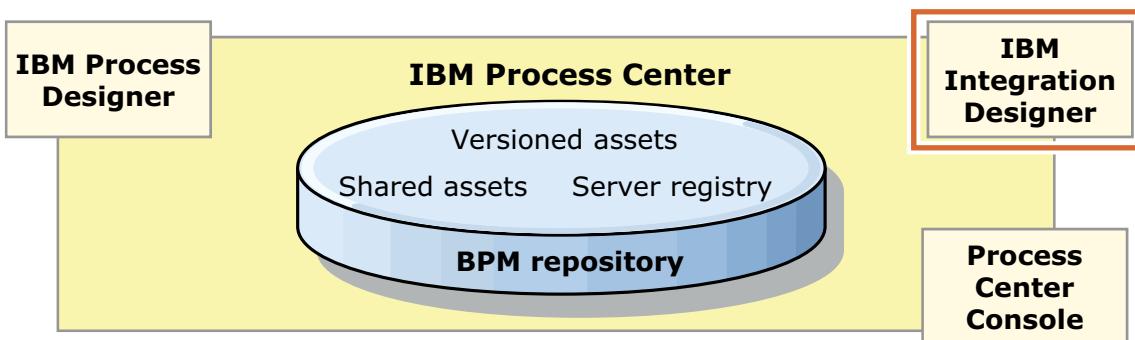
All Process Designer projects are contained in process applications. You store those process applications and associated artifacts in the Process Center repository.

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.

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)



© Copyright IBM Corporation 2016

Figure 1-24. IBM Integration Designer

WB8601.2

Notes:

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:

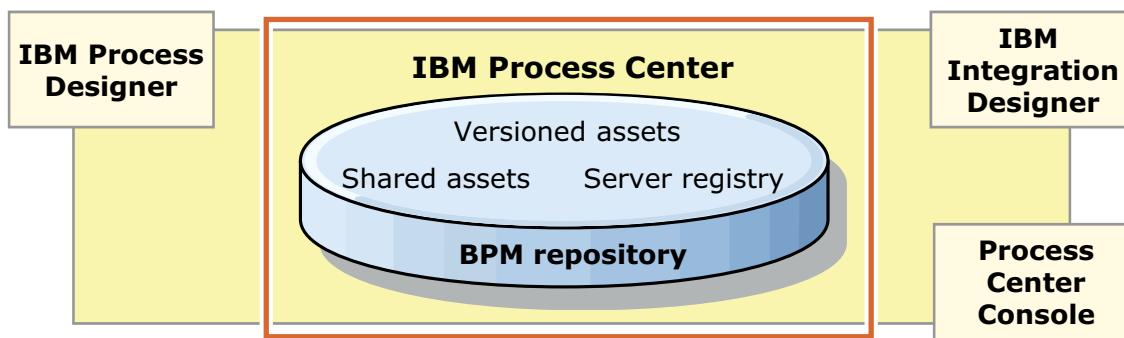
- SCA components that contain one or more modules and libraries and are deployable as EAR files
- Advanced Integration services that are traditional BPEL processes

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.

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



© Copyright IBM Corporation 2016

Figure 1-25. IBM Process Center

WB8601.2

Notes:

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.

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

Work 1 Total Open

Enter search text...

Task: Submit requisition
Open new position requested by (254)
Due: May 25, 2016 9:06 PM

Task completion user interface

Details Experts Stream

Experienced Users 0

No one in the system has completed a similar task. For more assistance, contact the system administrator.

Subject Matter Experts 1

Maria
Similar Tasks Completed: 0
Average Completion Time: 0 hours
Mention

© Copyright IBM Corporation 2016

Figure 1-26. Process Portal

WB8601.2

Notes:

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.

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**.

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 2016

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

WB8601.2

Notes:

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:

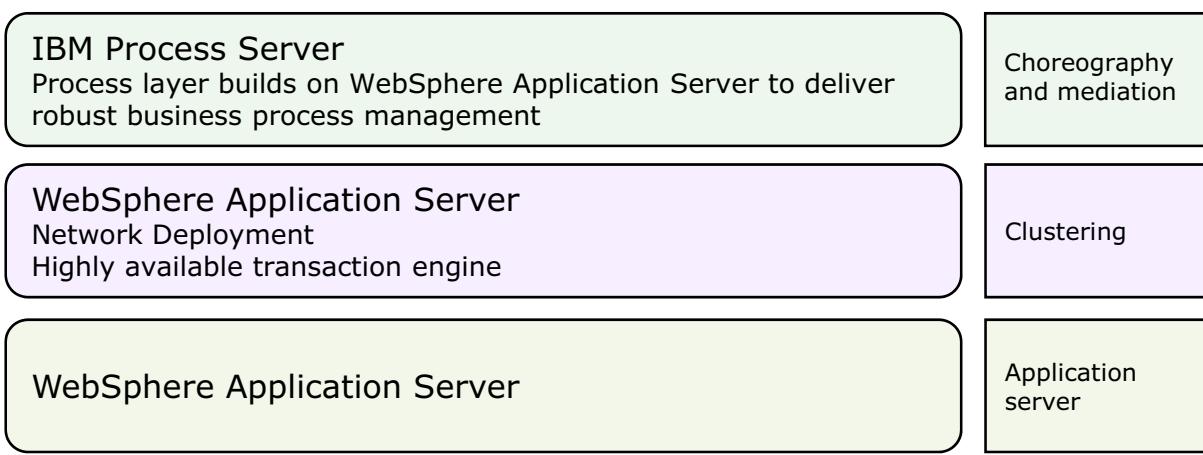
- Middleware
- An environment
- A group of functions and capabilities that make your life dramatically easier when your challenge or role is the development of SOA-based solutions

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.



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 2016

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

WB8601.2

Notes:

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.

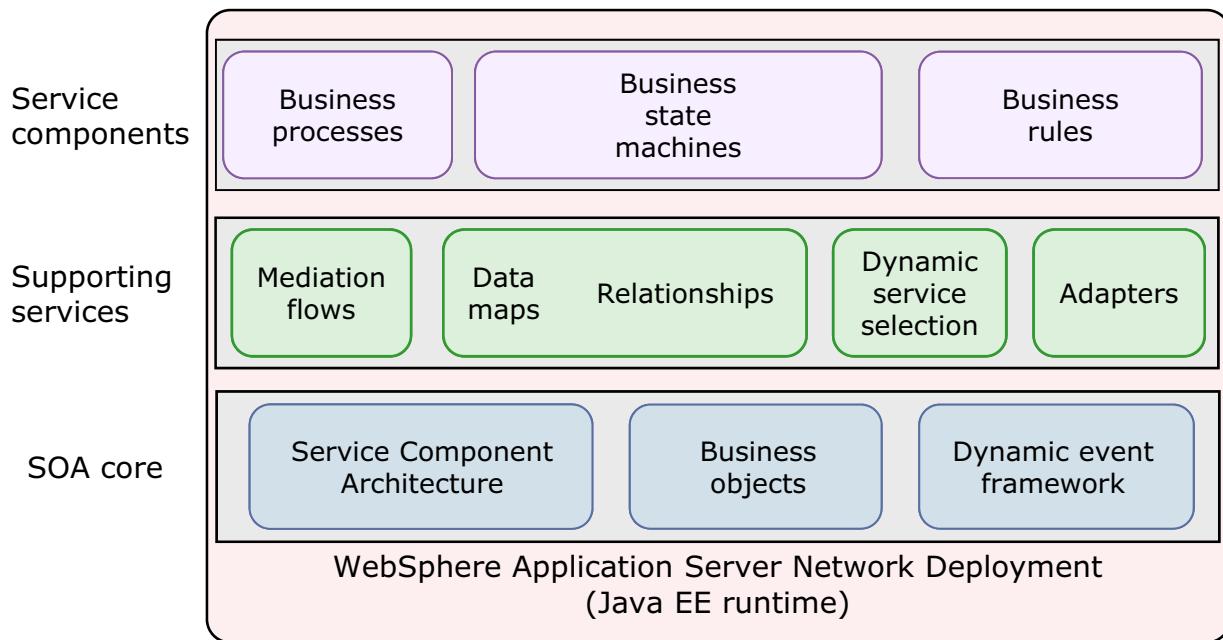
Although IBM Process Server is implemented on top of WebSphere Application Server, Java EE skills are not required for designing and implementing BPM solutions in IBM Process Server. IBM Process Server provides a higher level of abstraction, hiding its own implementation details.

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.

IBM Process Server components

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



© Copyright IBM Corporation 2016

Figure 1-29. IBM Process Server components

WB8601.2

Notes:

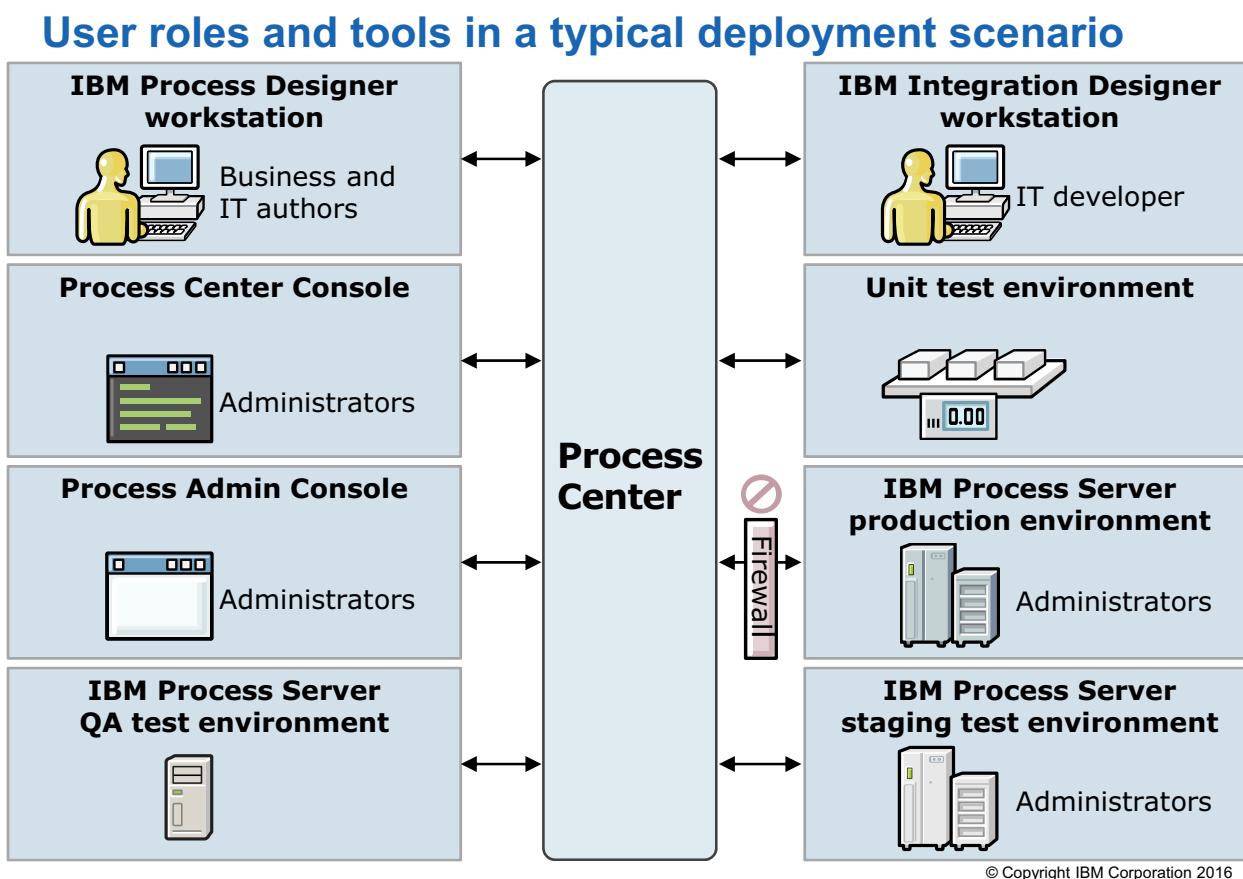


Figure 1-30. User roles and tools in a typical deployment scenario

WB8601.2

Notes:

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.



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)

© Copyright IBM Corporation 2016

Figure 1-31. Introduction to IBM BPM on Cloud

WB8601.2

Notes:

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

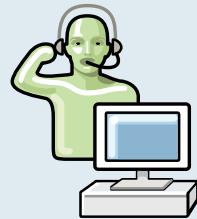
IBM Business Process Manager on Cloud provides a full lifecycle IBM BPM environment that includes development, test, and production – with tools and runtime for process design, execution, monitoring, and optimization. It is designed to enable business users to get started with process improvement quickly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

For more information on IBM BPM on Cloud, see Appendix A: “IBM BPM on Cloud.”

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



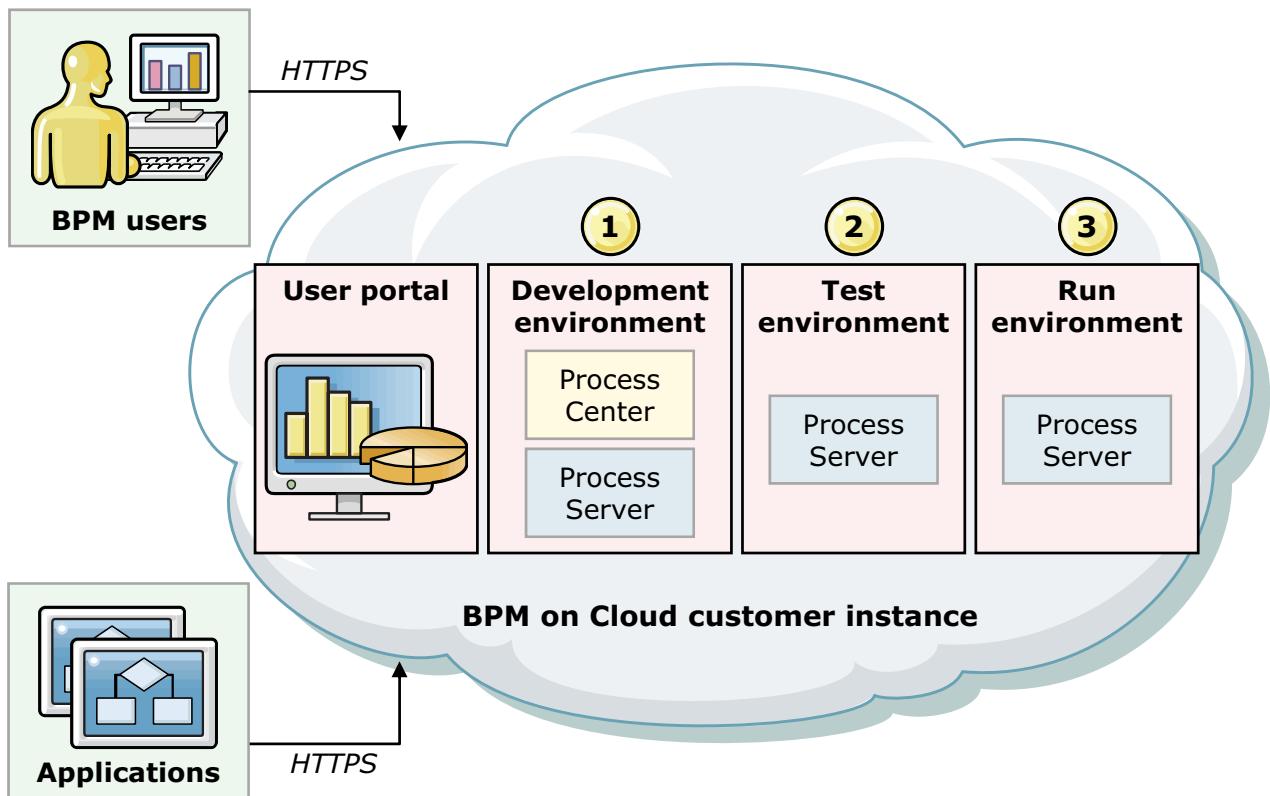
© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

IBM BPM on Cloud: Three runtime environments



© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

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

- Development
- Test
- Run

In this diagram:

- **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.



IBM BPM on Cloud user portal

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

Development Environment

Process Center  Install and run process applications, store performance data, and manage running instances of process applications on the Process Center servers. Launch More info Available Downloads (2) IBM® Process Designer IBM® Integration Designer	REST UI  Prototype IBM BPM REST resources and their associated parameters Launch More info	Process Portal  Collaborate on tasks and view the performance of individuals, teams, and processes on dashboards. Launch More info	Process Admin Console  Manage the Process Center server and the process servers in your runtime environments. Launch More info
Tech Preview: Responsive Federated Portal  Technical demonstrations of Responsive Federated Portal and Responsive Coach Toolkit as a sample Launch More info	Business Process Choreographer Explorer  Monitor and manage BPEL processes. The BPC Explorer provides a number of views that show process and task metadata Launch More info	Business Rules Manager  Manage business rules Launch More info	

© Copyright IBM Corporation 2016

Figure 1-34. IBM BPM on Cloud user portal

WB8601.2

Notes:

Unit summary

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

© Copyright IBM Corporation 2016

Figure 1-35. Unit summary

WB8601.2

Notes:



Checkpoint 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 2016

Figure 1-36. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.

Checkpoint answers

1. 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. False. The integration developer role requires only basic programming experience (loops, conditionals, and string manipulation).
3. False. IBM Integration Designer is not included in IBM BPM Standard. It is included only in the IBM BPM Advanced version.

© Copyright IBM Corporation 2016

Figure 1-37. Checkpoint answers

WB8601.2

Notes:



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 2016

Figure 1-38. References

WB8601.2

Notes:

Unit 2. Overview of IBM Business Process Manager Advanced V8.5.7

What this unit is about

This unit is an overview of IBM BPM Advanced 8.5.7. You learn its key capabilities and when to use the IBM BPM Advanced product.

What you should be able to do

After completing this unit, you should be able to:

- Describe the differences between BPMN and BPEL
- Explain when to use the IBM BPM Advanced product
- Describe the capabilities of IBM Business Process Manager V8.5.7

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the differences between BPMN and BPEL
- Explain when to use the IBM BPM Advanced product
- Describe the capabilities of IBM Business Process Manager V8.5.7

© Copyright IBM Corporation 2016

Figure 2-1. Unit objectives

WB8601.2

Notes:



Topics

- BPD and BPEL
- IBM BPM Standard or IBM BPM Advanced?
- Key components in IBM BPM Advanced

© Copyright IBM Corporation 2016

Figure 2-2. Topics

WB8601.2

Notes:

2.1. BPD and BPEL

BPD and BPEL



© Copyright IBM Corporation 2016

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

10.1

Figure 2-3. BPD and BPEL

WB8601.2

Notes:

Two kinds of business processes: BPD and BPEL

- Two kinds of business processes:
 - Business process definition or BPD (sometimes also referred to as a BPMN process)
 - Business process execution language or a BPEL process
- A BPD or business process definition is a business process that uses the Business Process Model and Notation (BPMN) standards
 - You use Process Designer to create a BPD process
- A BPEL or Business Process Execution Language is a business process that is XML-based and uses the BPEL standards
 - You use Integration Designer to create a BPEL process

© Copyright IBM Corporation 2016

Figure 2-4. Two kinds of business processes: BPD and BPEL

WB8601.2

Notes:

 WebSphere Education 

BPMN

- The standard flow chart-based notation for defining business processes (BPD)
- Creates a standardized bridge for the gap between business process design and process implementation
- IBM Business Process Manager's Process Designer uses several core elements from BPMN
 - Pool
 - Lane
 - Event
 - Activity
 - Flow
 - Gateway

Process Designer element palette



© Copyright IBM Corporation 2016

Figure 2-5. BPMN

WB8601.2

Notes:

Business Process Model and Notation or BPMN is a standardized graphical notation for creating diagrams of business processes.

BPMN is used so that everyone who is involved can interpret and understand the model. Throughout development, many different parties are involved in modeling. Every stakeholder, from the least technical to the most technical, understands the model to provide valuable feedback and continuously improve the process.

BPMN also allows a way to compact your process definition. Many of the symbols represent ideas, so symbols allow for a more concise and smaller model than drawing a diagram without BPMN.

Many benefits of BPMN exist, but most importantly, BPMN creates a standardized bridge for the gap between the business process design and process implementation. This single notation is agreed upon among multiple BPM vendors for the benefit of the user community.

IBM implements and interprets these elements to have specific meanings and terminology in the IBM Process Designer product. For definitions of the BPMN specification, see the BPMN Specification document version 2.0 from the Object Management Group. You can learn more about BPMN at: <http://www.bpmn.org>

The element palette of IBM Process Designer is shown in the slide image. These items correspond to BPMN elements and are used in modeling a process.

A BPMN standards group that is called Object Management Group is an international, open membership, not-for-profit technology standards consortium. You can search the Internet for this group.



© Copyright IBM Corporation 2016

Figure 2-6. BPMN process in Process Designer (desktop interface)

WB8601.2

Notes:

IBM Process Designer is available through a web-based interface or a desktop interface.

A business process definition (BPD) is a major unit of logic in IBM Business Process Manager (BPM). You create BPDs in the IBM Process Designer desktop editor.

In IBM Business Process Manager, you create two types of processes, depending on which editor you are using. To model a process in the desktop Process Designer, you use a BPD. To model a process in the web Process Designer, you use a process. At run time, both BPDs and processes become process instances. You can create most of the artifacts in the web editor. However, you can edit some artifacts only in the desktop editor. For example, you can view and select under cover agents in the web editor, but to create new under cover agents, you must use the desktop editor.

Depending on the artifact that you are working with, you use either the Eclipse-based Process Designer on the desktop, or the web Process Designer.

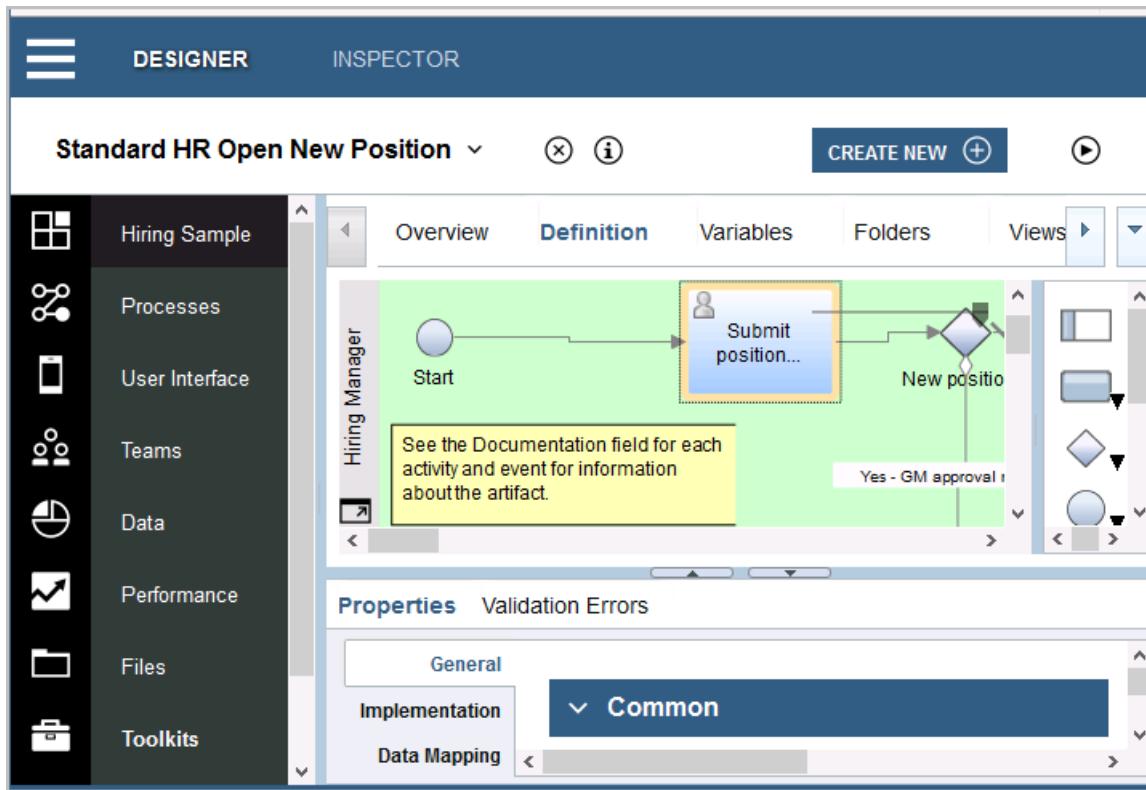
The main difference between the two editors is: you create and edit BPDs, services, and heritage user interfaces in the desktop Process Designer, and you create and edit processes and responsive user interfaces in the web Process Designer. You can edit some artifacts, such as business objects, in both.

To open a process application in the desktop editor, you must launch Process Center from the desktop Process Designer. To start the desktop Process Designer:

1. Double-click the **IBM Process Designer Windows** desktop shortcut.
2. Choose **Start > IBM > IBM Process Designer** from the Windows desktop.



BPMN process in Process Designer (web-based interface)



© Copyright IBM Corporation 2016

Figure 2-7. BPMN process in Process Designer (web-based interface)

WB8601.2

Notes:

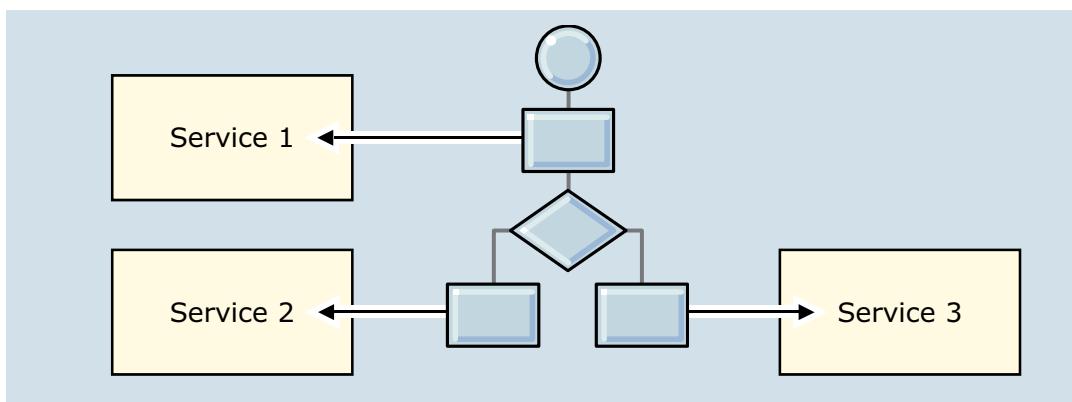
When you click the Open in Designer link in Process Center, the web editor opens by default. If you are working in the desktop Process Designer and you open an artifact that is created in the web Process Designer – for example, a process – the web Process Designer is launched. The artifact is opened in the web editor.

Multiple users can work simultaneously on the same process applications and artifacts in the two editors, and changes happen automatically and seamlessly.

Convert BPDs to processes so that you can work with them in the web Process Designer. When you convert a BPD into a process, the subprocesses inside it are also converted, and tasks and their properties are maintained. References to linked processes and heritage human services are maintained. However, the referenced artifacts are not converted. To convert your user interfaces, you must separately convert your heritage human services and coaches.

BPEL

- A BPEL process is a business flow of execution paths that are described in WS-BPEL (Web Services Business Process Execution Language), including:
 - Which services are invoked
 - In what order services are invoked
 - The movement of data between services
- BPEL facilitates the building of composite integration applications by allowing the reuse of existing IT assets that are exposed as services



© Copyright IBM Corporation 2016

Figure 2-8. BPEL

WB8601.2

Notes:

The BPEL process component of IBM Process Server is one of the most important aspects of the solution. This component provides the engine and management to run the steps for each instance of a process. Process orchestration occurs within the engine, and like the conductor of an orchestra, the process engine guides all the other components.

The diagram shows a set of services that provide their “black box” functions, in which only their externally visible behavior is considered and not their implementation or inner workings. To build a solution, you design a process that choreographs their execution. This process is described in the BPEL language. The core concepts that must be captured are: which services are used, in what order they are called, and what data is passed from one service to another.



BPEL process in IBM Integration Designer

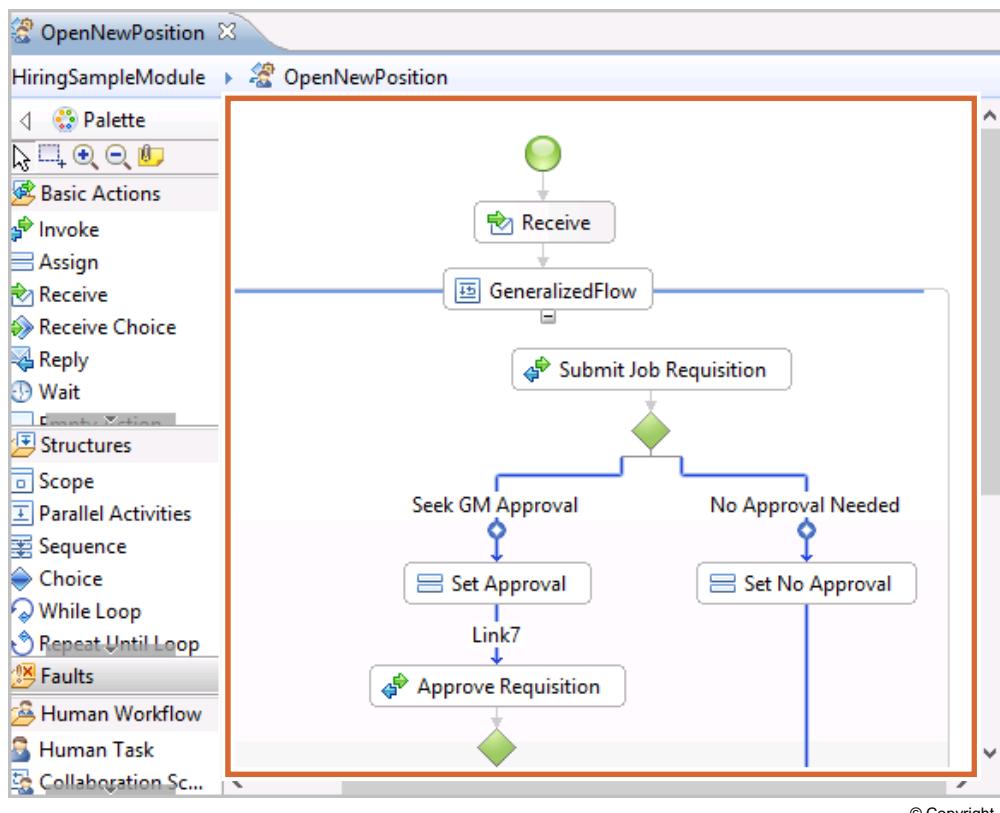


Figure 2-9. BPEL process in IBM Integration Designer

WB8601.2

Notes:

When to use BPMN?

You can use BPMN under following scenarios:

- Presence of slow-moving human processes
- Rich human interaction features are a requirement that might take advantage of the rich coach user interface
- Integration with other external complex systems is not a requirement
- Your environment has either IBM BPM Standard or IBM BPM Advanced

© Copyright IBM Corporation 2016

Figure 2-10. When to use BPMN?

WB8601.2

Notes:

When to use BPEL?

You can use BPEL under following scenarios:

- Need for short-running synchronous and long-running interruptible processes
 - The process might be a single transaction or can be run over an extended period
- Need for high-performance complex processes with error handling, compensation (rolling back), and asynchronous processing
- Your environment has IBM BPM Advanced (Integration Designer is included only with IBM BPM Advanced)

© Copyright IBM Corporation 2016

Figure 2-11. When to use BPEL?

WB8601.2

Notes:

Use BPEL in the following cases:

- **Need for short-running synchronous and long-running interruptible processes:** The process might be a single transaction, for example, when you send a request with a stock quote symbol and receive a purchase price as the value.
- **The process can also be run over an extended period:** For example, when you place an order for an item but it is not available then. The customer pays for the item, the item is delivered later when available, and the customer is notified.
- **Need for high-performance complex processes with error handling, compensation, and asynchronous processing:** An example is when an application can send a request to an enterprise information system, such as Siebel Business Applications server, to query or update an account record, represented by a Siebel business component instance. It can also receive events from the server; for example, to be notified that a customer record is updated.

Another example is where a purchase order is placed for an item and upon successful order completion and delivery, all the relevant records of data such as order entry, inventory system, and customer account are updated.

Assume, for example, that a vendor is offering a time-limited warranty, and a customer returns a defective object within that period. In such a case, the original business process is still in effect, and it resumes the moment the customer returns with the object. Now, in a different scenario, the customer wants something in return for a broken object, and the transaction was fully committed in that money was exchanged for goods. A different procedure (a refund) must take place to return the conditions to a balanced state. It is not necessarily the same state that existed before (for example, the customer pays in cash but receives a store credit in return), but nonetheless it is one that is balanced and consistent. If either the customer or the vendor is unhappy, then business compensation not successful.

2.2. IBM BPM Standard or IBM BPM Advanced?

IBM BPM Standard or IBM BPM Advanced?



© Copyright IBM Corporation 2016

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

10.1

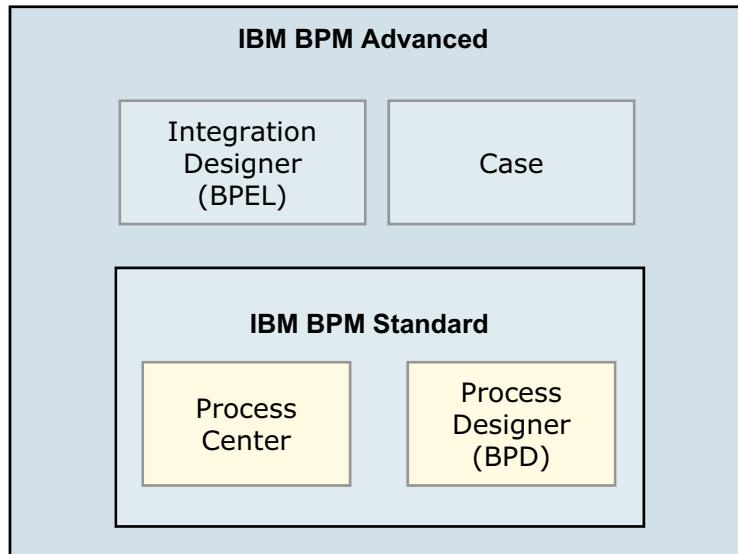
Figure 2-12. IBM BPM Standard or IBM BPM Advanced?

WB8601.2

Notes:

IBM BPM Advanced contains all of IBM BPM Standard

- All features of IBM BPM Standard are **included** in IBM BPM Advanced version
- Integration Designer and Case are available only in the Advanced version



© Copyright IBM Corporation 2016

Figure 2-13. IBM BPM Advanced contains all of IBM BPM Standard

WB8601.2

Notes:

IBM BPM Standard is a comprehensive Business Process Management Platform (BPM), providing full visibility and insight to managing business processes. It provides tools and runtime for process design, execution, monitoring, and optimization, along with basic system integration support. 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.

IBM BPM Advanced supports enterprise integration and transaction process management as part of an overall service-oriented architecture (SOA). This software includes the same features as the standard version of the business process manager (IBM BPM) offering with more advanced integration and connectivity capabilities. It combines simplicity with ease of use for task management, and extended support for high-volume automation.



When to use IBM BPM Standard?

IBM BPM Standard can be used when the following requirements exist:

- Need for high business involvement in modeling business processes, and rapid deployment of business processes
- Need to implement human-centric business processes in a highly collaborative business environment
- Ability to use business rules to drive decisions in business processes
- Ability to proactively monitor business process performance and team performance
- Need for basic system integration support
- Whenever a need for a BPMN process exists

© Copyright IBM Corporation 2016

Figure 2-14. When to use IBM BPM Standard?

WB8601.2

Notes:

When to use IBM BPM Advanced?

IBM BPM Advanced can be used when the following requirements exist

- Any of the requirements that apply to IBM BPM Standard (listed in the previous slide)
 - All IBM BPM Standard features are available in IBM BPM Advanced
- Need for basic case management capabilities in business processes
- Need for straight-through processing (STP) of business processes with no human interaction, and the ability to maintain process state between steps
- Need for integration with complex processes and systems that require complex connectivity and data mapping
- Whenever a need to use a BPEL process exists

© Copyright IBM Corporation 2016

Figure 2-15. When to use IBM BPM Advanced?

WB8601.2

Notes:

IBM BPM Advanced can be considered when the following requirements exist in addition to all or some of those requirements that apply to IBM BPM Standard:

- Need for basic case management capabilities in business processes
- Need for straight-through processing (STP) of business processes with no human interaction, and the ability to maintain process state between steps
- Need for advanced integration support with external systems

IBM BPM Advanced supports various integration bindings, including Service Component Architecture (SCA), Java Message Service (JMS), HTTP, and web services, which are instrumental in providing complex integration features:

- Need for interaction with popular external applications that have complex interfaces and arcane integration styles.
- IBM BPM Advanced provides a set of inbound and outbound adapters to ease the challenges of accessing popular application environments, such as SAP and Siebel.

- Need for domain-specific industry content packs. These packs are designed to integrate seamlessly with IBM BPM, providing a set of prebuilt assets based on industry standards. These packs help accelerate delivery of standards-based industry solutions for the banking, telecommunications, and healthcare industries, ensuring consistency and compliance across multiple lines of businesses and geographical areas.

Use IBM Process Designer or IBM Integration Designer?

- Use Process Designer when a need to create top-down human-centric processes and user interfaces exists
 - Focus is on BPMN and JavaScript skills with UI focus
- Use Integration Designer when a need for a straight-through process and integration elements of an overall solution exists
 - Focus is on BPEL, Java, and integration with other systems
- Use both Process Designer and Integration Designer when BPMN processes and BPEL processes must communicate
 - An Advanced Integration service (AIS) is a collaboration between a business user who is working in Process Designer and an integration developer who is working in Integration Designer

© Copyright IBM Corporation 2016

Figure 2-16. Use IBM Process Designer or IBM Integration Designer?

WB8601.2

Notes:




Artifacts in IBM Process Designer and IBM Integration Designer tool

	IBM Process Designer	IBM Integration Designer
Container for integration artifacts	Process App includes: <ul style="list-style-type: none"> Processes (BPD, human tasks, rules) Data and services 	Module includes: <ul style="list-style-type: none"> Integration logic (BPEL processes, human tasks, business rules) Data and interfaces Transformations
Container for shareable artifacts	Toolkit includes: <ul style="list-style-type: none"> Processes Data and services 	Library includes: <ul style="list-style-type: none"> Integration logic Data and interfaces Transformations Web Service Ports
Container for mediation services	N/A	Mediation module includes: <ul style="list-style-type: none"> Mediation flows

© Copyright IBM Corporation 2016

Figure 2-17. Artifacts in IBM Process Designer and IBM Integration Designer tool

WB8601.2

Notes:

Process applications and toolkits in IBM Process Designer are analogous to modules and libraries in IBM Integration Designer. Some similarities include:

- Process applications (like modules) are deployed to the server.
- Toolkits (like libraries) are not deployed to the server.
- Process applications contain business process modeling artifacts.
- Process applications have dependencies on any number of toolkits.
- Toolkits have dependencies on other toolkits.

Modules and libraries contain multiple SCA artifacts that are grouped by type. Libraries are projects that are used to store shared resources and are accessed by adding them to module dependencies. Integration modules provide the business services, and mediation modules provide connectivity logic. Mediation flows and business services are modeled as SCA components. SCA components are wired together in the assembly diagram to form applications.

2.3. Key components in IBM BPM Advanced

Key components in IBM BPM Advanced



© Copyright IBM Corporation 2016

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

10.1

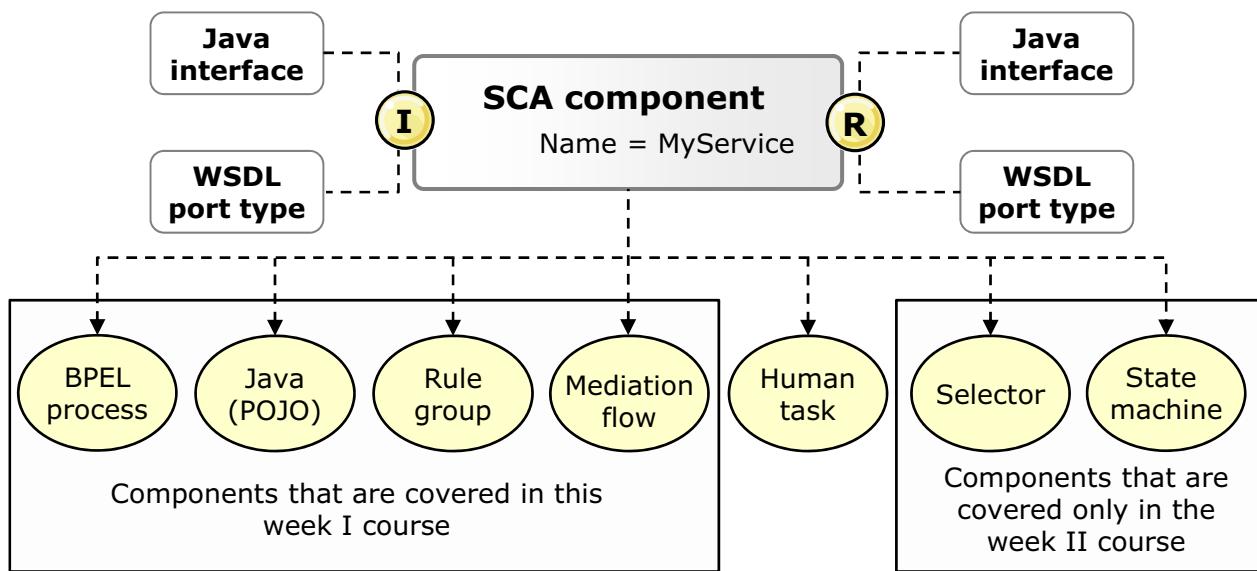
Figure 2-18. Key components in IBM BPM Advanced

WB8601.2

Notes:

Service Component Architecture (SCA)

- SCA components are discrete units of business logic that contain:
 - **Interfaces:** Are used to call the component and provide the service contract
 - **References:** Are used to call other components



© Copyright IBM Corporation 2016

Figure 2-19. Service Component Architecture (SCA)

WB8601.2

Notes:

Components that are covered in this course WB860 are:

- BPEL process
- Java
- Rule group
- Mediation flow
- Human task

More components that are covered in the second week of the IBM BPM development course are:

- Selector
- State machine

The human task component is already built in the lab exercises and is not covered in the units. While the human task components are available in IBM BPM Advanced edition, you are not encouraged to create new human task components for your IBM BPM solution. Coach user interface is the right solution. You learn coaches in the IBM BPM Standard course.



Business value of Service Component Architecture (SCA)

- SCA provides a single service component abstraction for services that might already be implemented as business processes, Java classes, and mediation flows
 - Abstraction separates “business logic” from “infrastructure logic,” allowing developers to focus on business problems, not infrastructure code
 - You use IBM Integration Designer, which is included in IBM BPM Advanced, to assemble and test SCA components
- Without SCA:
 - You must write code to communicate with the services in your application
 - The application is not loosely coupled, and is not easy to change
 - You must change application code to respond to infrastructure changes

© Copyright IBM Corporation 2016

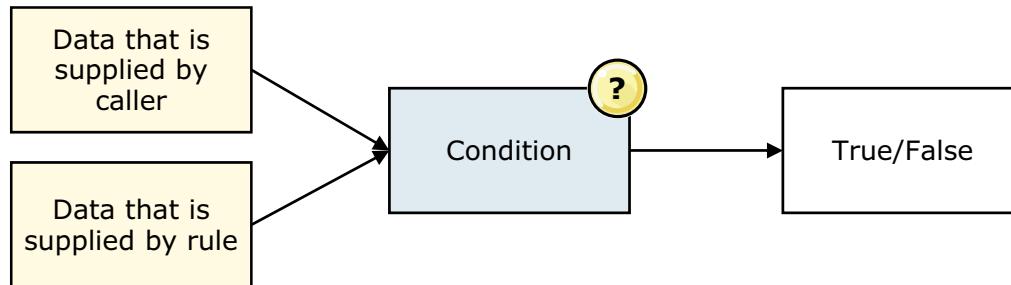
Figure 2-20. Business value of Service Component Architecture (SCA)

WB8601.2

Notes:

Business rules

- A business rule captures and implements business policies and practices by using one or more if-then statements
 - For example: If $orderTotal \geq 1000$, then the discount = 0.10
- A business rule consists of a condition (an expression that uses data that the caller and the rule supply) and one or more actions
 - The condition is the “if” portion of the statement
 - Evaluation of the condition is either true or false
 - The action is the “then” portion of the statement



© Copyright IBM Corporation 2016

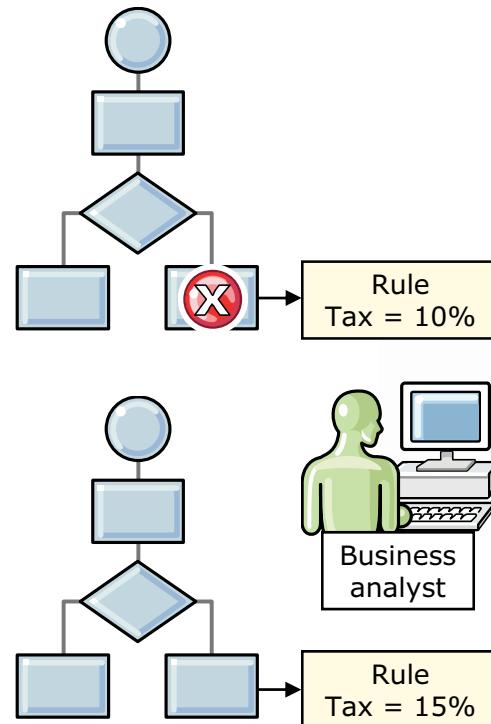
Figure 2-21. Business rules

WB8601.2

Notes:

Business value of rules

- By using rule groups to expose rules as services, rules are separated from processes that use them
- The business analyst can quickly change the rules that are exposed at run time, providing business agility and responsiveness
 - If rules are not in application code, you are no longer bound to IT development cycles
 - The developer is needed only for more complex changes
- Rule groups are SCA components
 - As an SCA implementation type, it abstracts and decouples the rule implementation



© Copyright IBM Corporation 2016

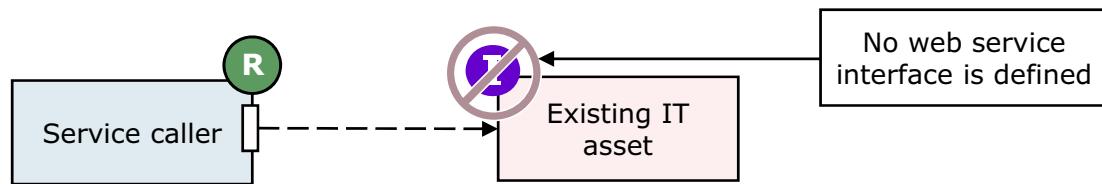
Figure 2-22. Business value of rules

WB8601.2

Notes:

Adapters

- Adapters provide integration with enterprise information systems (EIS) without service interfaces
- An EIS provides the information infrastructure for an enterprise by providing a set of services to clients:
 - Enterprise resource planning (ERP)
 - Customer relationship management (CRM)
 - Human resource systems (HR)



© Copyright IBM Corporation 2016

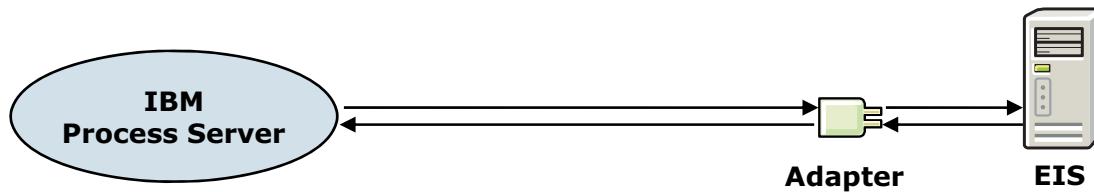
Figure 2-23. Adapters

WB8601.2

Notes:

Business value of adapters

- Adapters connect EIS systems and applications to a central server
 - Adapter is a mediator between an asset and a broker (IBM Process Server)
- Adapters provide a layer of abstraction to simplify connectivity
- Adapters use existing IT assets in your environment without significant programming, which simplifies integration
 - Adapters are presented as SCA components
 - EIS is invoked as a service through the adapter



© Copyright IBM Corporation 2016

Figure 2-24. Business value of adapters

WB8601.2

Notes:



Mediations

- Mediation module
 - Special type of SCA module
 - Mediate messages that flow between service requesters and providers
- Mediation flow component
 - Contains the mediation flow logic
 - Unique flow logic for every interface operation
 - Modules can contain zero or multiple mediation flow components
- Mediation primitives
 - Used to construct the logic of a mediation flow
 - Each primitive does a specific part of the flow logic
 - An encapsulated unit of logic that manipulates the message as it passes through the enterprise service bus
- Enterprise service bus
 - An ESB supplies a communication layer to support service interactions

© Copyright IBM Corporation 2016

Figure 2-25. Mediations

WB8601.2

Notes:

Mediation modules are a special type of SCA modules that can change the format, content, or target of service requests. They use SCA exports and imports to communicate with service requesters and service providers, which provide the key to handling protocol conversions within the bus.

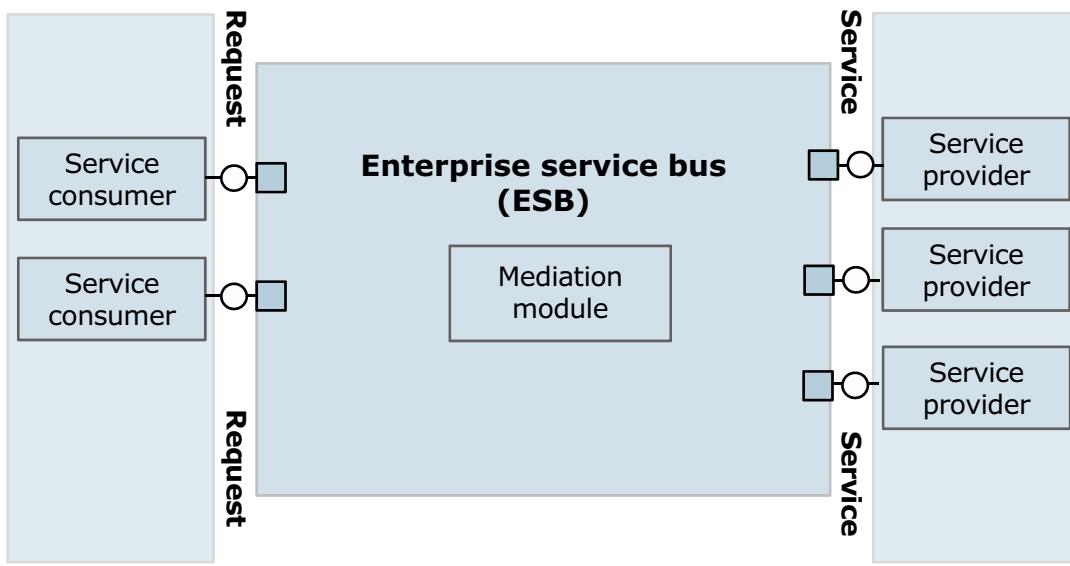
The mediation module also contains a mediation flow component. The mediation flow component contains the logic for the mediation. For every operation defined on an input interface, a unique mediation flow logic is defined for the request and response of the operation. The mediation flow logic transforms and dynamically routes messages.

The flow logic is defined in mediation flow components by using mediation primitives. Each mediation primitive provides some specific portion of the logic and is wired to other primitives into a logical flow. A service message object represents the data that mediation primitives manipulate.

ESB provides a flexible connectivity infrastructure for integrating applications and services.

Business value of mediations

- Mediation provides functions for
 - Dynamically routing messages
 - Converting transport protocols
 - Transforming message formats



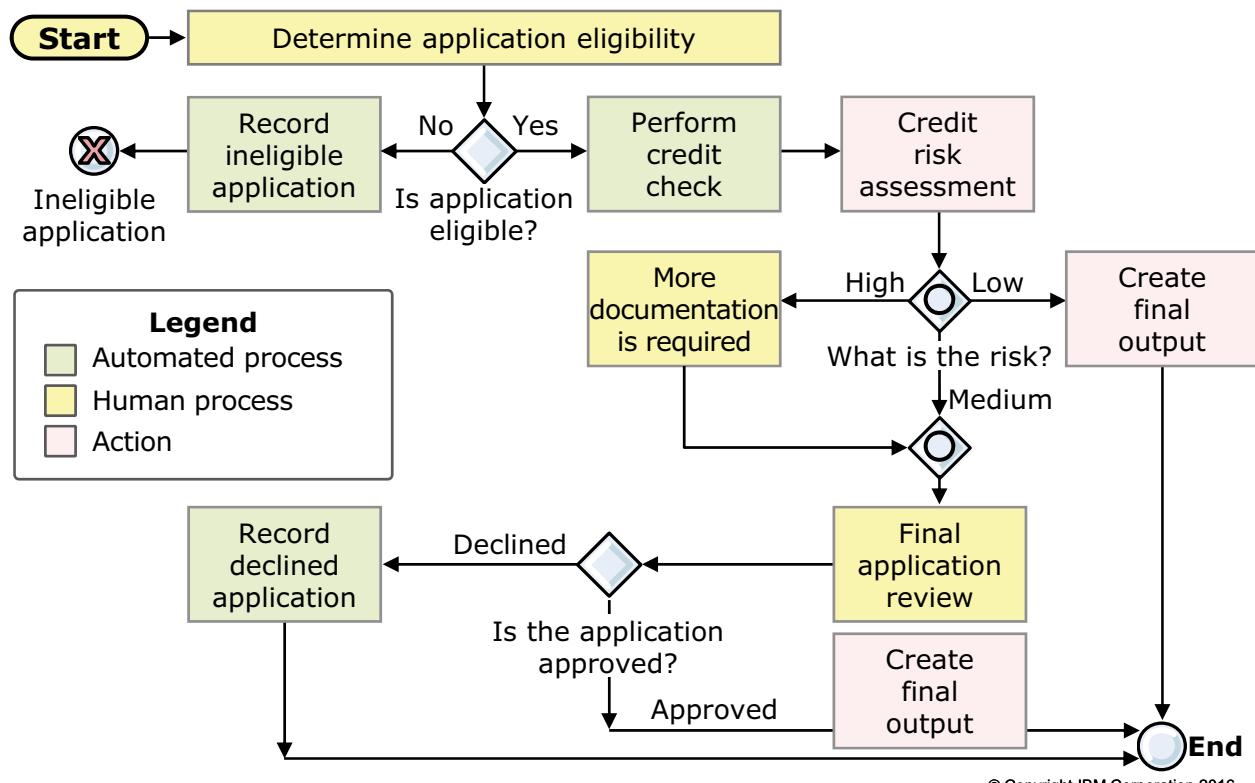
© Copyright IBM Corporation 2016

Figure 2-26. Business value of mediations

WB8601.2

Notes:

Account verification business process that you build in this course



© Copyright IBM Corporation 2016

Figure 2-27. Account verification business process that you build in this course

WB8601.2

Notes:

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. Ineligible applications are those requests that do not fall within the defined criteria for a credit request application. These applications do not go through any credit check process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit risk assessment is done 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; the customer must seek final approval for the application.

- 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.

Account verification process: BPD or BPEL?

- The account verification process can be built as either a BPMN process (BPD) or a BPEL process
- You use a slightly lighter version of this account verification process as a BPD in Exercise 1

In this course you create the account verification process as a BPEL process due to the following factors:

- Each individual activity in the BPD in Exercise 1 did not accomplish anything: the activities had no implementations
 - You build the implementations for the individual activities by using IBM Integration Designer
- Rich human interaction features are not a requirement
- Need for high-performance complex processes with error handling, compensation, and asynchronous processing
- Need for integration with complex external systems that require complex connectivity and data mapping

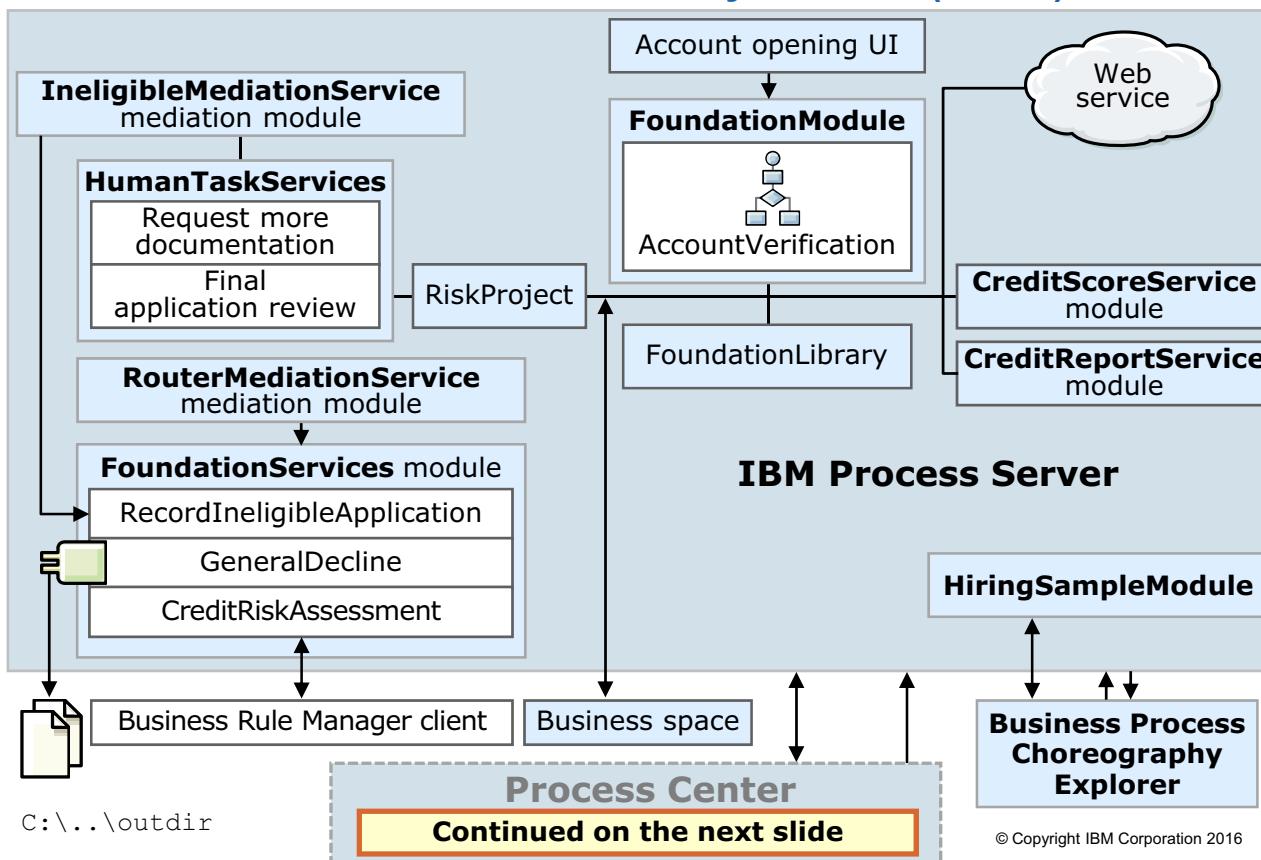
© Copyright IBM Corporation 2016

Figure 2-28. Account verification process: BPD or BPEL?

WB8601.2

Notes:

Course business scenario: What you build (1 of 2)



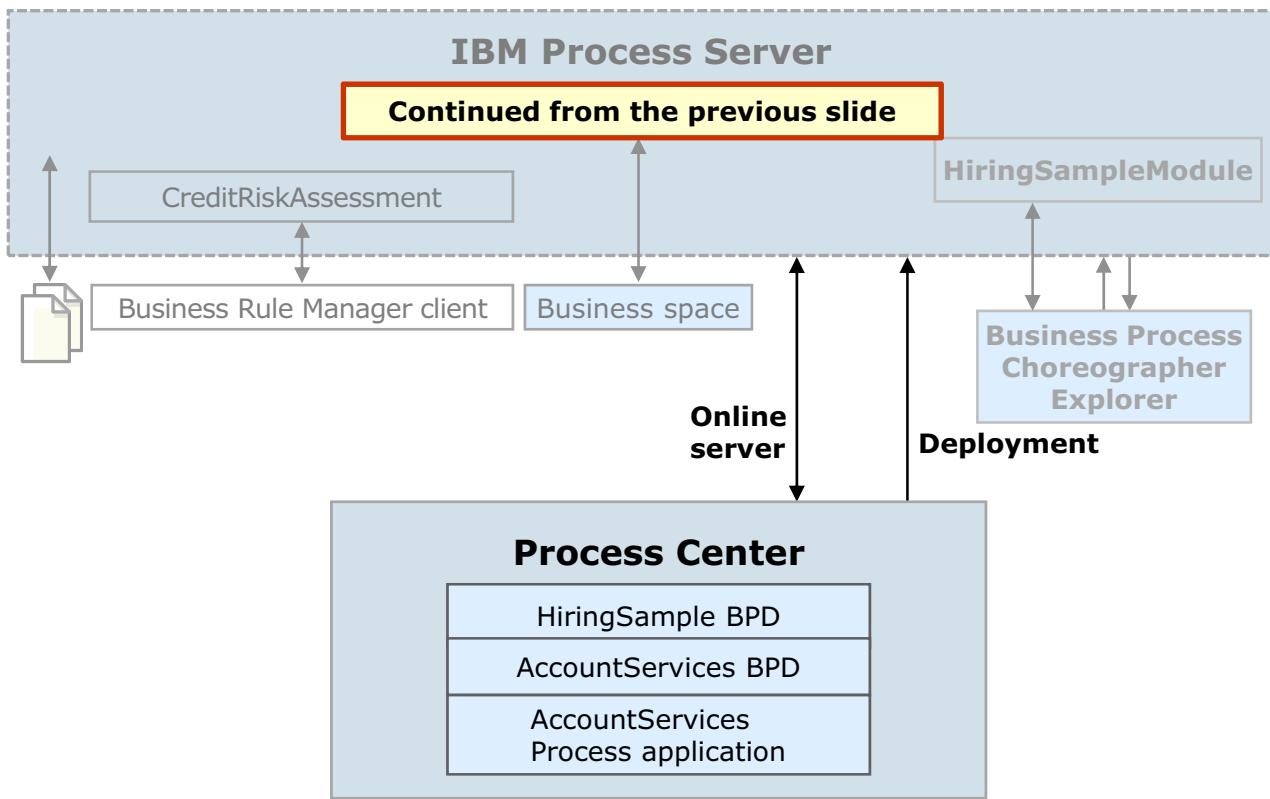
© Copyright IBM Corporation 2016

Figure 2-29. Course business scenario: What you build (1 of 2)

WB8601.2

Notes:

Course business scenario: What you build (2 of 2)



© Copyright IBM Corporation 2016

Figure 2-30. Course business scenario: What you build (2 of 2)

WB8601.2

Notes:



Unit summary

- Describe the differences between BPMN and BPEL
- Explain when to use the IBM BPM Advanced product
- Describe the capabilities of IBM Business Process Manager V8.5.7

© Copyright IBM Corporation 2016

Figure 2-31. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. True or False: A BPD is a business process that is created by using IBM Integration Designer.
2. True or False: An organization that purchased IBM BPM Advanced can develop its processes only in Integration Designer that uses BPEL.
3. True or False: You would use IBM BPM Standard when you require straight-through processing (STP) of business processes with no human interaction.

© Copyright IBM Corporation 2016

Figure 2-32. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. False. A BPD is a business process that is created by using IBM Process Designer.
2. False. An organization that purchased IBM BPM Advanced can develop its processes in both Integration Designer that uses BPEL and Process Designer that uses BPMN. Process Designer is available in both IBM BPM Standard and IBM BPM Advanced editions. All IBM BPM Standard features and tools are included in IBM BPM Advanced.
3. False. You would use IBM BPM Advanced when you require straight-through processing (STP) of business processes with no human interaction. IBM BPM Standard is best suited for human-centric interactive business processes.

© Copyright IBM Corporation 2016

Figure 2-33. Checkpoint answers

WB8601.2

Notes:

Exercise 1



Exploring IBM Process Designer and
IBM Process Portal

© Copyright IBM Corporation 2016

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

10.1

Figure 2-34. Exercise 1

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Start IBM Process Designer
- Open a business process activity in IBM Process Designer
- Explore a business process in IBM Process Designer
- Use the Playback feature to examine a running business process in IBM Process Designer Inspector
- Use a coach to work with a running business process
- Wire activities together in a business process
- Start IBM Process Portal and explore its collaboration capabilities

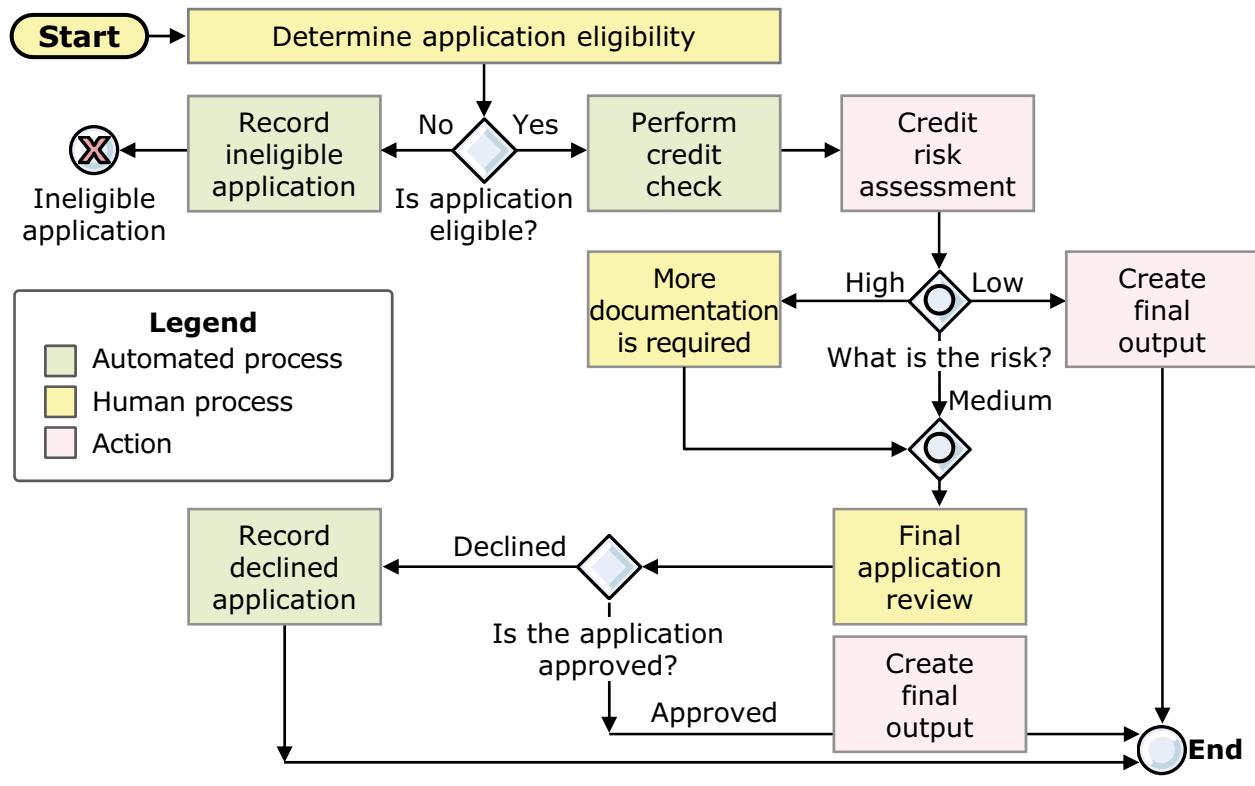
© Copyright IBM Corporation 2016

Figure 2-35. Exercise objectives

WB8601.2

Notes:

Account verification process: Overview



© Copyright IBM Corporation 2016

Figure 2-36. Account verification process: Overview

WB8601.2

Notes:

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 done 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.
- 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.

WebSphere Education



Course business scenario: What you build

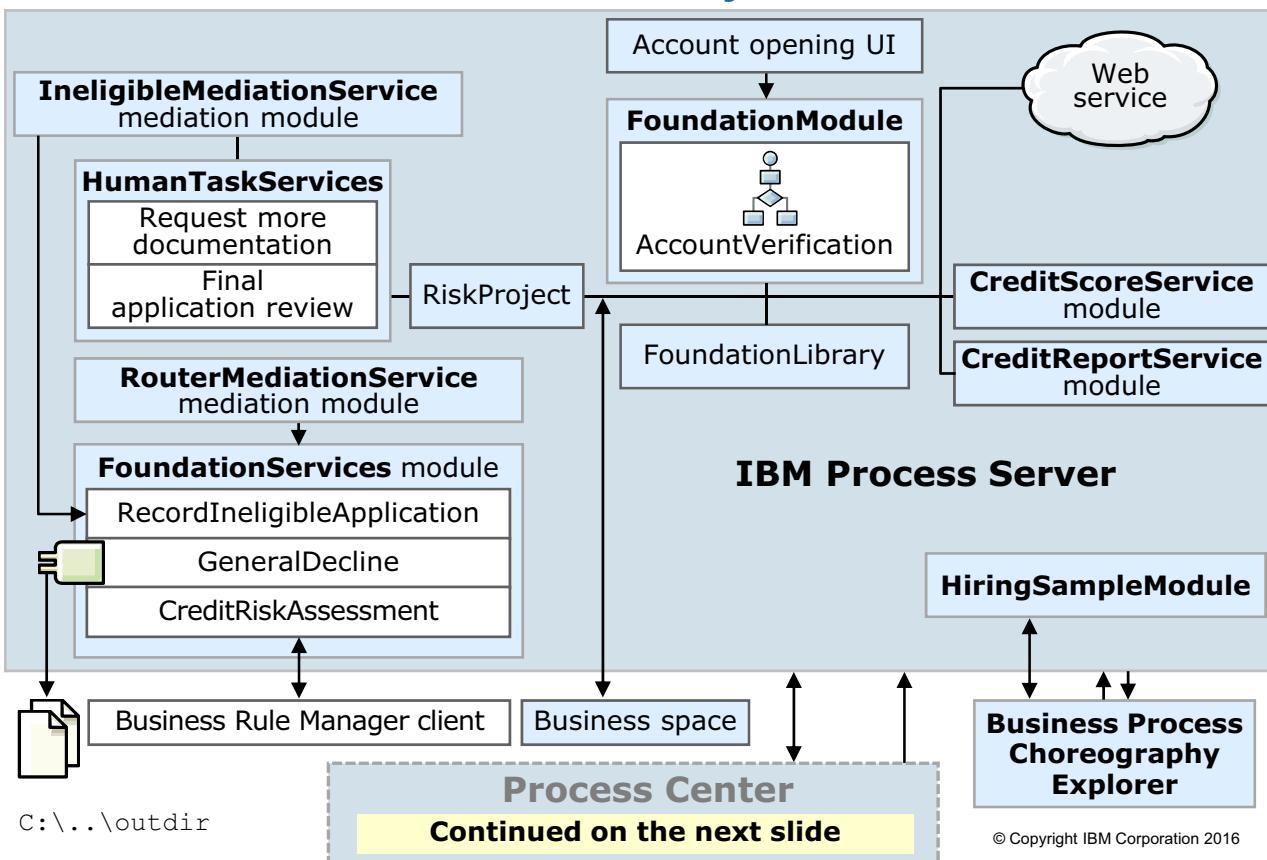


Figure 2-37. Course business scenario: What you build

WB8601.2

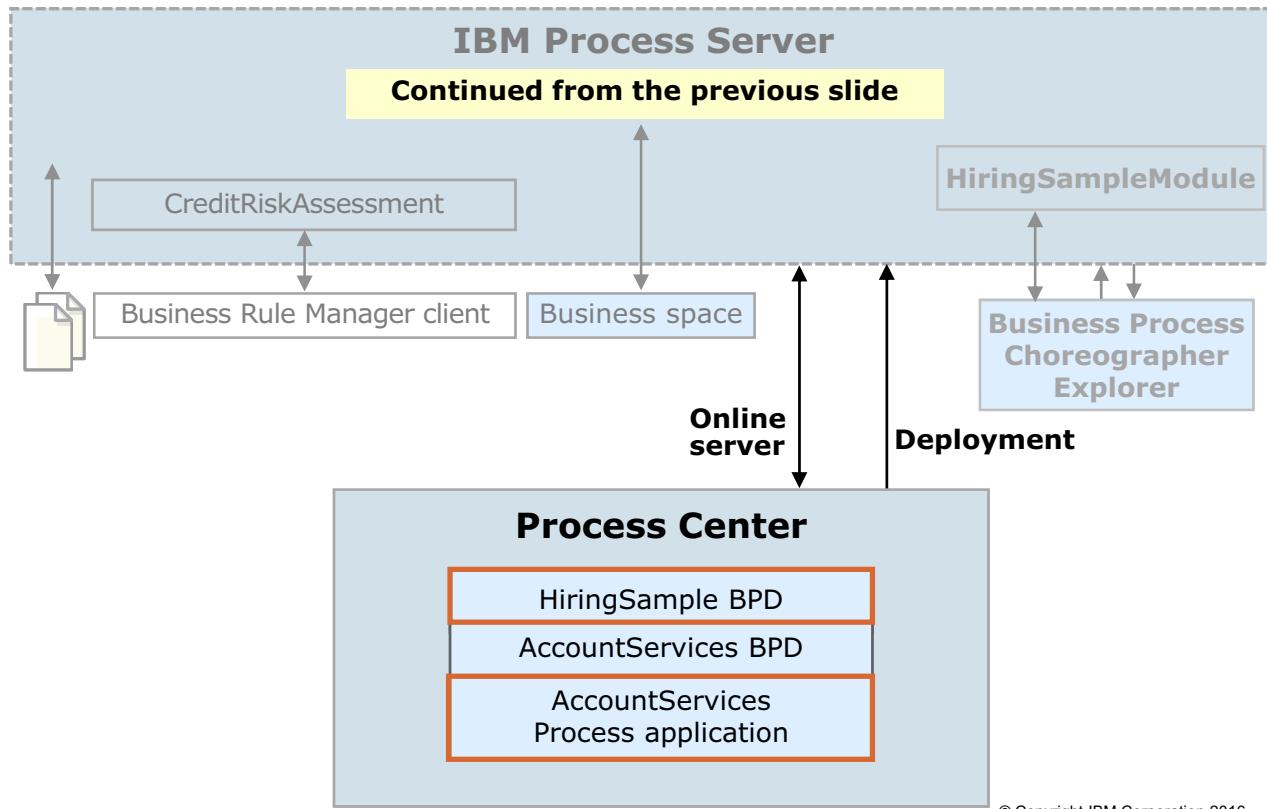
Notes:

In this course, you create the Account verification BPEL process from scratch. You also build integration solution with several SCA components, integration module, mediation module, and libraries.

For every exercise, you either build new artifacts or build on top of existing assets.

The HumanTaskServices module and its related human tasks are already prebuilt for this course. You do not explore, build, or enhance these human tasks. While the human task components are available in IBM BPM Advanced edition, you are not encouraged to create new human task components for your IBM BPM solution. For human interaction with the business process, the coach user interface is the right solution. You learn about coaches in the IBM BPM Standard course.

Course business scenario: What you work with



© Copyright IBM Corporation 2016

Figure 2-38. Course business scenario: What you work with

WB8601.2

Notes:

You work with the Process Center environment only in Exercises 1, 15, and 16.

Components that are required for Exercise 1

Prebuilt components that are imported in the lab:

- 1. AccountVerification.twx** process application
- 2. HiringSample.twx** process application

The screenshot shows a process diagram titled "Standard HR Open New Position". It starts with a "Start" node, followed by an activity labeled "Submit job requisition". From this activity, two parallel paths branch out. The top path leads to a decision diamond labeled "Need G approval". From there, it can lead to another decision diamond labeled "Approval r", which then leads to a final activity labeled "Approve/reject requis". The bottom path from the initial activity leads directly to the "Approve/reject requis" activity. A callout box contains the text: "See the Documentation field for each activity and event for information about the artifact.".

New components that you create in the lab:

None

© Copyright IBM Corporation 2016

Figure 2-39. Components that are required for Exercise 1

WB8601.2

Notes:

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, you familiarize yourself with Process Center and some of the tools available with it.

`AccountVerification.twx` and `HiringSample.twx` are two separate process applications that are already installed on the Process Center. Both of these applications have their own business processes that you review in this exercise.

The `AccountVerification.twx` business process or BPD that you review in this exercise is also implemented as a BPEL process that you work with in the Integration Designer in later exercises.

The screenshot shows the WebSphere Education Process Portal interface. At the top, there's a navigation bar with 'WebSphere Education' and the IBM logo. Below it, a main content area has a title 'Collaborate by using Process Portal in Exercise 1'. The interface includes a sidebar with 'Details', 'Experts', and 'Stream' tabs, and a central panel for user 'Bob'. The user info includes a yellow star icon, 'Due Date: Apr 4, 2016', 'Department: Product', 'Employment Status: Full-time', 'Hiring Manager: Henry Miller', and 'Location: Minneapolis'. A 'Tasks' section lists a completed task for 'ajane' and a pending step for 'GM Approval'. A callout box points to the 'Stream' tab, stating 'Users can collaborate in real time with other users and experts'. Another callout box highlights the 'GM Approval' step, which is part of a process named 'Step: GM Approval'.

© Copyright IBM Corporation 2016

Figure 2-40. Collaborate by using Process Portal in Exercise 1

WB8601.2

Notes:

IBM Business Process Manager provides several features that allow business users to collaborate with other users while working with processes.

In this exercise, you work with the Process Portal tool. With this tool, users can request help from other users, collaborate in real time with other users on task completion, and communicate through messages in the process activity stream. All of these features are available without any configuration of your process applications.

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 invite one or more users to collaborate with you.

Unit 3. Course business scenario: What you build

What this unit is about

This unit describes the lab scenario and lists the components that you build in each lab exercise of this course.

What you should be able to do

After completing this unit, you should be able to:

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

How you will check your progress

- Checkpoint questions

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 2016

Figure 3-1. Unit objectives

WB8601.2

Notes:

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

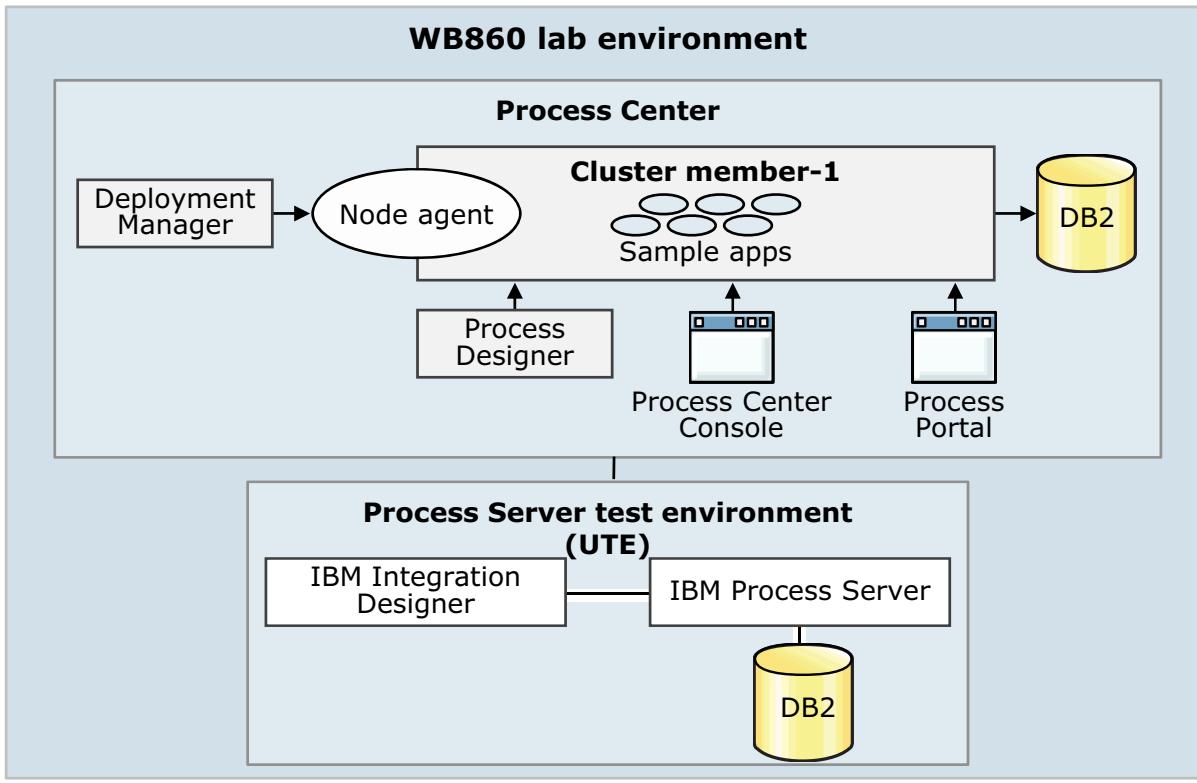
© Copyright IBM Corporation 2016

Figure 3-2. Some important points

WB8601.2

Notes:

WB860 lab environment



© Copyright IBM Corporation 2016

Figure 3-3. WB860 lab environment

WB8601.2

Notes:

Exercise 1



Exploring IBM Process Designer and
IBM Process Portal

© Copyright IBM Corporation 2016

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

10.1

Figure 3-4. Exercise 1

WB8601.2

Notes:



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

- Explored the Process Designer tool and the Process Center environment
- Reviewed and changed a business process (BPD) in Process Designer
- Used the Process Portal tool for team collaboration

© Copyright IBM Corporation 2016

Figure 3-5. What are some key tasks that you did in Exercise 1?

WB8601.2

Notes:

Components that are required for Exercise 1

Prebuilt components that are imported in the lab:

- 1. AccountVerification.twx** process application
- 2. HiringSample.twx** process application

New components that you created in the lab:

None

© Copyright IBM Corporation 2016

Figure 3-6. Components that are required for Exercise 1

WB8601.2

Notes:

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, you familiarize yourself with Process Center and some of the tools available with it.

`AccountVerification.twx` and `HiringSample.twx` are two separate process applications that are already installed on the Process Center. Both of these applications have their own business process that you review in this exercise.

The `AccountVerification.twx` business process or BPD that you review in this exercise is also implemented as a BPEL process that you work with in the Integration Designer in later exercises.

Account verification process: Overview

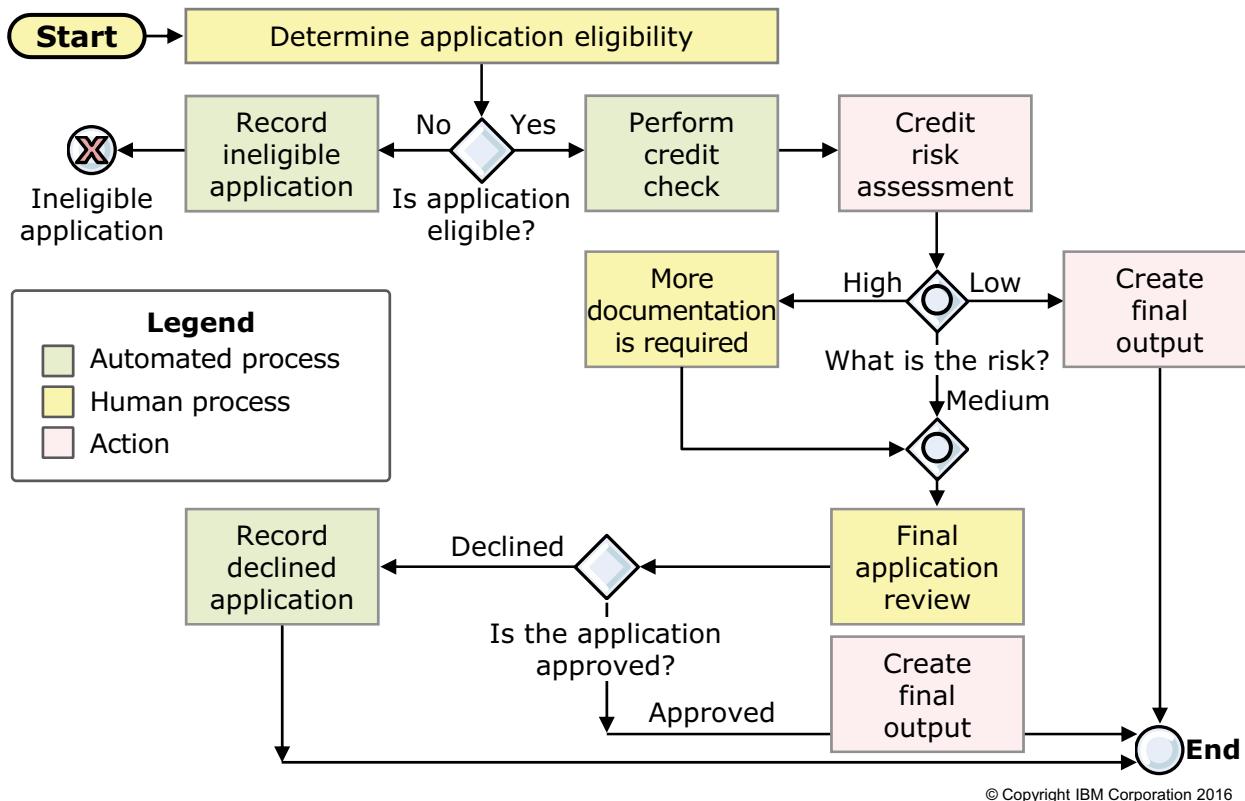


Figure 3-7. Account verification process: Overview

WB8601.2

Notes:

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.
- 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.

The screenshot shows the WebSphere Education Process Portal. At the top, there's a navigation bar with 'WebSphere Education' and the IBM logo. Below it, a main title says 'Collaborate by using Process Portal in Exercise 1'. The interface includes a sidebar with 'Details', 'Experts', and 'Stream' tabs. A yellow callout box points to the 'Stream' tab with the text: 'Users can collaborate in real time with other users and experts'. The main content area shows a user profile for 'Bob' with details like 'Due Date: Apr 4, 2016', 'Department: Product', 'Employment Status: Full-time', 'Hiring Manager: Henry Miller', and 'Location: Minneapolis'. Below this is a 'Tasks' section. Two specific items are highlighted with orange boxes: one for a task 'Submit requisition' completed by 'ajane' and another for a step 'GM Approval' assigned to 'GeneralManagers'. An arrow points from the 'GM Approval' step to the 'Stream' tab in the sidebar.

© Copyright IBM Corporation 2016

Figure 3-8. Collaborate by using Process Portal in Exercise 1

WB8601.2

Notes:

IBM Business Process Manager provides several features that allow business users to collaborate with other users while working with processes.

In this exercise, you work with the Process Portal tool. With this tool, users can request help from other users, collaborate in real time with other users on task completion, and communicate through messages in the process activity stream. All of these features are available without any configuration of your process applications.

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 invite one or more users to collaborate with you.

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

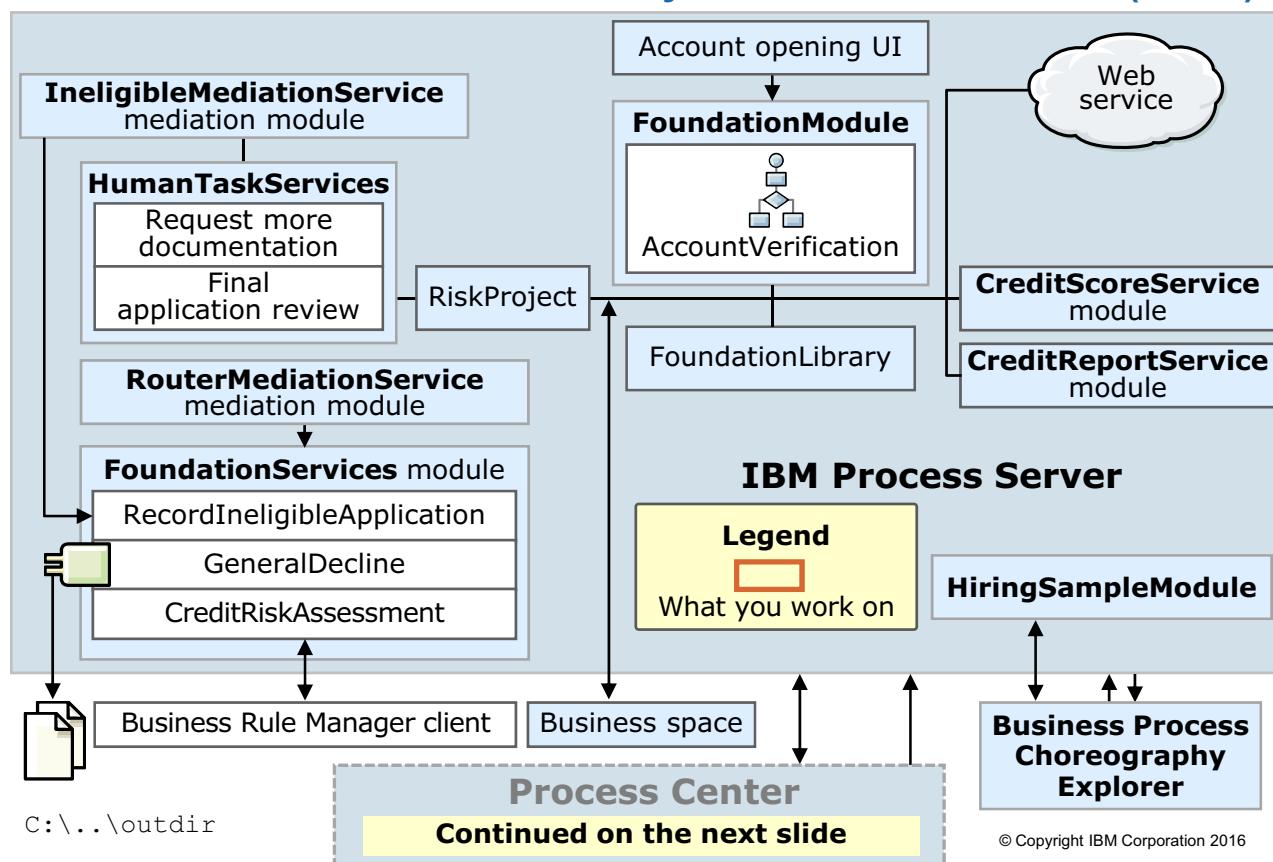


Figure 3-9. Course business scenario: What you worked with in Exercise 1 (1 of 2)

WB8601.2

Notes:

In this lab, you do not work with Integration Designer.

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

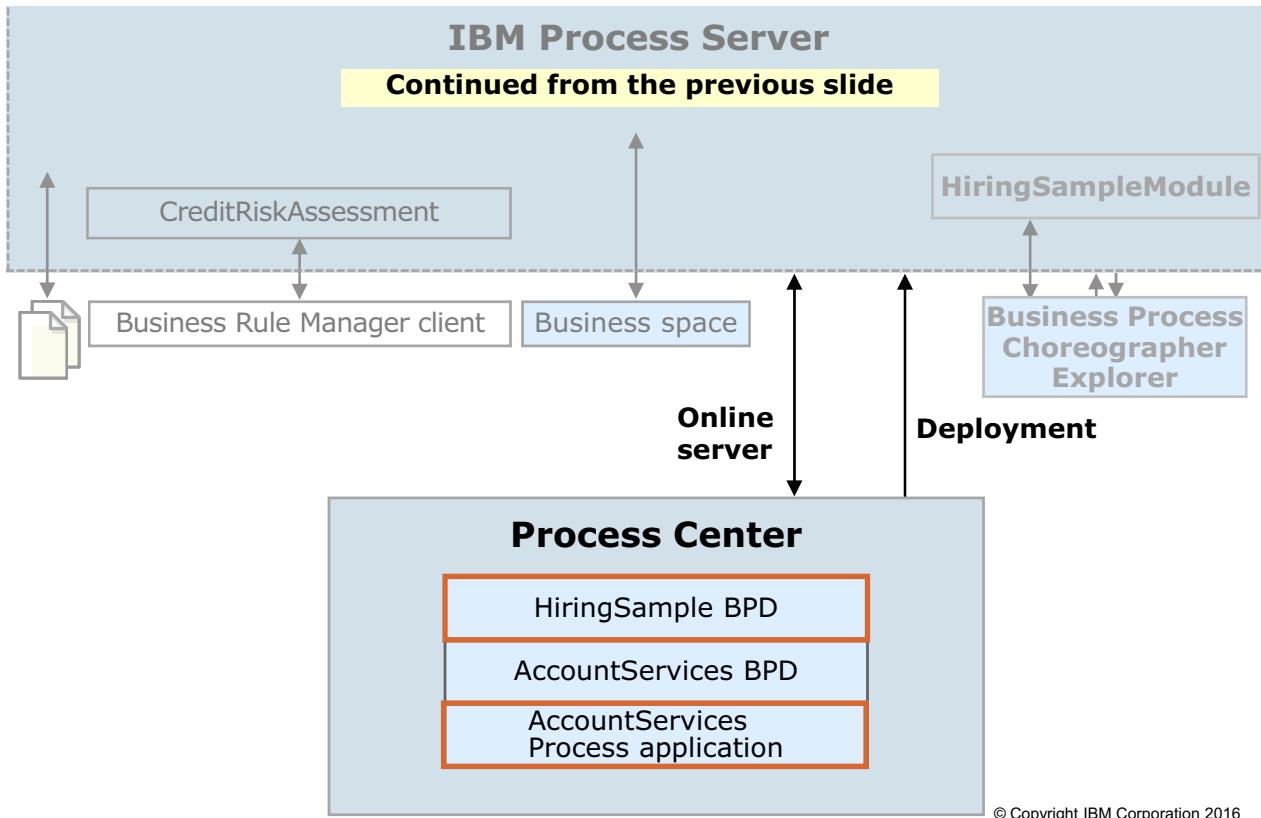


Figure 3-10. Course business scenario: What you worked with in Exercise 1 (2 of 2)

WB8601.2

Notes:

You work only with the Process Center environment.

Exercise 2



Exploring IBM Integration Designer,
part I

© Copyright IBM Corporation 2016

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

10.1

Figure 3-11. Exercise 2

WB8601.2

Notes:



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

- Explore the IBM Integration Designer tool and its capabilities

© Copyright IBM Corporation 2016

Figure 3-12. What are some key tasks that you do in Exercise 2?

WB8601.2

Notes:

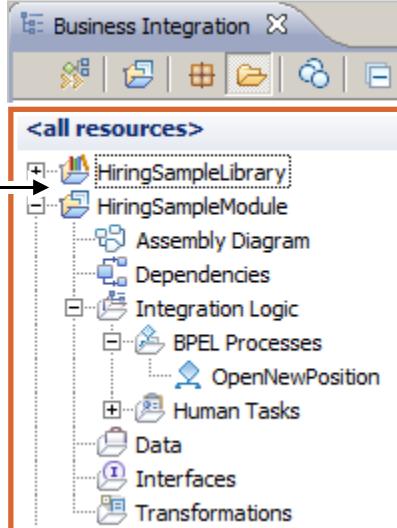
Recall that IBM Integration Designer is the tool that is available **only** with the Advanced edition of IBM Business Process Manager. To get the most value of the product, you need to understand the capabilities that the Integration Designer tool offers. This exercise gives a high-level overview of some of the tool capabilities.

Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

- 1. HiringSampleLibrary
- 2. HiringSampleModule

Business Integration X



<all resources>

- + HiringSampleLibrary
- HiringSampleModule
- Assembly Diagram
- Dependencies
- Integration Logic
 - BPEL Processes
 - OpenNewPosition
 - Human Tasks
- Data
- Interfaces
- Transformations

New components that you create in the lab:

None

© Copyright IBM Corporation 2016

Figure 3-13. Components that are required for Exercise 2

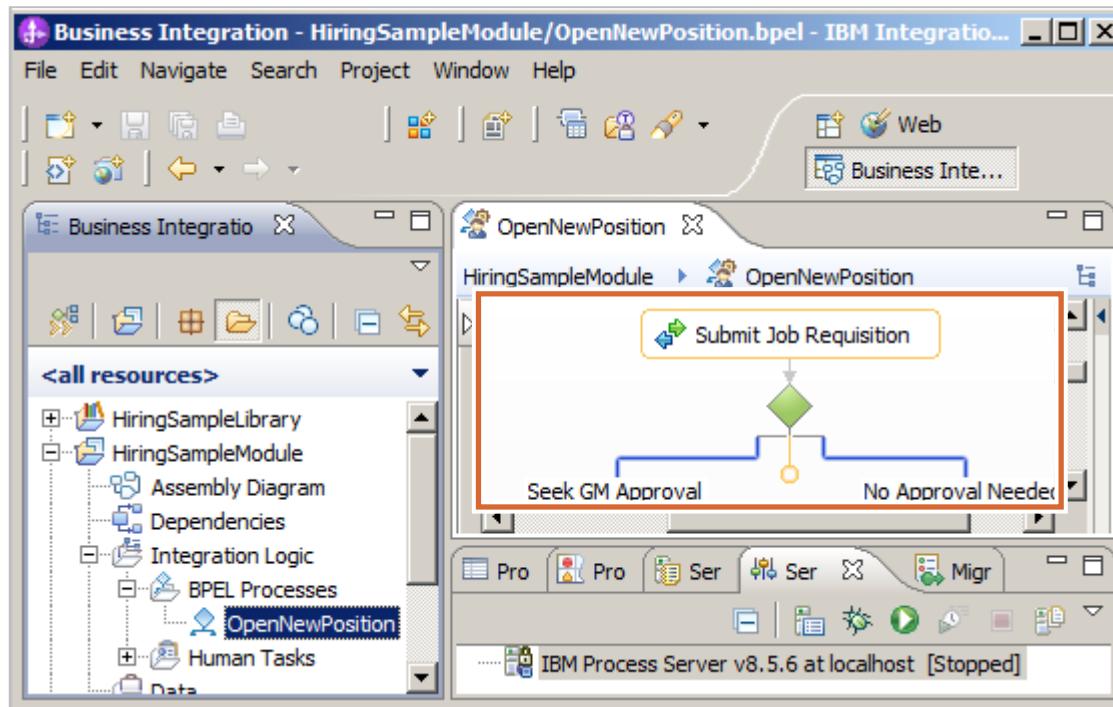
WB8601.2

Notes:

HiringSampleLibrary and HiringSampleModule are two of the artifacts in IBM Integration Designer that are already built for you. In this exercise, you explore these two artifacts.

WebSphere Education

Explore IBM Integration Designer in Exercise 2



© Copyright IBM Corporation 2016

Figure 3-14. Explore IBM Integration Designer in Exercise 2

WB8601.2

Notes:

You view the HiringSample process in IBM Integration Designer. This process is the same BPD that you explored in Exercise 1 by using the Process Center. The difference is that in this exercise you use the BPEL version of the HiringSample process. For differences between BPD and BPEL, see the previous unit.

Course business scenario: What you work with in Ex 2

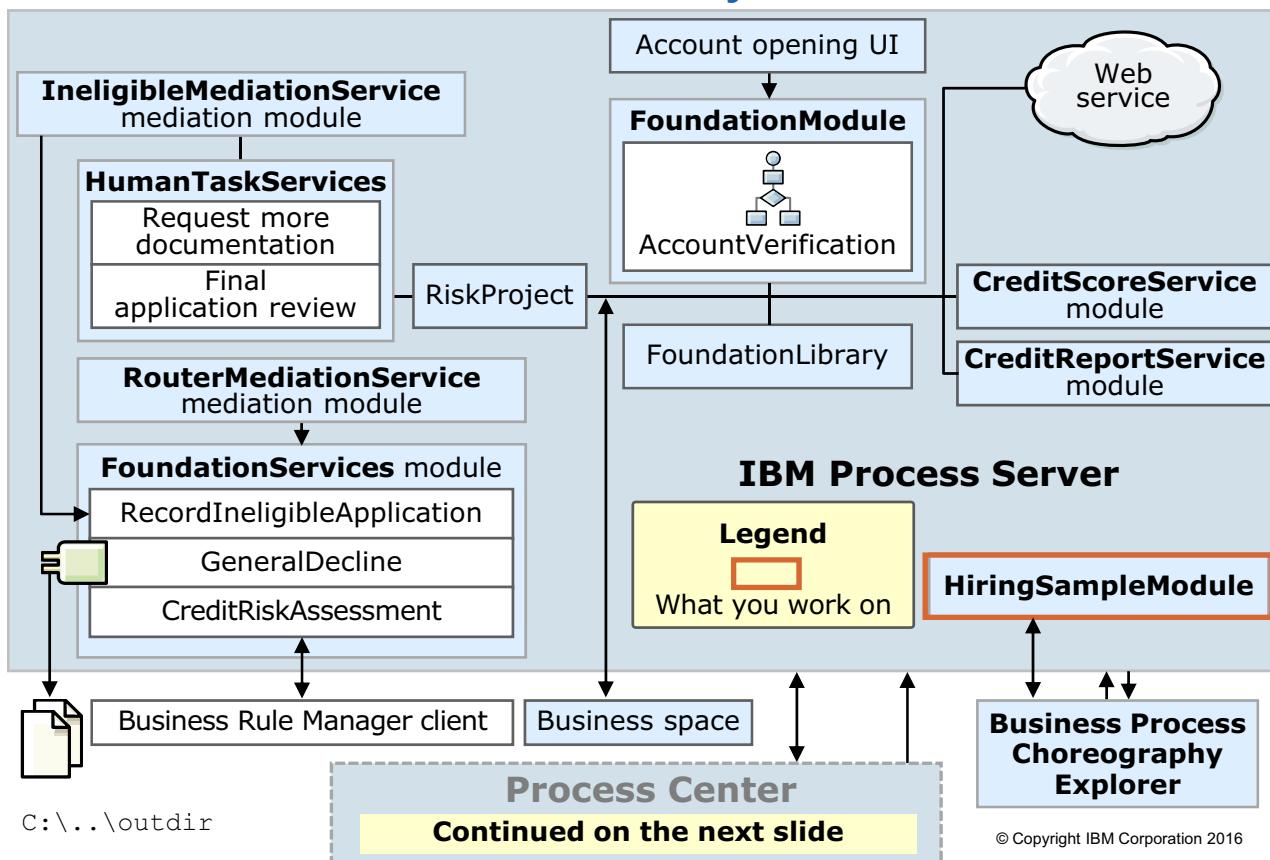


Figure 3-15. Course business scenario: What you work with in Exercise 2

WB8601.2

Notes:

In this exercise, you explore **HiringSampleModule**.

Exercise 3



Exploring IBM Integration Designer,
part II

© Copyright IBM Corporation 2016

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

10.1

Figure 3-16. Exercise 3

WB8601.2

Notes:

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

- Continue exploring the various components of IBM Integration Designer
- Test an SCA application
- Explore Business Process Choreographer Explorer
- Use cross-component trace to examine a running business process

© Copyright IBM Corporation 2016

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

WB8601.2

Notes:

You continue where you left off from the previous exercise of exploring IBM Integration Designer. Next, you work with an SCA application, and then test it to verify the results.

An important aspect of building SCA applications is testing them, and you do that by investigating the server logs. Cross-component trace is an extensive way to troubleshoot SCA components in logs. You do that in this exercise.

You also use and become familiar with the Business Process Choreographer Explorer tool, which is a web application that is already installed as part of the configuration of the process container.



Components that are required for Exercise 3

Prebuilt components that are imported in the lab:

- 1. **HiringSampleLibrary**
- 2. **HiringSampleModule**

Previously imported in Exercise 2

A screenshot of the Business Integration tool interface. A red box highlights the 'call resources' section on the left, which contains a tree view of artifacts. An arrow points from the 'HiringSampleLibrary' item in the tree to the list of imported components on the left. The tree view includes categories like Assembly Diagram, Dependencies, Integration Logic (with BPEL Processes), Human Tasks, Data, Interfaces, and Transformations.

New components that you create in the lab:

None

© Copyright IBM Corporation 2016

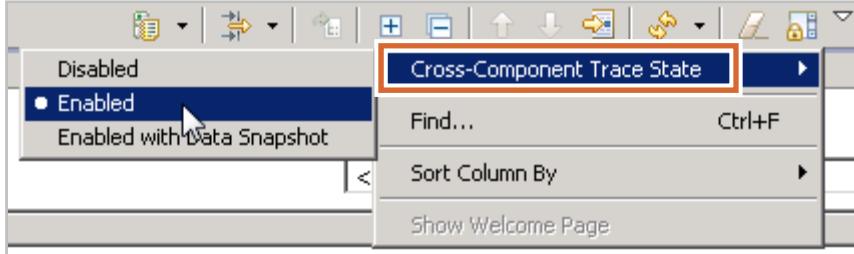
Figure 3-18. Components that are required for Exercise 3

WB8601.2

Notes:

HiringSampleLibrary and HiringSampleModule are already built and imported for you in this lab. You do not build any new artifacts in this exercise.

Use cross-component tracing in Exercise 3



The screenshot shows the application's main menu bar with several icons. Below the menu bar, a context-sensitive menu is open, listing options: 'Disabled', 'Enabled' (which is selected and highlighted in blue), 'Enabled with Data Snapshot', 'Find...', 'Sort Column By', and 'Show Welcome Page'. The 'Enabled' option is the current selection.

Type	Thread ID	Contents
Invocation sequence (OpenNewPosition:createPosition)	000000f5	
Start invoke (OpenNewPosition:createPosition)	000000f5	Start of the one-way invocation of
Start component (OpenNewPosition:createPosition)	00000128	Start of the component processing
In BPEL process	00000128	07e8a3dd-ddab-4640-be64-54349
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL proce
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL proces
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL proce
Log message	00000126	>>> Find Job Candidates --- No A
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL proces
End component (OpenNewPosition:createPosition)	00000128	End of the component processing
End invoke (OpenNewPosition:createPosition)	000000f5	End of the one-way invocation of

© Copyright IBM Corporation 2016

Figure 3-19. Use cross-component tracing in Exercise 3

WB8601.2

Notes:

The reason that cross-component trace is introduced early on in this course is so that you can use this feature on your own to troubleshoot any lab. Any time that you work with an SCA application, enabling this trace often identifies the problem component.

Welcome to Business Process Choreographer Explorer.

The page you requested is only available to registered users. Enter your username and password and click Login.

User Name: _____

Password: _____

Login

My To-dos

Use this page to work on tasks that are assigned to you. [\[i\]](#)

Work on Release Transfer Start Change Bus

<input type="checkbox"/> Priority	Task Name	State	Kind	Owner
<input checked="" type="checkbox"/> 5	SubmitRequisition	Ready	To-do Task	Tom
<input type="checkbox"/> 5	GMAApproval	Ready	To-do Task	Tom

Items found: 1 Items selected: 1

Work on Release Transfer Start Change Business Category

<input type="checkbox"/> Priority	Task Name	State	Kind	Owner
<input type="checkbox"/> 5	SubmitRequisition	Ready	To-do Task	Tom
<input type="checkbox"/> 5	GMAApproval	Ready	To-do Task	Tom

Items found: 2 Items selected: 0

Page 1 of 1

© Copyright IBM Corporation 2016

Figure 3-20. Test by using the Business Process Choreographer Explorer

WB8601.2

Notes:

In this exercise, you start and log in to the Business Process Choreographer Explorer web application.

When working with IBM Business Process Manager, depending on your user role, you can use these client interfaces to manage BPEL processes and human tasks. You can also work with your assigned tasks, view completed BPEL processes and human tasks that are in an archive database, or delete processes and tasks from the archive. Both client interfaces offer a search function that you can use to discover BPEL processes and their related activities and human tasks that meet specific criteria. For example, you can check the status of these instances, go between related instances and templates, and retrieve a graphical view of the process structure and process states, which include the associated activities and human tasks.

In this exercise, you work on the existing submitRequisition task and accept or reject it.

Course business scenario: What you work with in Ex 3

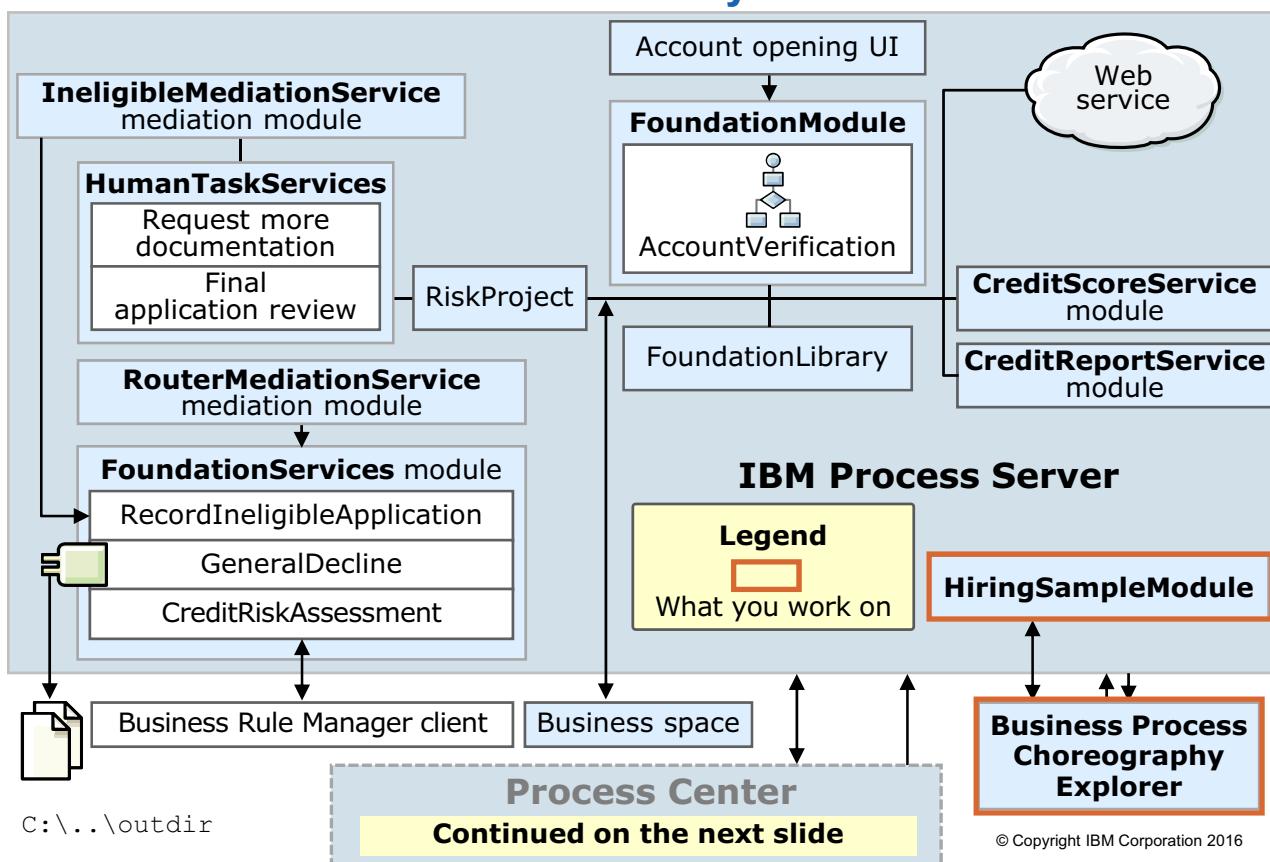


Figure 3-21. Course business scenario: What you work with in Exercise 3

WB8601.2

Notes:



Exercise 4



Working with web services

© Copyright IBM Corporation 2016

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

10.1

Figure 3-22. Exercise 4

WB8601.2

Notes:



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

- Use the integration test client to invoke and test a web service
- Expose an IBM Process Server application as a web service

© Copyright IBM Corporation 2016

Figure 3-23. What are some key tasks that you do in Exercise 4?

WB8601.2

Notes:

Components that are required for Exercise 4 (Section 1)

Prebuilt components that are imported in the lab:

1. CreditReportService.war

- Contains the **CreditReportServiceEAR** application
- You deploy the application to IBM Process Server

2. CreditReportService.wsdl

- Interface that describes the **CreditReportService** web service
- Defines an operation **CalculateCreditScore**
- Defines the **CreditCheckDetail** business object

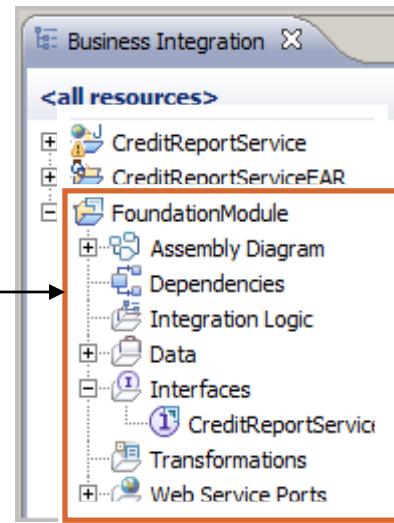
New components that you create in the lab:

1. FoundationModule

- Encapsulates and groups artifacts by type

2. CreditReportService

- SCA Import component that invokes the **CreditReportService** web service



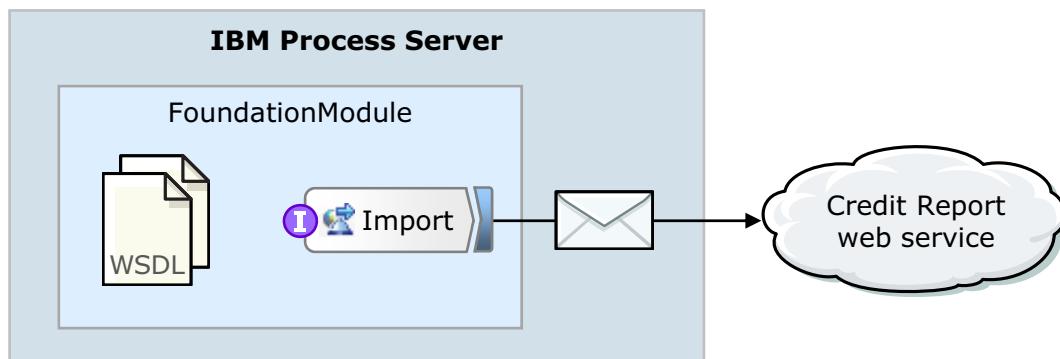
© Copyright IBM Corporation 2016

Figure 3-24. Components that are required for Exercise 4 (Section 1)

WB8601.2

Notes:

Invoke an external web service in Exercise 4 (Section 1)



© Copyright IBM Corporation 2016

Figure 3-25. Invoke an external web service in Exercise 4 (Section 1)

WB8601.2

Notes:

WebSphere Education

Components that are required for Exercise 4 (Section 2)

Prebuilt components that are imported in the lab:

1. CreditScoreService

- Module containing the **CreditScoreRG** rule group that calculates the credit score

2. CreditScoreService.wsdl

- Interface that describes the **CreditScoreService** web service
- Defines the **CreditCheckRequest** business object

New components that you create in the lab:

1. FoundationLibrary

- Container for shared business objects and interfaces:
CreditScoreService.wsdl interface and
CreditCheckRequest business object

2. CreditScoreService

- Export SCA component that exposes the **CreditScoreService** web service

3. CreditScoreService

- Import SCA component that calls the service through the SCA export

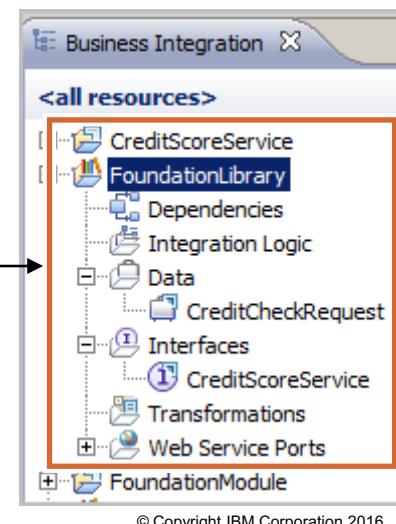
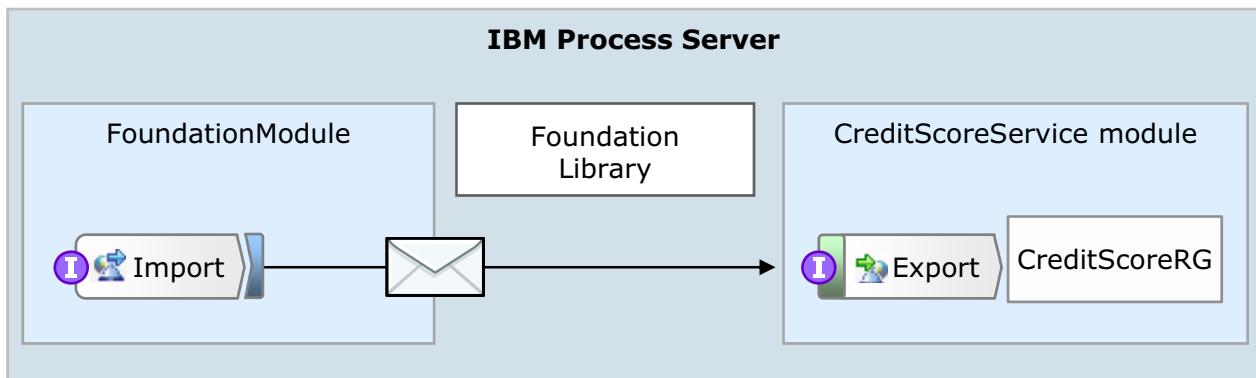


Figure 3-26. Components that are required for Exercise 4 (Section 2)

WB8601.2

Notes:

Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)



© Copyright IBM Corporation 2016

Figure 3-27. Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 4

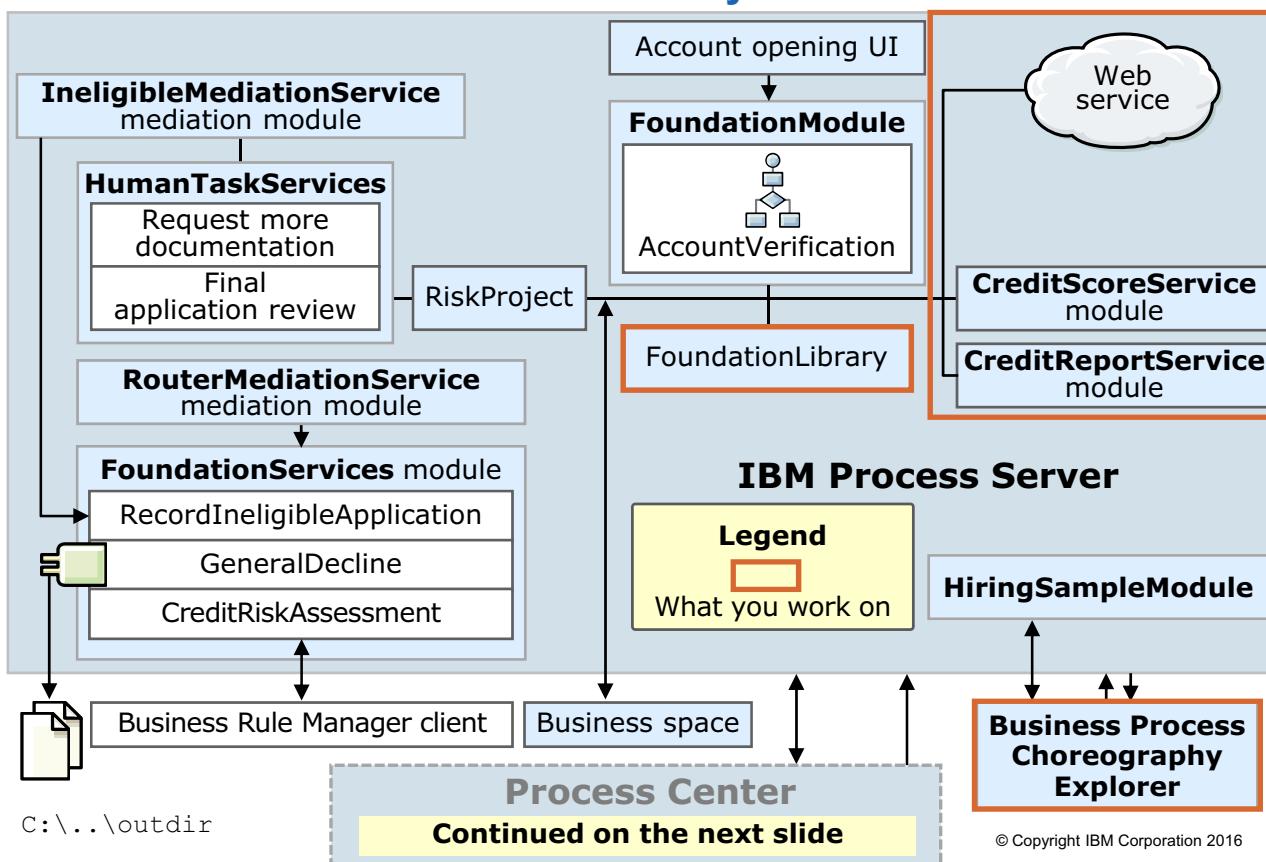


Figure 3-28. Course business scenario: What you work with in Exercise 4

WB8601.2

Notes:

Exercise 5



Creating business objects and shared interfaces

© Copyright IBM Corporation 2016

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

10.1

Figure 3-29. Exercise 5

WB8601.2

Notes:



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

- Implement business objects and define their elements
- Implement interfaces that use business objects as inputs and outputs

© Copyright IBM Corporation 2016

Figure 3-30. What are some key tasks that you do in Exercise 5?

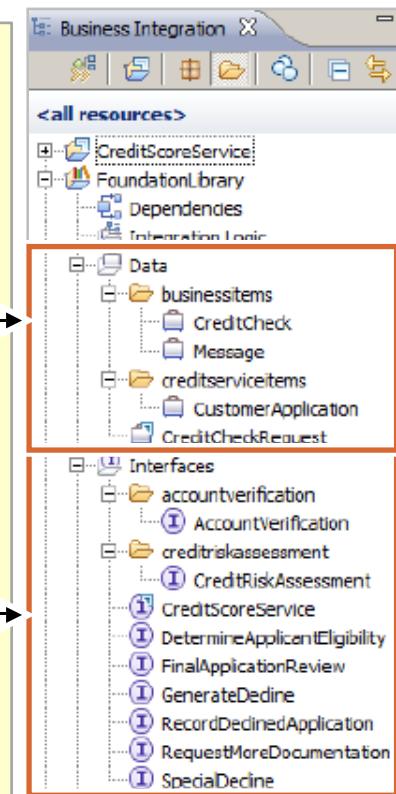
WB8601.2

Notes:

Components that are required for Exercise 5 (1 of 2)

Prebuilt components that are imported in the lab:

1. **FoundationModule** module
2. **CreditScoreService** module
3. **FoundationLibrary** container for **business objects**:
 - CreditCheck
 - Message
 - CustomerApplication
 - CreditCheckRequest
4. **FoundationLibrary** container for **interfaces**:
 - AccountVerification
 - CreditRiskAssessment
 - CreditScoreService
 - DetermineApplicantEligibility
 - FinalApplicationReview
 - GenerateDecline
 - RecordDeclineApplication
 - RequestMoreDocumentation
 - SpecialDecline



© Copyright IBM Corporation 2016

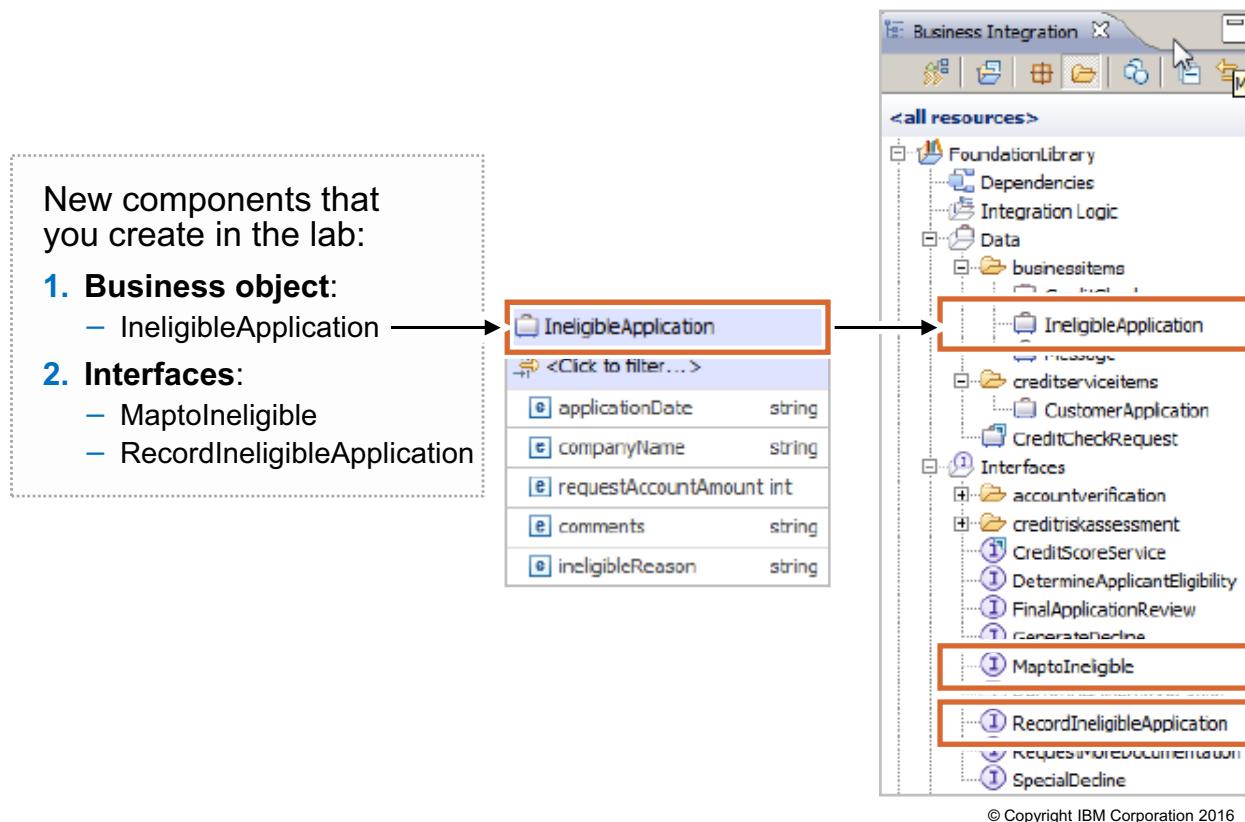
Figure 3-31. Components that are required for Exercise 5 (1 of 2)

WB8601.2

Notes:

WebSphere Education 

Components that are required for Exercise 5 (2 of 2)



© Copyright IBM Corporation 2016

Figure 3-32. Components that are required for Exercise 5 (2 of 2)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 5

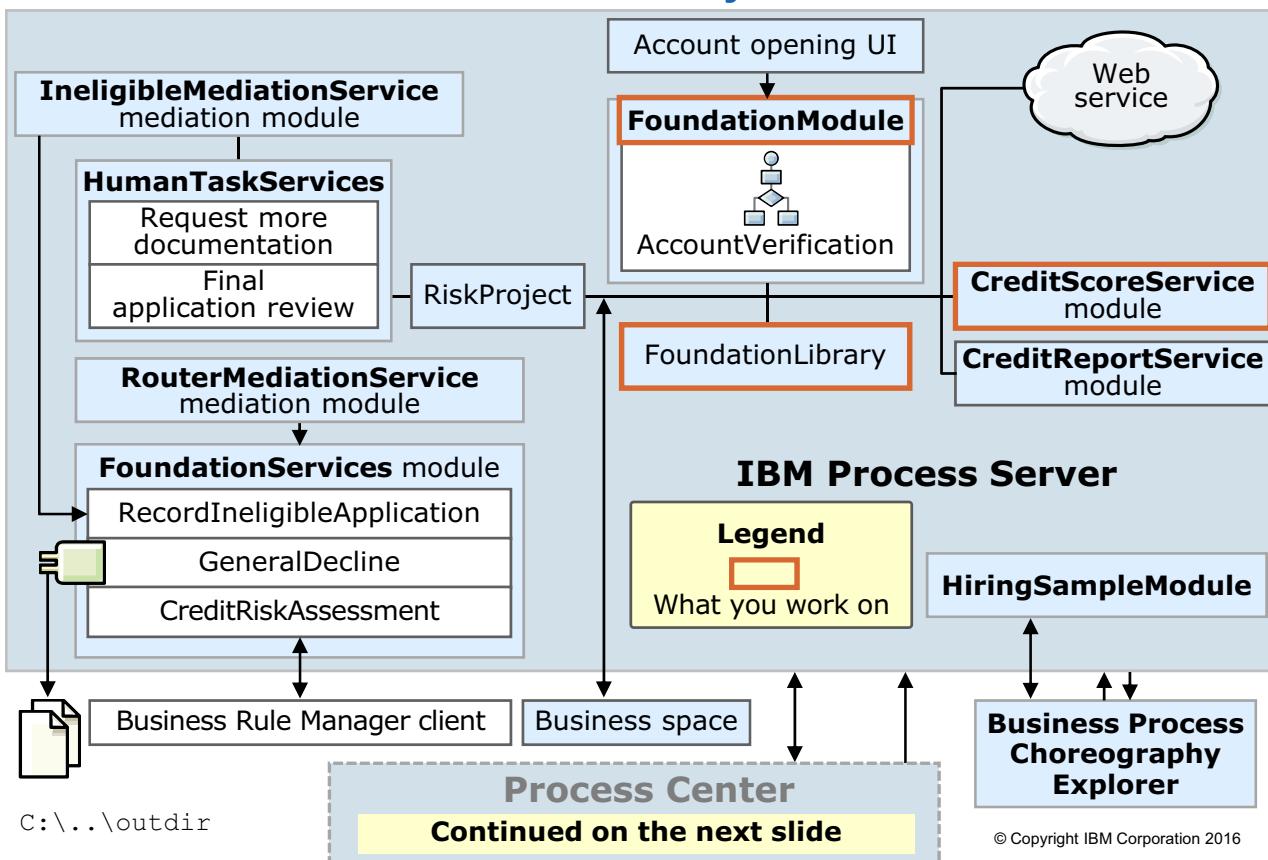


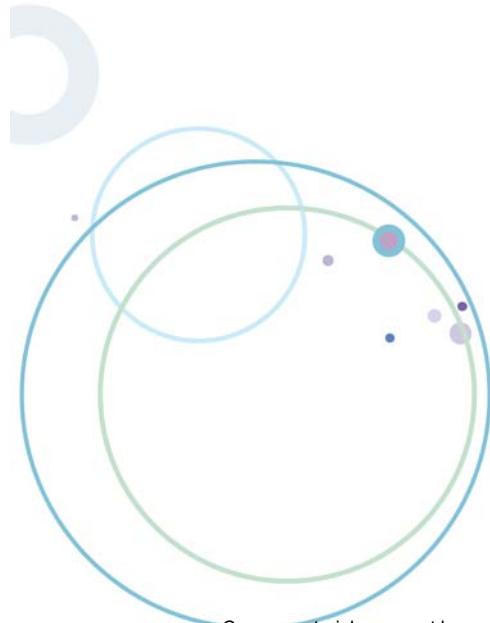
Figure 3-33. Course business scenario: What you work with in Exercise 5

WB8601.2

Notes:



Exercise 6



Creating a business process, part I

© Copyright IBM Corporation 2016

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

10.1

Figure 3-34. Exercise 6

WB8601.2

Notes:

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

- Build some of the various pieces of a BPEL process
- Create interface partners and references
- Create process variables
- Why create the account verification business process in IBM Integration Designer?
 - The business process application that you examined in IBM Process Designer in Exercise 1 is incomplete
 - Although the activities were in place and in a later section of the exercise, you wire them together and each individual activity did not accomplish anything: the activities had no implementations
 - In this exercise and the subsequent exercises, you build this process in IBM Integration Designer
 - More importantly, you focus on building the implementations for the individual activities

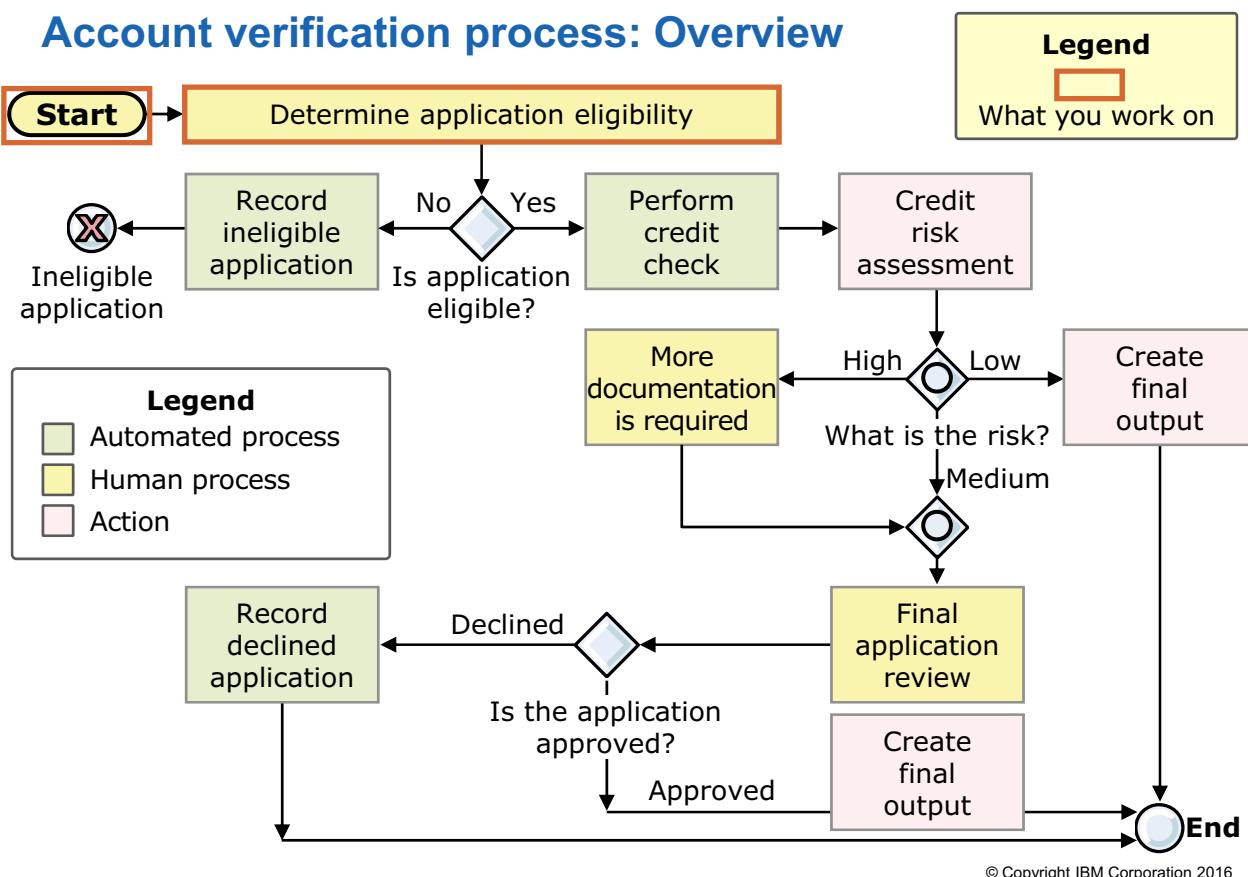
© Copyright IBM Corporation 2016

Figure 3-35. What are some key tasks that you do in Exercise 6?

WB8601.2

Notes:

Account verification process: Overview



© Copyright IBM Corporation 2016

Figure 3-36. Account verification process: Overview

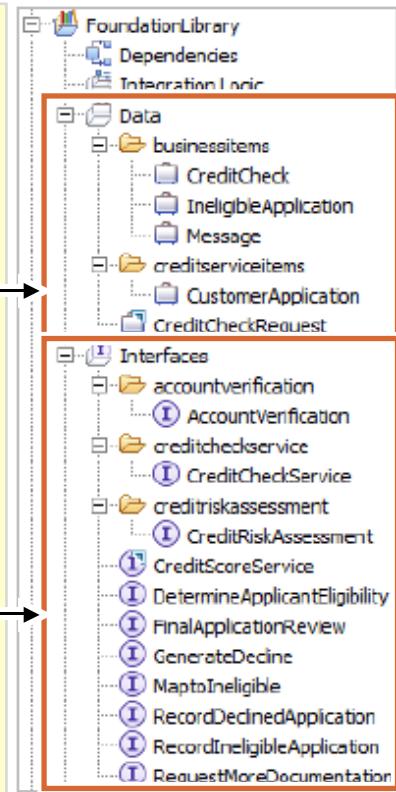
WB8601.2

Notes:

Components that are required for Exercise 6 (1 of 7)

Prebuilt components that are imported in the lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary container for business objects:**
 - CreditCheck
 - IneligibleApplication
 - Message
 - CustomerApplication
 - CreditCheckRequest
- 4. FoundationLibrary container for interfaces:**
 - AccountVerification
 - CreditCheckService
 - CreditRiskAssessment
 - CreditScoreService
 - DetermineApplicantEligibility
 - FinalApplicationReview
 - GenerateDecline
 - MaptoIneligible
 - RecordDeclineApplication
 - RequestMoreDocumentation



© Copyright IBM Corporation 2016

Figure 3-37. Components that are required for Exercise 6 (1 of 7)

WB8601.2

Notes:

Components that are required for Exercise 6 (2 of 7)

New components that you create in the lab:

1. **AccountVerification** BPEL process:
 - Uses the **AccountVerification** interface

Operation - InputCriterion	
Description	
Interface	AccountVerification
Operation	InputCriterion
Input	Input
Output	CustomerApplication Message

© Copyright IBM Corporation 2016

Figure 3-38. Components that are required for Exercise 6 (2 of 7)

WB8601.2

Notes:

An interface provides the input and output of a component. It is created independent of the internal implementation of the component. The AccountVerification interface dictates how to use the AccountVerification process. It specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions on the AccountVerification process.

The interface partner is a direct link to the interface where the partner is configured. An interface partner is the process interface and exposes operations that external users or services call.

Components that are required for Exercise 6 (3 of 7)

New components that you create in the lab:

2. Global variables:

- CreditCheckVariable
- IneligibleApplicationVariable
- CustomerApplicationVariable
- CustomerApplicationVariable2
- MessageVariable

Variables
CustomerApplicationVariable
MessageVariable
CreditCheckVariable
IneligibleApplicationVariable
CustomerApplicationVariable2

© Copyright IBM Corporation 2016

Figure 3-39. Components that are required for Exercise 6 (3 of 7)

WB8601.2

Notes:

The strength of IBM Integration Designer is to use the Service Component Architecture, so other implementations can be placed in separate modules. Because implementations are spread throughout other modules, it maximizes reusability, and necessitates intercommunication and passing variables between modules.

In this exercise, you add the following global variables to the AccountVerification business process. These variables store business objects for manipulation by the process activities:

- The global variable that is named CreditCheckVariable stores a CreditCheckRequest business object.
- The global variable that is named IneligibleApplicationVariable stores an IneligibleApplication business object.
- The global variable that is named CustomerApplicationVariable2 stores a CustomerApplication business object.
- CustomerApplicationVariable and MessageVariable correspond to the type of business objects that form the input and output variables for the AccountVerification process interface.

Components that are required for Exercise 6 (4 of 7)

New components that you create in the lab:

3. Reference partners:

- DetermineApplicationEligibilityPartner
- RecordIneligibleApplicationPartner
- CreditCheckServicePartner
- CreditRiskAssessmentPartner
- RequestMoreDocumentationPartner
- FinalApplicationReviewPartner
- GenerateDeclinePartner
- RecordDeclinedApplicationPartner
- MapToIneligiblePartner

© Copyright IBM Corporation 2016

Figure 3-40. Components that are required for Exercise 6 (4 of 7)

WB8601.2

Notes:

In this portion of the exercise, you add WS-BPEL reference partners to your process. Reference partners represent the service interfaces your WS-BPEL activities call during process execution.

1. Add the DetermineApplicationEligibilityPartner reference partner with the DetermineApplicantEligibility interface.
2. Add the RecordIneligibleApplicationPartner reference partner with the RecordIneligibleApplication interface.
3. Add the CreditCheckServicePartner reference partner with the CreditScoreService interface.
4. Add the CreditRiskAssessmentPartner reference partner with the CreditRiskAssessment interface.
5. Add the RequestMoreDocumentationPartner reference partner with the RequestMoreDocumentation interface.
6. Add the FinalApplicationReviewPartner reference partner with the FinalApplicationReview interface.
7. Add the GenerateDeclinePartner reference partner with the GenerateDecline interface.

8. Add the RecordDeclinedApplicationPartner reference partner with the RecordDeclinedApplication interface.
9. Add the MaptolineligiblePartner reference partner with the Maptolineligible interface.



Components that are required for Exercise 6 (5 of 7)

New components that you create in the lab:

1. Interface partner

- AccountVerification interface partner →

© Copyright IBM Corporation 2016

Figure 3-41. Components that are required for Exercise 6 (5 of 7)

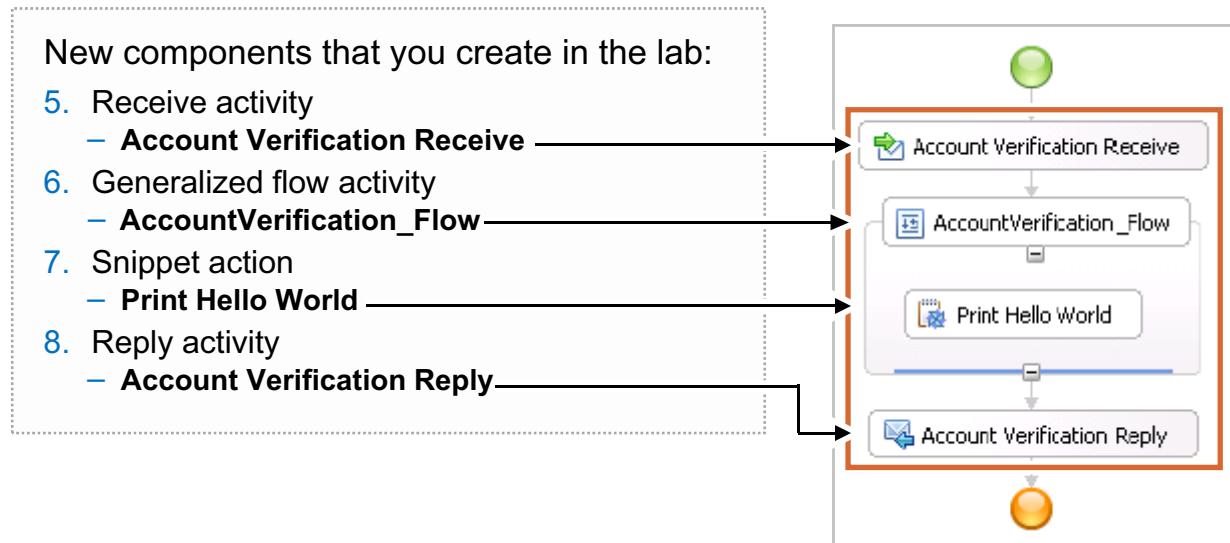
WB8601.2

Notes:

An interface provides the input and output of a component. It is created independent of the internal implementation of the component. The AccountVerification interface dictates how to use the AccountVerification process. It specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions on the AccountVerification process.

The interface partner is a direct link to the interface where the partner is configured. An interface partner is the process interface and exposes operations that external users or services call.

Components that are required for Exercise 6 (6 of 7)



© Copyright IBM Corporation 2016

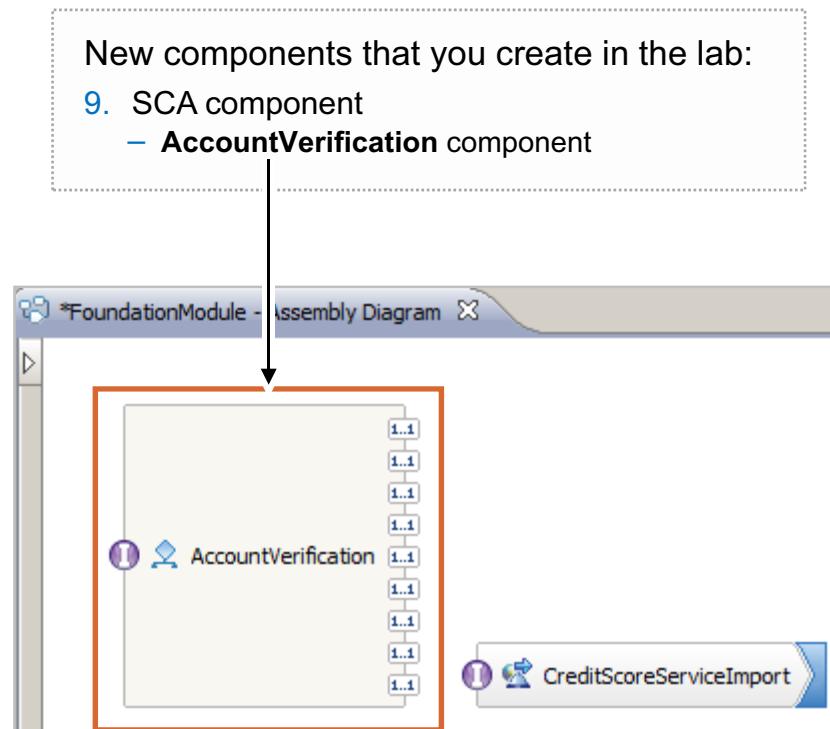
Figure 3-42. Components that are required for Exercise 6 (6 of 7)

WB8601.2

Notes:

In this exercise, you add a generalized flow that is a structured activity that acts as the container for the simple and complex BPEL activities.

Components that are required for Exercise 6 (7 of 7)



© Copyright IBM Corporation 2016

Figure 3-43. Components that are required for Exercise 6 (7 of 7)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 6

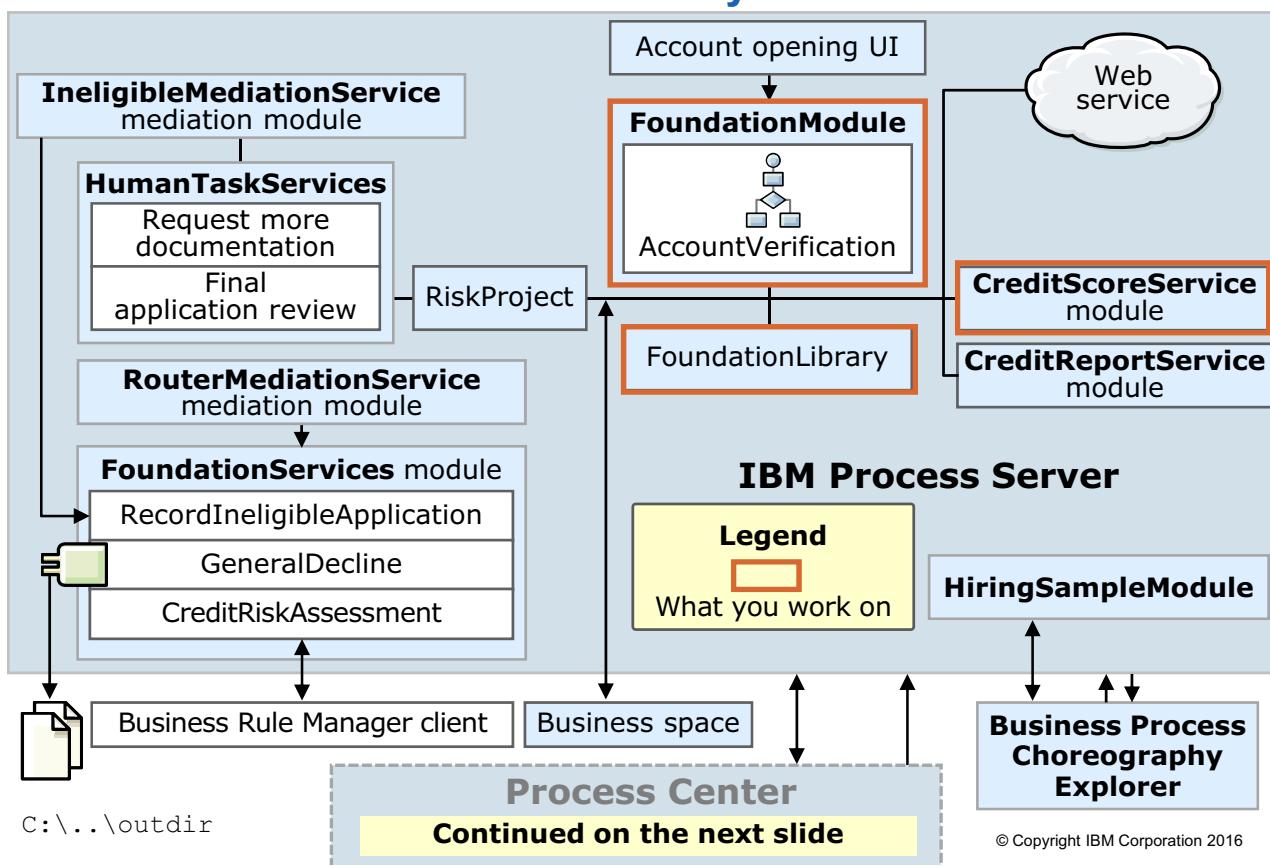


Figure 3-44. Course business scenario: What you work with in Exercise 6

WB8601.2

Notes:

Exercise 7

Creating a business process, part II



© Copyright IBM Corporation 2016

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

10.1

Figure 3-45. Exercise 7

WB8601.2

Notes:



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

- Implement basic and structured BPEL activities in a business process

© Copyright IBM Corporation 2016

Figure 3-46. What are some key tasks that you do in Exercise 7?

WB8601.2

Notes:

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

Prebuilt components that are imported in the lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. AccountVerification**
 - BPEL process that you started in Exercise 6
 - You complete building the process in this exercise

New components that you create in the lab:

- 1. AccountVerification**
 - BPEL process that you started in Exercise 6
 - You complete building the process in this exercise

Note:

The next slide lists all of the subcomponents of the **AccountVerification** process that are built in this lab

© Copyright IBM Corporation 2016

Figure 3-47. Components that are required for Exercise 7 (1 of 2)

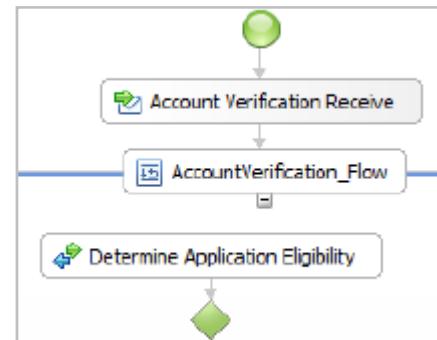
WB8601.2

Notes:

Components that are required for Exercise 7 (2 of 2)

- Activities that are created in the **AccountVerification** process:

1. Invoke activity
 - **Determine Application Eligibility**
 - **Map to Ineligible**
 - **Record Ineligible Application**
 - **Credit Check Service**
 - **Credit Risk Assessment.**
 - **Request More Documentation**
 - **Final Application Review**
 - **Generate Decline**
2. Empty action activity
 - **Map to Credit Check**
 - **Map Credit Checking Result**
3. Assign and loop activity
 - **Create Output** assign activity
 - **While More Documents Required**
while loop activity



© Copyright IBM Corporation 2016

Figure 3-48. Components that are required for Exercise 7 (2 of 2)

WB8601.2

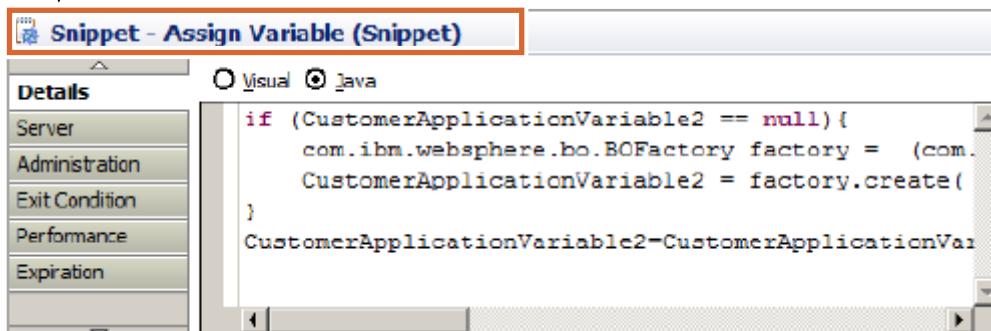
Notes:

1. You create the first activity, **Determine Application Eligibility**, from where the **AccountVerification** process flow starts. The **Determine Application Eligibility** activity examines the customer application to determine whether the customer is eligible for an account. It invokes the **InputCriterion** operation of the **DetermineApplicantEligibility** interface of **DetermineApplicationEligibilityPartner**.
2. The **Map to Ineligible** invoke activity invokes the **InputCriterion** operation of the **MaptoIneligible** interface of **MaptoIneligiblePartner**.
3. The **Record Ineligible Application** activity archives any customer application that is determined to be ineligible. You implement the application that archives the application in a later exercise.
4. The **Map to Credit Check** transforms the data from a customer application business object into a format suitable for the credit score service to understand.
5. The **Credit Check Service** activity invokes the **calculateCreditScore** operation of the **CreditScoreService** interface of **CreditCheckServicePartner**.

6. As soon as the credit score is returned, the data is again transformed. In this case, **Map Credit Checking Result** transforms it back into a customer application business object.
7. The **Credit Risk Assessment** activity examines the credit score that is received and does one of the following three actions:
 - If the credit risk is high, more documentation is requested before final review.
 - If the credit risk is medium, final employee review is requested before approval.
 - If the credit risk is low, the application is approved.
8. The **While More Documents Required** activity continues to request documentation while the comment attribute is equal to None. After an employee reviews the application, the comment attribute is populated and the loop ends.
9. The **Request More Documentation** activity invokes the **InputCriterion** operation of the **RequestMoreDocumentation** interface of **RequestMoreDocumentationPartner**.
10. The **Final Application Review** activity invokes the **InputCriterion** operation of the **FinalApplicationReview** interface of **FinalApplicationReviewPartner**. This activity invokes a human task that allows an employee to review an application.
11. The **Generate Decline** activity invokes the **InputCriterion** operation of the **GenerateDecline** interface of **GenerateDeclinePartner**.
12. The **Record Declined Application** activity invokes the **InputCriterion** operation of the **RecordDeclinedApplication** interface of **RecordDeclinedApplicationPartner**.
13. The **Create Output** activity sets the process output message to: "Application was approved." The process returns the message when creditRisk is HIGH or MED, and the person who reviews the application approves it. This **Create Output** activity sets the process output message to: "Risk was LOW. Application automatically approved." The process returns the message when creditRisk is LOW, and the application flows directly from Credit Risk Assessment to Account Verification Reply.

Code snippets that are created in the AccountVerification process in Exercise 7

- Determine App Eligibility → Map to Ineligible code snippet
- Determine App Eligibility → Map to Credit Check code snippet
- Assign Variable code snippet
- Credit Risk Assessment → Assign Variable code snippet
- Merge Assign code snippet
- Credit Risk Assessment → Final Application Review code snippet
- Final Application Review → Generate Decline code snippet
- Final Application Review → Create Output code snippet
- Credit Risk Assessment → Create Output code snippet



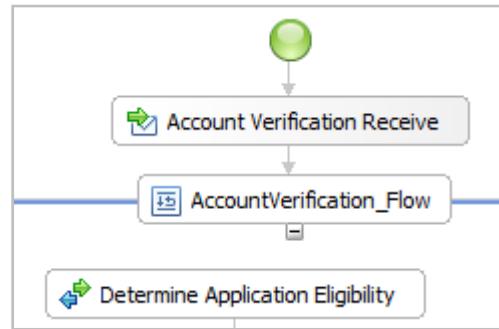
© Copyright IBM Corporation 2016

Figure 3-49. Code snippets that are created in the AccountVerification process in Exercise 7

WB8601.2

Notes:

Complete Account Verification process (1 of 4)



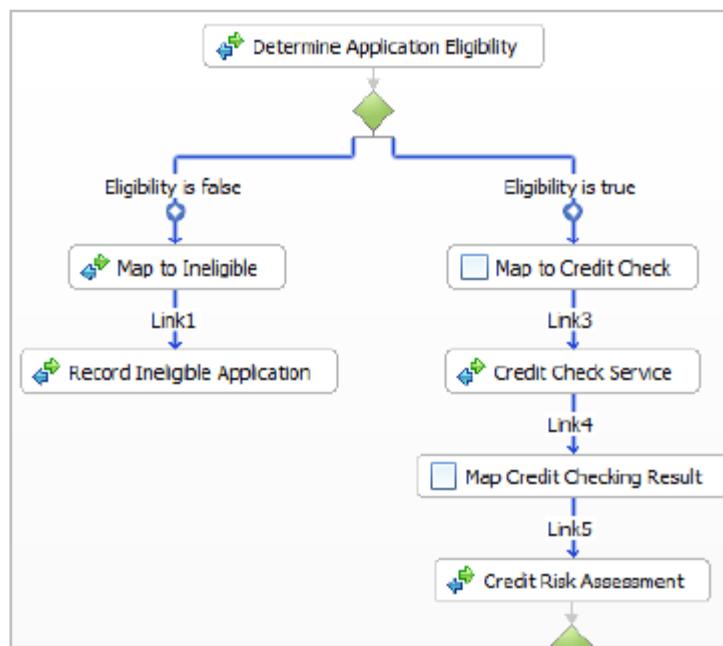
© Copyright IBM Corporation 2016

Figure 3-50. Complete Account verification process (1 of 4)

WB8601.2

Notes:

Complete Account Verification process (2 of 4)



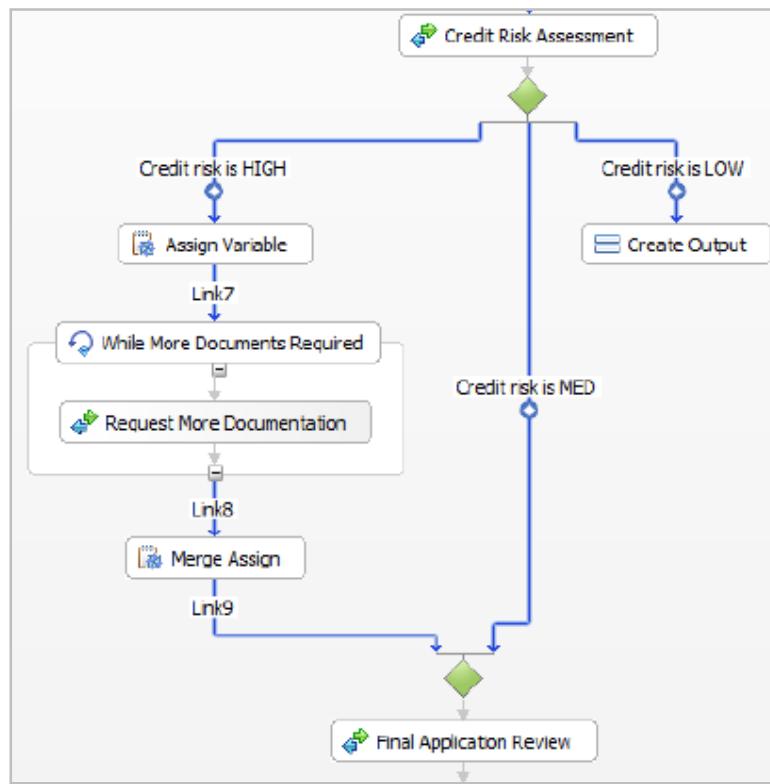
© Copyright IBM Corporation 2016

Figure 3-51. Complete Account verification process (2 of 4)

WB8601.2

Notes:

Complete Account Verification process (3 of 4)



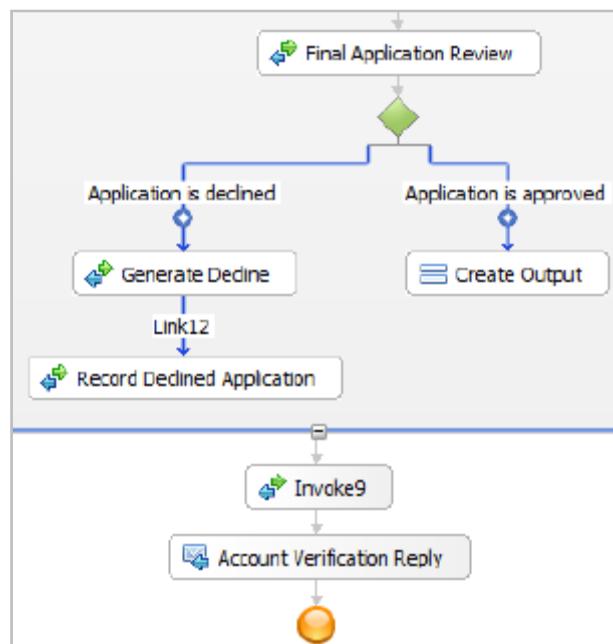
© Copyright IBM Corporation 2016

Figure 3-52. Complete Account verification process (3 of 4)

WB8601.2

Notes:

Complete Account Verification process (4 of 4)



© Copyright IBM Corporation 2016

Figure 3-53. Complete Account verification process (4 of 4)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 7

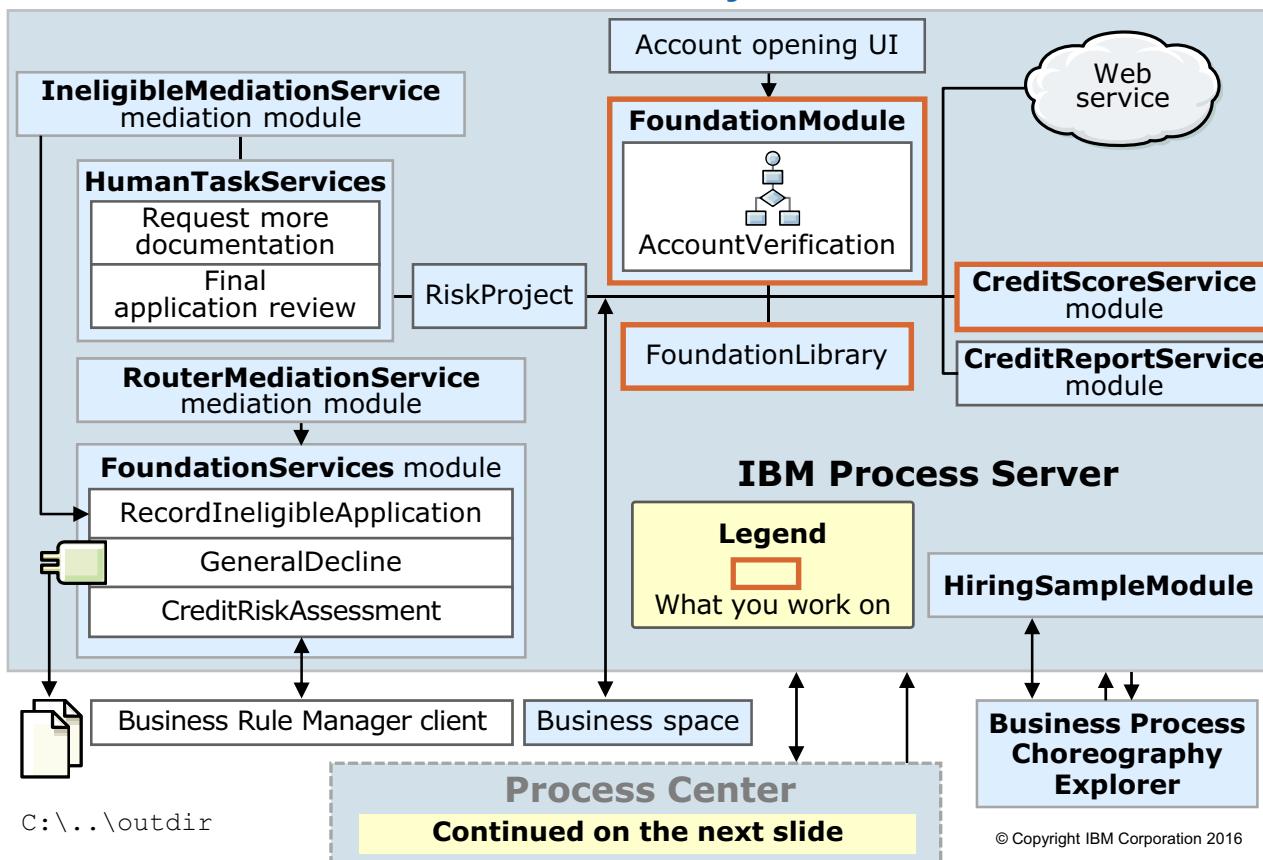


Figure 3-54. Course business scenario: What you work with in Exercise 7

WB8601.2

Notes:

Exercise 8



Creating a business process, part III

© Copyright IBM Corporation 2016

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

10.1

Figure 3-55. Exercise 8

WB8601.2

Notes:



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

- Transform data by using data maps
- Assemble an SCA application that contains a business process
- Test the business process in the IBM Integration Designer test environment

© Copyright IBM Corporation 2016

Figure 3-56. What are some key tasks that you do in Exercise 8?

WB8601.2

Notes:

Components that are required for Exercise 8

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **AccountVerification**
 - BPEL process that you built in Exercises 6 and 7

New components that you create in the lab:

1. **AccountVerification**
 - BPEL process that you completed in Exercise 7
 - You complete implementation of the process in this lab

Note:

The next slide lists all of the subcomponents of the **AccountVerification** process that are built in this lab

© Copyright IBM Corporation 2016

Figure 3-57. Components that are required for Exercise 8

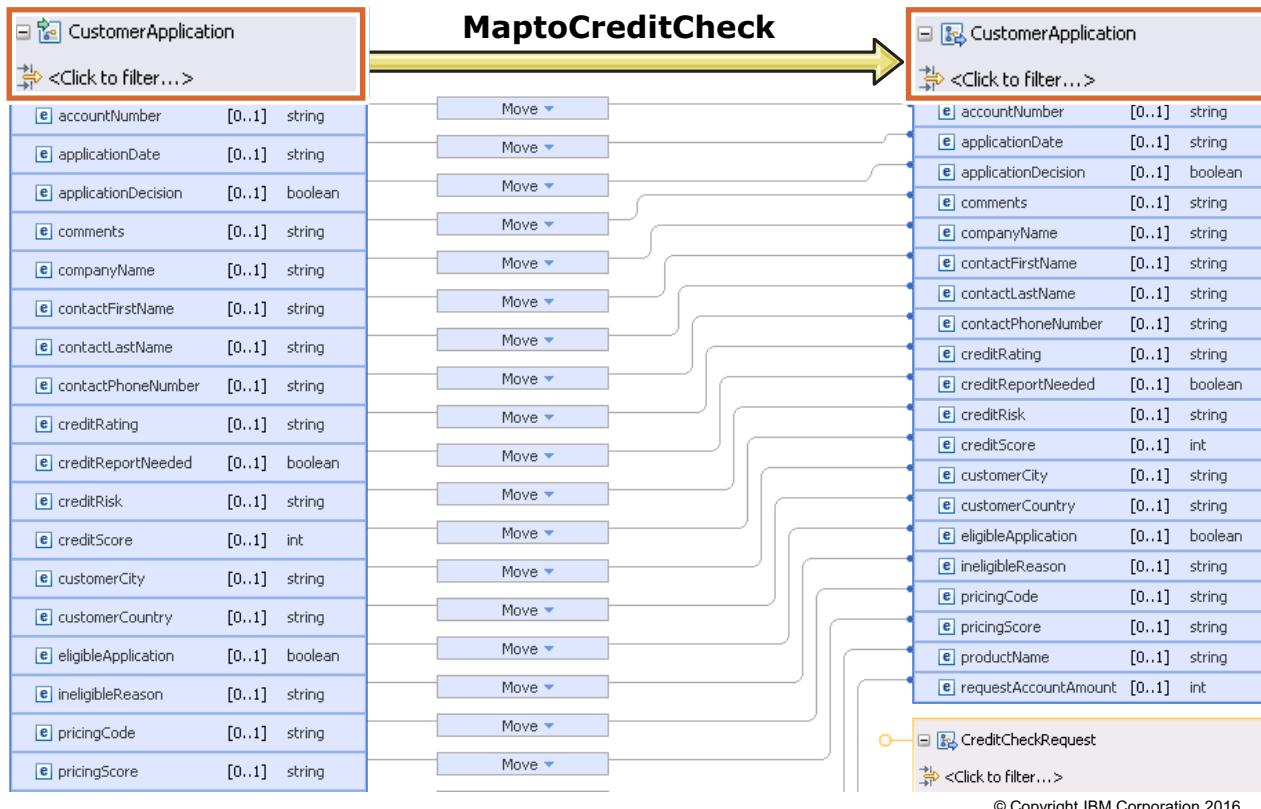
WB8601.2

Notes:

In this exercise, you create data maps to transform business objects between service requesters and providers. You also complete the core business logic for several of the services that are invoked by the AccountVerification business process. When you complete the service logic, you assemble the application that contains the AccountVerification process, and you test it.



Create data map in Exercise 8 (1 of 2)



© Copyright IBM Corporation 2016

Figure 3-58. Create data map in Exercise 8 (1 of 2)

WB8601.2

Notes:

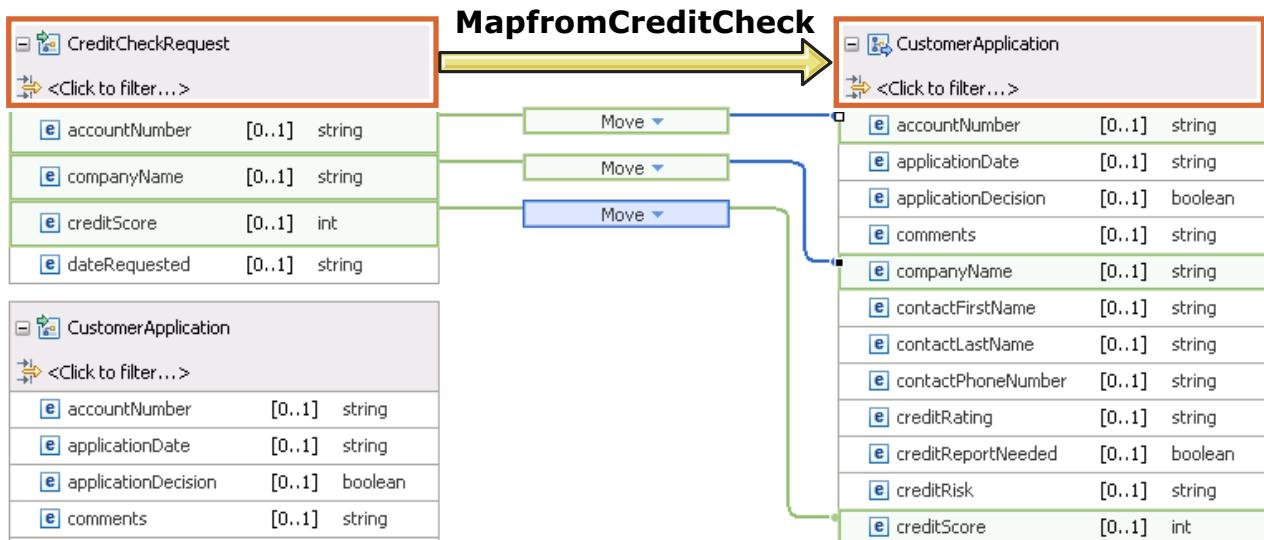
The AccountVerification business process that you implement receives and operates on a CustomerApplication business object input. During process execution, your business process calls an external web service, CreditScoreService, which provides the credit score that is needed to determine the customer's credit risk. Unlike the AccountVerification process (which uses a CustomerApplication business object input), the CreditScoreService interface uses a CreditCheckRequest business object for both the input and the output.

Because of this disparity, before you can invoke the CreditScoreService, you must transform the CustomerApplication business object into a CreditCheckRequest business object input. When the credit score is returned, you must transform the output from a CreditCheckRequest business object into a CustomerApplication business object. Transforming the data before and after service invocations in a business process can be done easily by using the data map activity in IBM Integration Designer.

The MapToCreditCheck data map moves data from the CustomerApplicationVariable input to both a CustomerApplicationVariable output and a CreditCheckVariable output.

You transform the data into both objects so you can preserve the existing data in CustomerApplicationVariable and merge the response data back into it.

Create data map in Exercise 8 (2 of 2)



© Copyright IBM Corporation 2016

Figure 3-59. Create data map in Exercise 8 (2 of 2)

WB8601.2

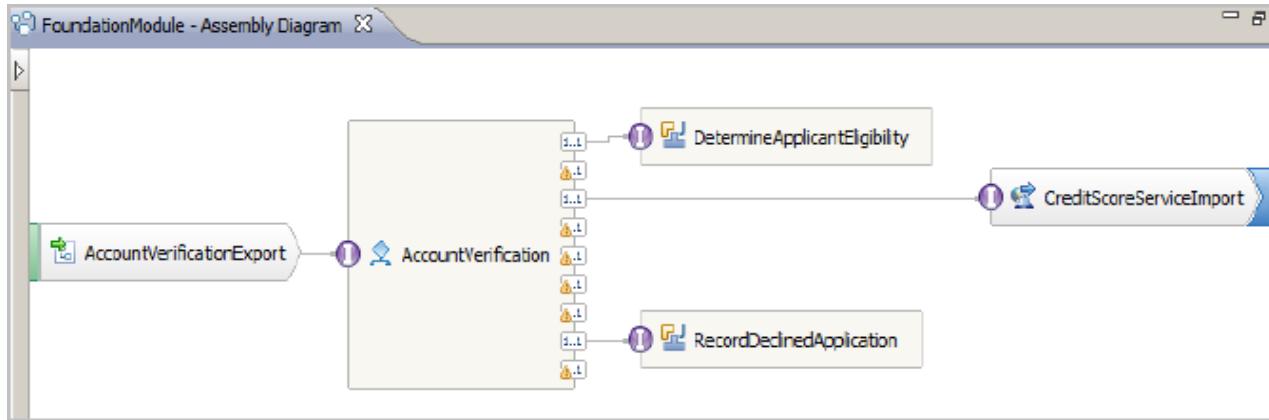
Notes:

You create a **MapfromCreditCheck** data map that merges data from the CreditCheckVariable and CustomerApplicationVariable inputs into a CustomerApplicationVariable output.

WebSphere Education

IBM

Assemble SCA application that contains the business process in Exercise 8



© Copyright IBM Corporation 2016

Figure 3-60. Assemble SCA application that contains the business process in Exercise 8

WB8601.2

Notes:

In the assembly diagram, you wire the SCA components into an integrated application that is deployed to the runtime environment.

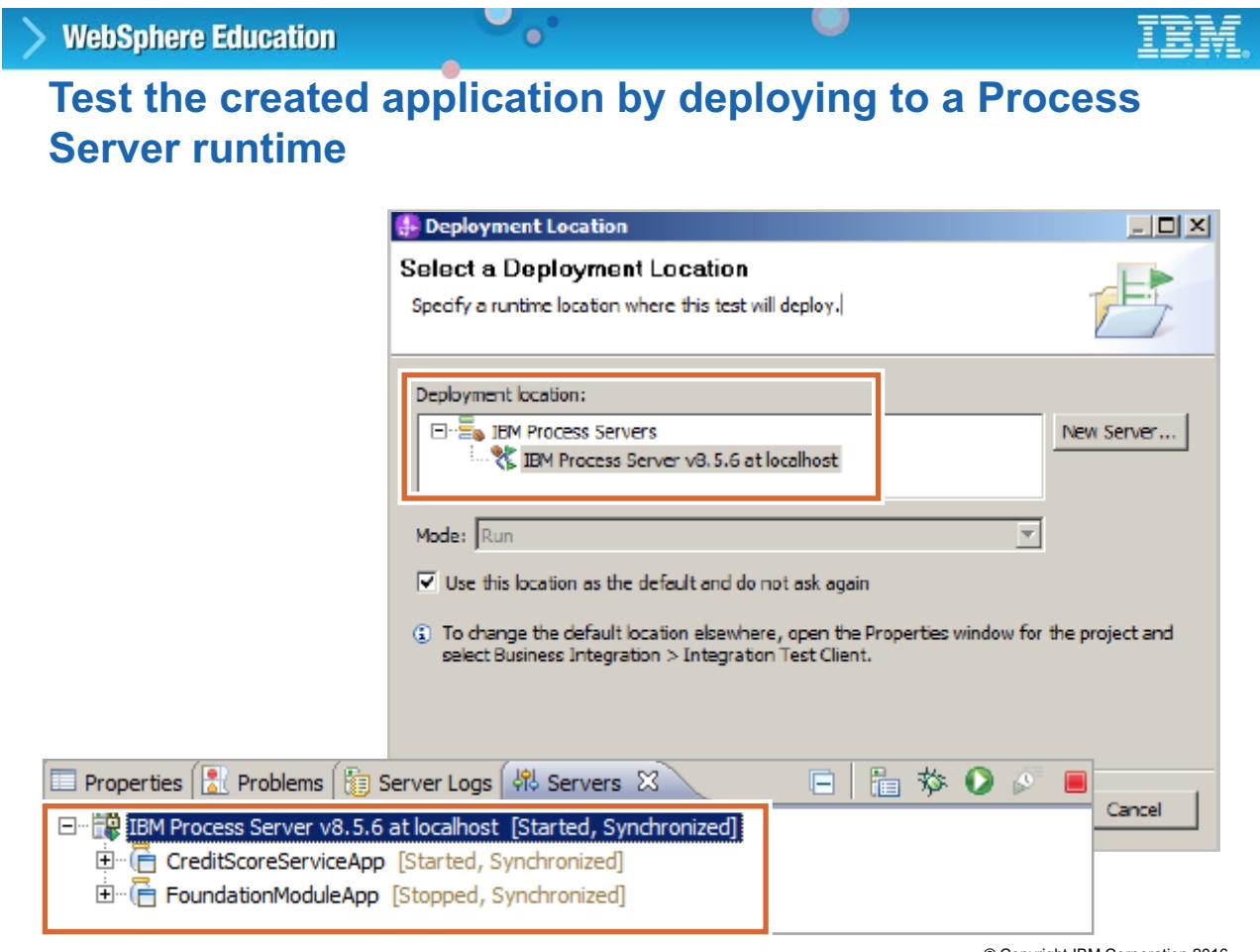


Figure 3-61. Test the created application by deploying to a Process Server runtime

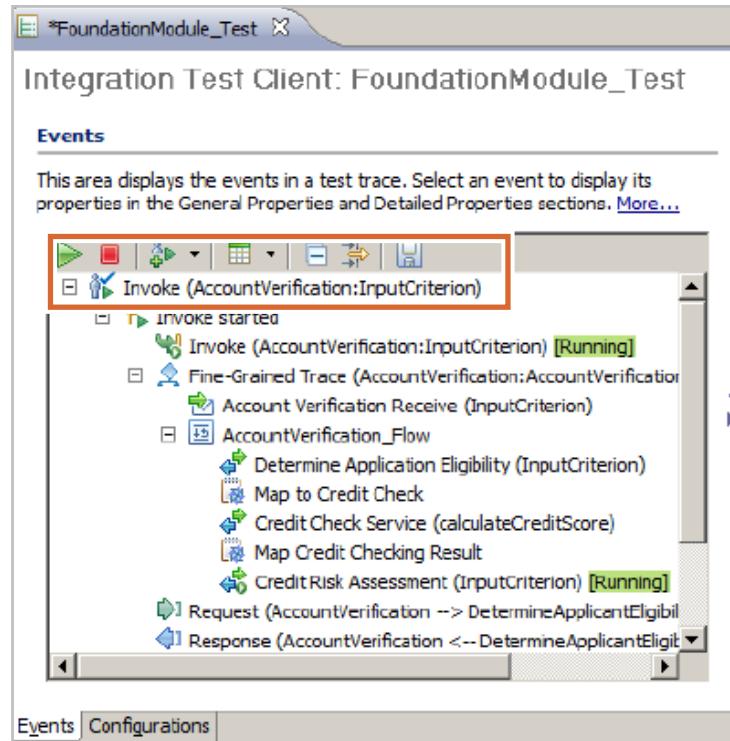
WB8601.2

Notes:



Use the test client in Exercise 8

- Use the Integration Test client to run and test the **AccountVerification** process



© Copyright IBM Corporation 2016

Figure 3-62. Use the test client in Exercise 8

WB8601.2

Notes:

Course business scenario: What you work with in Ex 8

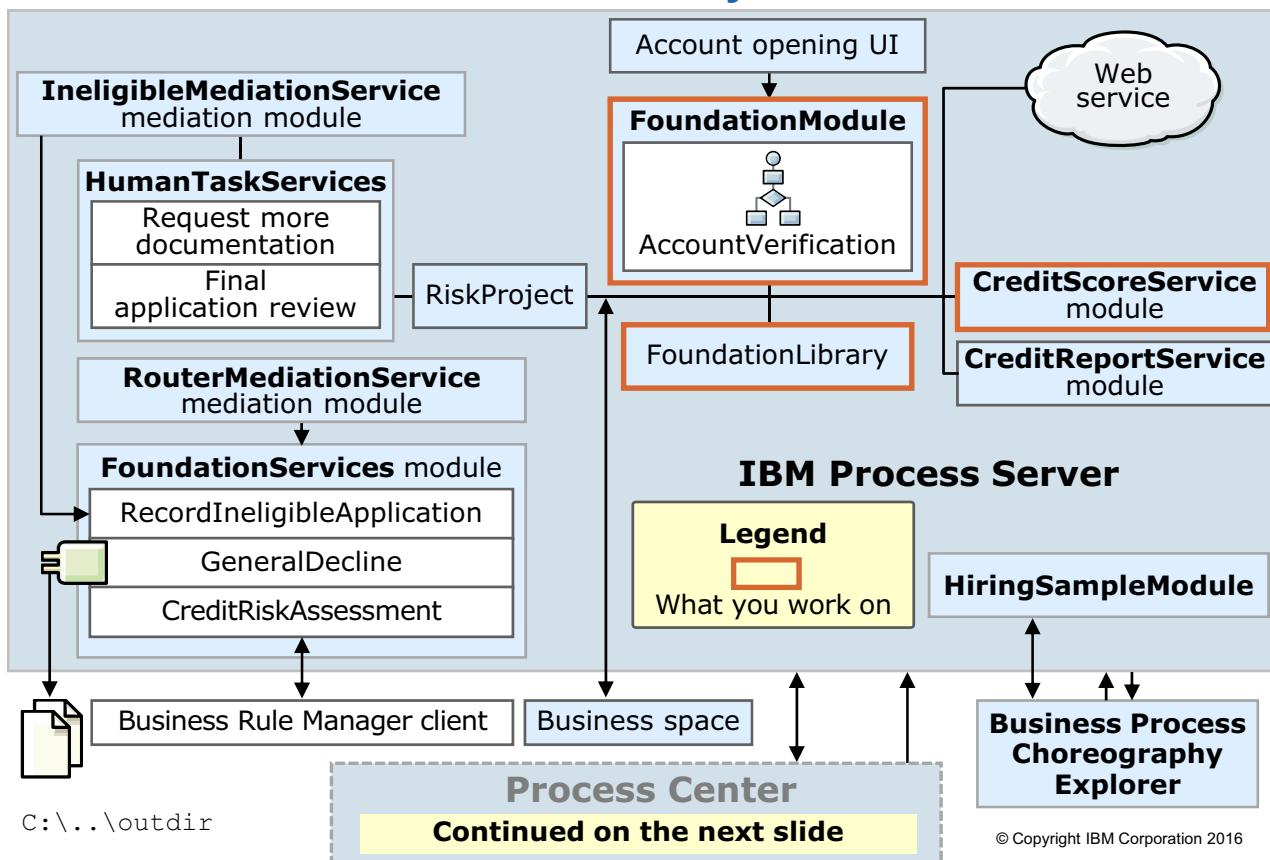


Figure 3-63. Course business scenario: What you work with in Exercise 8

WB8601.2

Notes:

Exercise 9



Creating business rules

© Copyright IBM Corporation 2016

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

10.1

Figure 3-64. Exercise 9

WB8601.2

Notes:

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

- Create rule sets and decision tables that contain business rules
- Create a rule group component
- Test a business rule group in the integration test client

© Copyright IBM Corporation 2016

Figure 3-65. What are some key tasks that you do in Exercise 9?

WB8601.2

Notes:

Components that are required for Exercise 9

Prebuilt components that are imported in the lab:

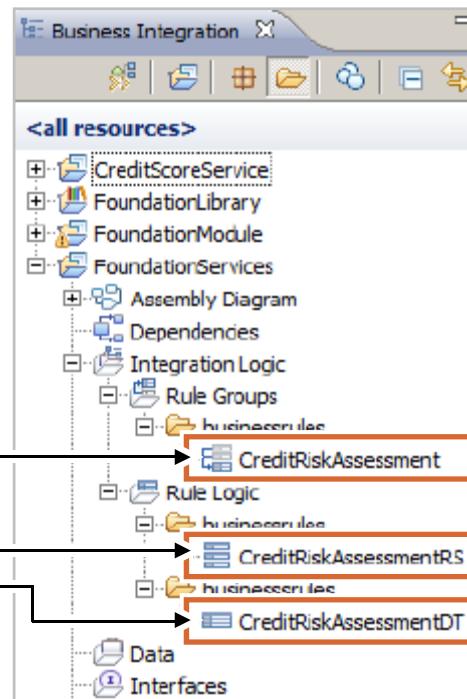
- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. AccountVerification**

- BPEL process that you completed in Exercise 8

New components that you create in the lab:

- 1. FoundationServices** module that contains:

- **CreditRiskAssessment** rule group
- **MapInputToOutput** rule
- **CreditRiskAssessmentRS** rule set
- **CreditRiskAssessmentDT** decision table



© Copyright IBM Corporation 2016

Figure 3-66. Components that are required for Exercise 9

WB8601.2

Notes:

In this exercise, you create rule sets and decision tables that contain business rules. You create a rule group component, incorporate the rule group component in an assembly diagram, and test it. Finally, you use the Business Rules Manager web client to interact with business rules at run time.

To determine the credit risk, a series of business rules are used to evaluate the credit score that the credit score service returns.

In this exercise, you create the business rules that the credit risk assessment service uses. If the creditScore value is less than 4, then the creditRisk is HIGH. If the creditScore value is in the range of 4 – 7, then the creditRisk is MED (medium). If the creditScore value is in the range of 8 – 11, then the creditRisk is LOW. Customer applications that are HIGH risk require more documentation and are subject to more review. Customer applications that are MED risk require more review but not more documentation. Customer applications that are LOW risk are automatically approved.

You create a **CreditRiskAssessment** business rule group in the **FoundationServices** module that uses the **CreditRiskAssessment** interface. You also create a **CreditRiskAssessmentDT** decision table in the rule group.

You then create an action rule that is named **MapInputToOutput**, which copies the data from the input business object and assigns it to the output business object.



Create rule group in Exercise 9

Rule Set

Name	CreditRiskAssessmentRS
------	------------------------

Interface

Scheduled Rule Logic

Default Rule Logic	CreditRiskAssessmentRS
Start Date	Nov 3, 2015 12:00 AM
End Date	Nov 3, 2016 12:00 AM
Rule Logic	CreditRiskAssessmentRS

Variables

Rules

Name	MapOutput
Presentation	
Action	Output =copyBO(Input)

Name	RiskHIGH
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is HIGH

Name	RiskMED
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than <input type="text" value="3"/> and less than <input type="text" value="8"/> then the credit risk is MED

Name	RiskLOW
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than <input type="text" value="7"/> and less than <input type="text" value="12"/> then the credit risk is LOW

- Test the created application by deploying to a Process Server runtime

© Copyright IBM Corporation 2016

Figure 3-67. Create rule group in Exercise 9

WB8601.2

Notes:

Course business scenario: What you work with in Ex 9

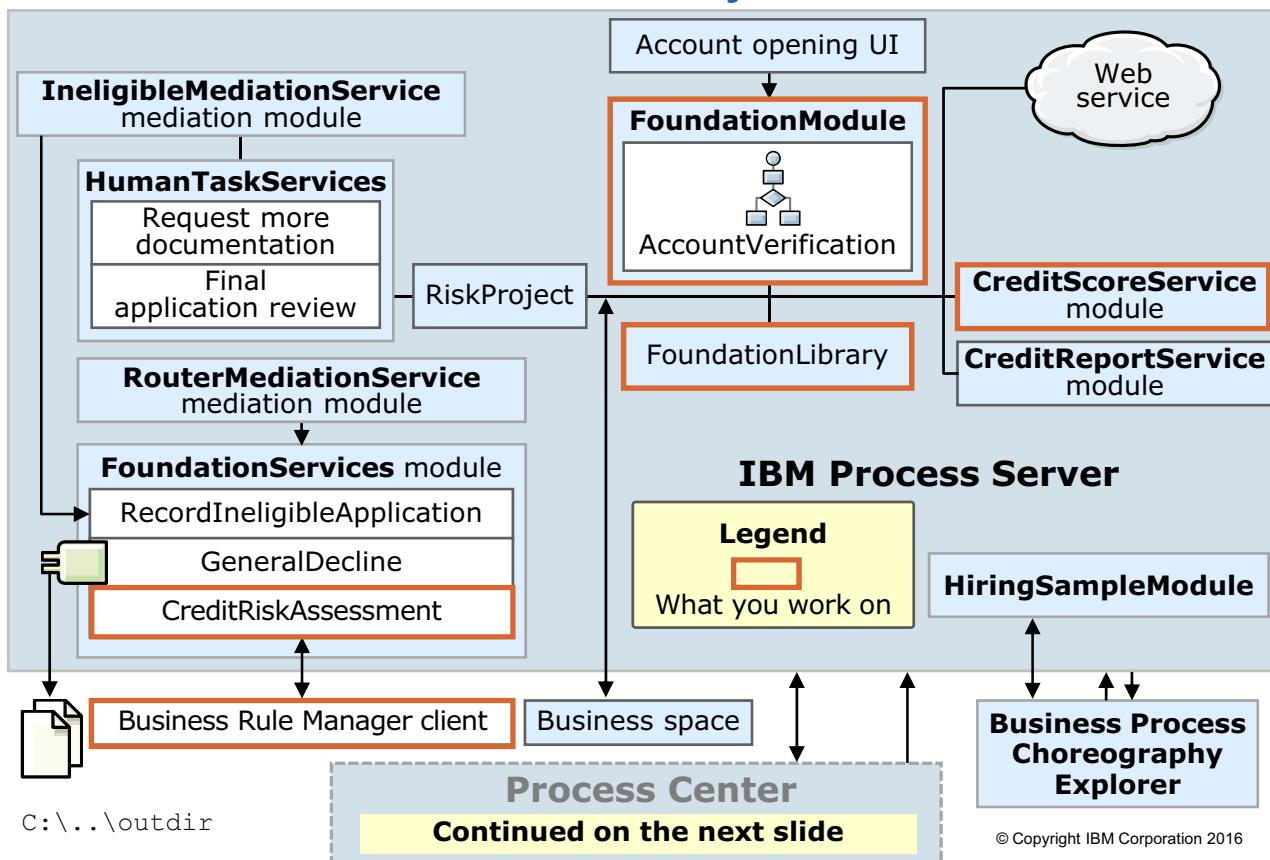


Figure 3-68. Course business scenario: What you work with in Exercise 9

WB8601.2

Notes:

Exercise 10



Implementing WebSphere (JCA) adapters

© Copyright IBM Corporation 2016

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

10.1

Figure 3-69. Exercise 10

WB8601.2

Notes:



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

- Configure the WebSphere Adapter for Flat Files and test with account verification process

© Copyright IBM Corporation 2016

Figure 3-70. What are some key tasks that you do in Exercise 10?

WB8601.2

Notes:

IBM WebSphere Adapters make it possible for Java Platform, Enterprise Edition (Java EE) components, such as applications, to communicate with enterprise information system (EIS) resources. An EIS is the information infrastructure for an enterprise (for example, an enterprise resource planning system). A WebSphere adapter acts as an intermediary between the Java EE component and the EIS. This way, it is not necessary for the Java EE component to understand the low-level API or data structures of the EIS.

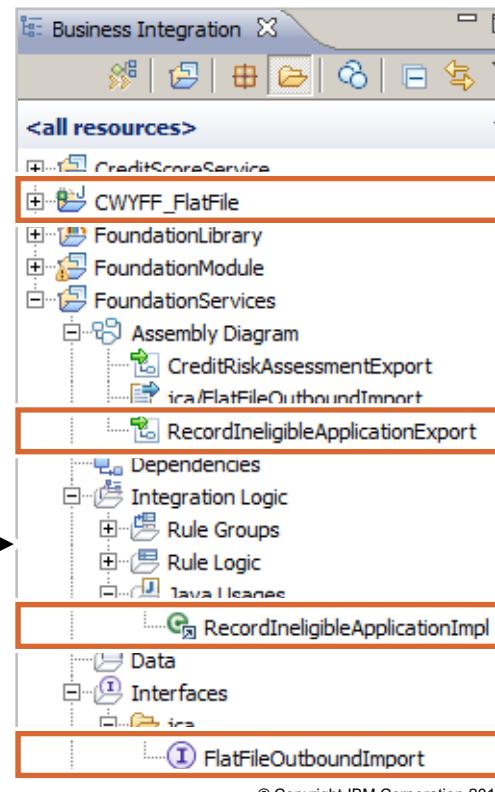
Components that are required for Exercise 10 (1 of 2)

Prebuilt components that are imported in the lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. FoundationServices** that you built in Exercise 9

New components that you create in this lab:

- 1. CWYFF_FlatFile** connector project
- 2. FlatFileOutboundImport** import component
- 3. RecordIneligibleApplicationExport** export component
- 4. RecordIneligibleApplicationImpl.java** component



© Copyright IBM Corporation 2016

Figure 3-71. Components that are required for Exercise 10 (1 of 2)

WB8601.2

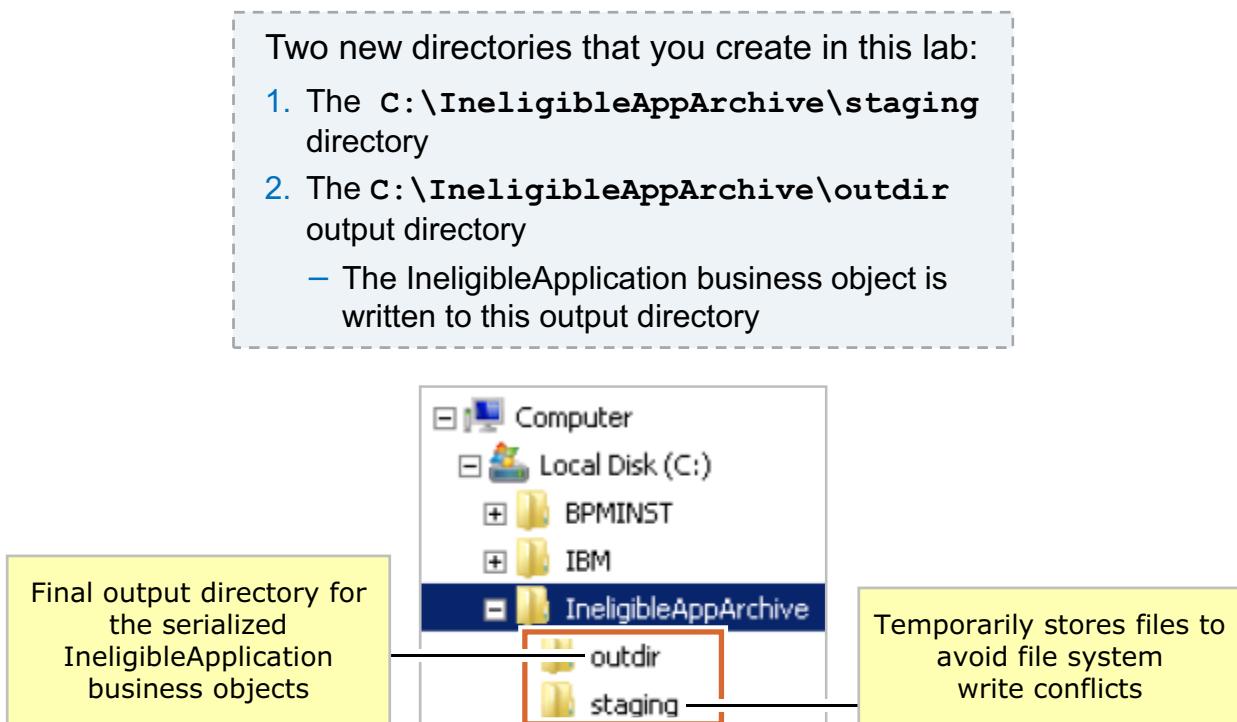
Notes:

IBM WebSphere Adapter for Flat Files facilitates the exchange of business data in the form of delimited records between file systems and Java EE applications. The adapter supports inbound and outbound operations and the use of business objects, business components, and business services.

In this exercise, you implement a service to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files. The AccountVerification process invokes the service when the eligibleApplication attribute is set to false.

See the process application model that was created previously. In this exercise, you implement the Record Ineligible Application activity. The purpose of this activity is to record ineligible applications as XML files on the file system. This service uses WebSphere Adapter for Flat Files.

Components that are required for Exercise 10 (2 of 2)



© Copyright IBM Corporation 2016

Figure 3-72. Components that are required for Exercise 10 (2 of 2)

WB8601.2

Notes:

You run the External Service wizard to configure the WebSphere Adapter for Flat Files. The adapter uses the directories that you examined previously. The adapter code is deployed inside the FoundationServices module, and a `FlatFileOutboundImport` component is created on the FoundationServices assembly diagram. The `FlatFileOutboundImport` component invokes the `recordIneligibleApplication` operation to write an `IneligibleApplication.txt` file to `C:\IneligibleAppArchive\outdir`.

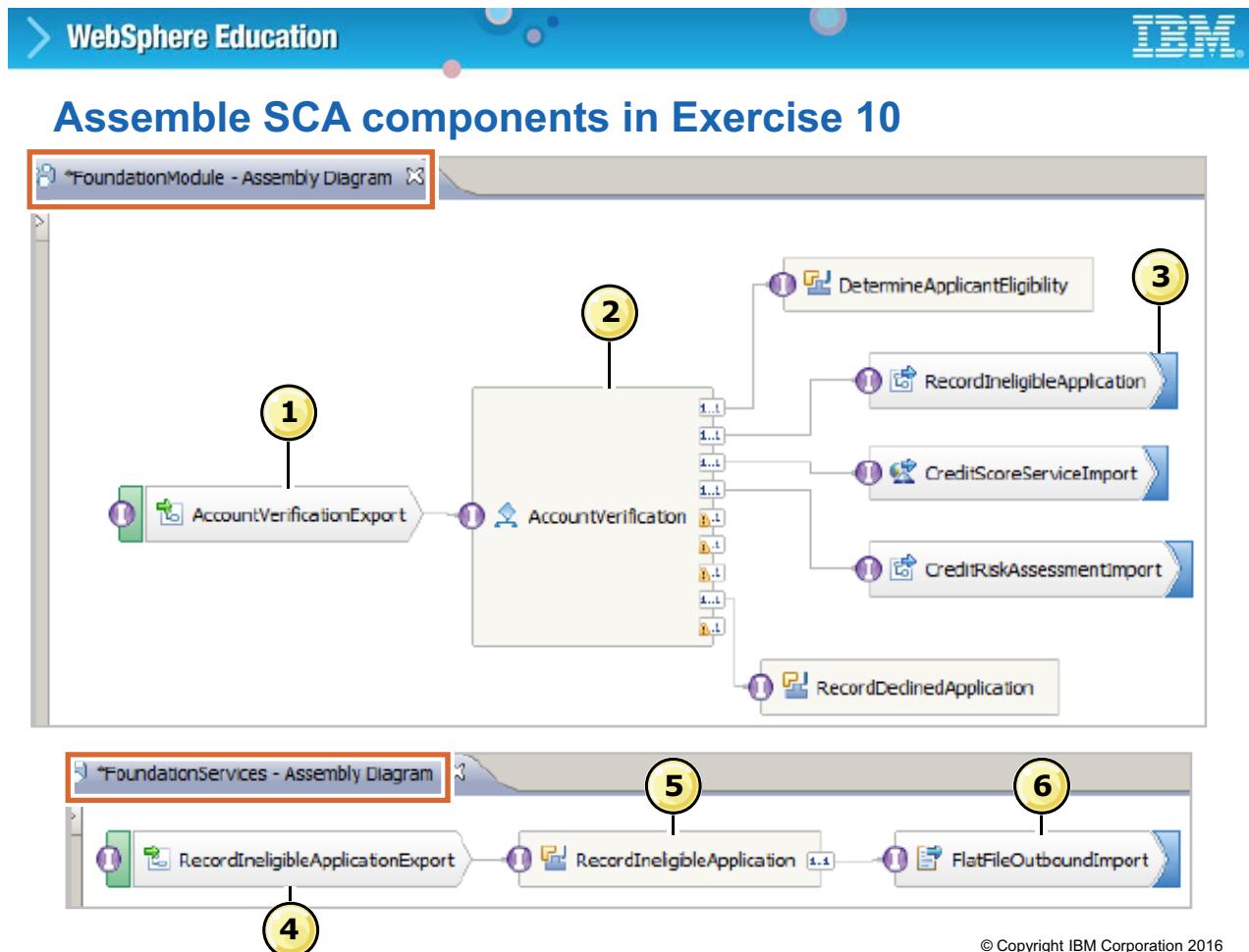


Figure 3-73. Assemble SCA components in Exercise 10

WB8601.2

Notes:

Finally, after creating the flat file adapter component, you assemble the SCA components in IBM Integration Designer and then test the application.

The following steps are illustrated in the diagram:

- **Step 1:** The **AccountVerificationExport** component exposes the **AccountVerification** business process.
- **Step 2:** When the application is ineligible, the **AccountVerification** process needs to record the ineligible application in the database and terminate the process. It calls the **RecordIneligibleApplication** import component.
- **Step 3:** The **RecordIneligibleApplication** import component is used to call the application or service outside FoundationModule. In this scenario, it calls the **RecordIneligibleExport** component in the FoundationServices module.
- **Step 4:** The **RecordIneligibleExport** component calls the **RecordIneligibleApplication** Java component.
- **Step 5:** The **RecordIneligibleApplication** component is a Java component that creates the output message.

"Account verification recorded this application as ineligible for the customer <company name>" is going to be recorded to the system. In this scenario, it calls the FlatFileOutboundImport component.

- **Step 6:** The FlatFileOutboundImport component writes the output message to the file system. In this scenario, it writes to a text file in the `C:\IneligibleAppArchive\outdir` output directory.

Course business scenario: What you work with in Ex 10

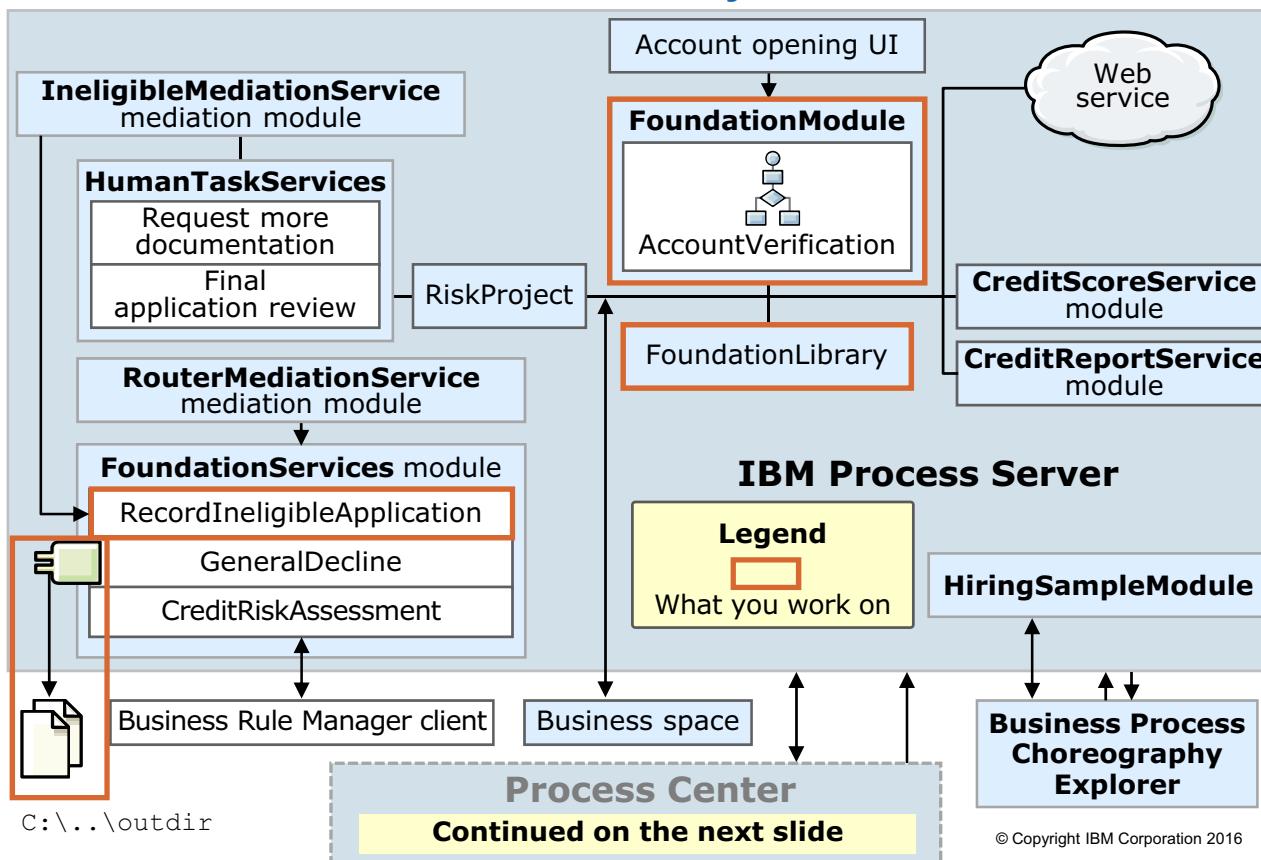


Figure 3-74. Course business scenario: What you work with in Exercise 10

WB8601.2

Notes:

Exercise 11



Creating mediation services, part I

© Copyright IBM Corporation 2016

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

10.1

Figure 3-75. Exercise 11

WB8601.2

Notes:



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

- Create a mediation module and an XSL Transformation primitive
- Create an XML data map

© Copyright IBM Corporation 2016

Figure 3-76. What are some key tasks that you do in Exercise 11?

WB8601.2

Notes:

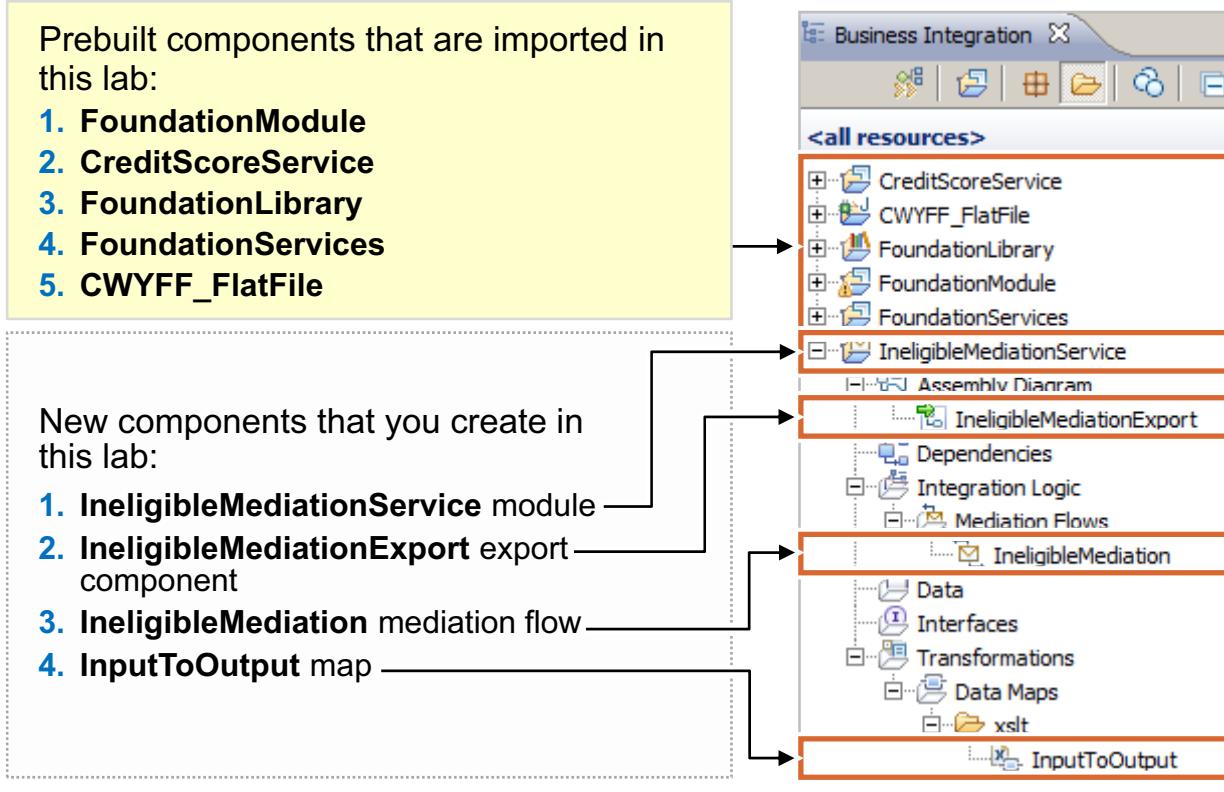
Components that are required for Exercise 11

Prebuilt components that are imported in this lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **FoundationServices**
5. **CWYFF_FlatFile**

New components that you create in this lab:

1. **IneligibleMediationService** module
2. **IneligibleMediationExport** export component
3. **IneligibleMediation** mediation flow
4. **InputToOutput** map



© Copyright IBM Corporation 2016

Figure 3-77. Components that are required for Exercise 11

WB8601.2

Notes:

In this exercise, you implement a mediation module that is named **IneligibleMediationService**, which transforms messages between the **AccountVerification** process and the **RecordIneligibleApplication** service (which archives ineligible applications). Suppose the data input for the **RecordIneligibleApplication** service is different from the **CustomerApplication** input that the **AccountVerification** process uses. Under this scenario, the **RecordIneligibleApplication** component does not read the data and no output message is sent. To fix this problem, you must transform the messages that are passed between the services. This situation is a common challenge that your application must be prepared to resolve, and IBM Integration Designer solves it by creating a transformation between the two messages.

You implement a Mapping primitive inside a mediation flow to transform the messages by using a data map. After implementing the transformation in the mediation flow, you assemble the application by wiring the mediation service to the **AccountVerification** process.

Finally, after creating the flat file adapter component, you assemble the SCA components in IBM Integration Designer and then test the application.



Assemble SCA components in Exercise 11

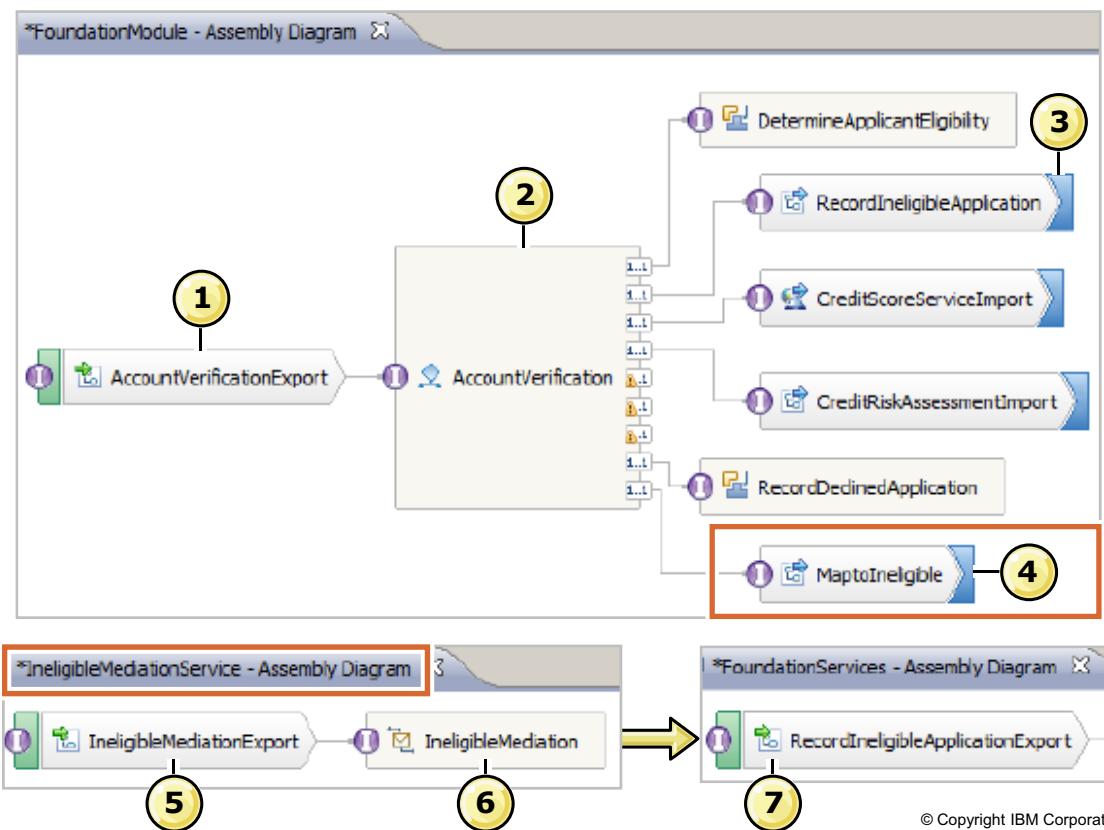


Figure 3-78. Assemble SCA components in Exercise 11

WB8601.2

Notes:

The following steps are illustrated in the diagram:

- **Step 1:** The AccountVerificationExport component exposes the AccountVerification business process.
- **Step 2:** When the application is ineligible, the AccountVerification process needs to record the ineligible application in the database and terminate the process. It calls the RecordIneligibleApplication import component.
- **Step 3:** The RecordIneligibleApplication import component is used to call the application or service outside FoundationModule. In this scenario, it calls the RecordIneligibleExport component in the FoundationServices module.
- **Step 4:** The MaptoIneligible import invokes the IneligibleMediationExport component in the IneligibleMediationService mediation module.
- **Step 5:** IneligibleMediationExport exposes the mediation flow by calling the IneligibleMediation component.

- **Step 6:** The IneligibleMediation component transforms a CustomerApplication into an IneligibleApplication before invoking the RecordIneligibleApplicationExport component to archive the application.
- **Step 7:** The RecordIneligibleExport component calls the RecordIneligibleApplication Java component. The RecordIneligibleApplication component is a Java component that creates the output message.

"Account verification recorded this application as ineligible for the customer <company name>" is going to be recorded to the system. In this scenario, it calls the FlatFileOutboundImport component. The FlatFileOutboundImport component writes the output message to the file system. In this scenario, it writes to a text file in the C:\IneligibleAppArchive\outdir output directory.

Course business scenario: What you work with in Ex 11

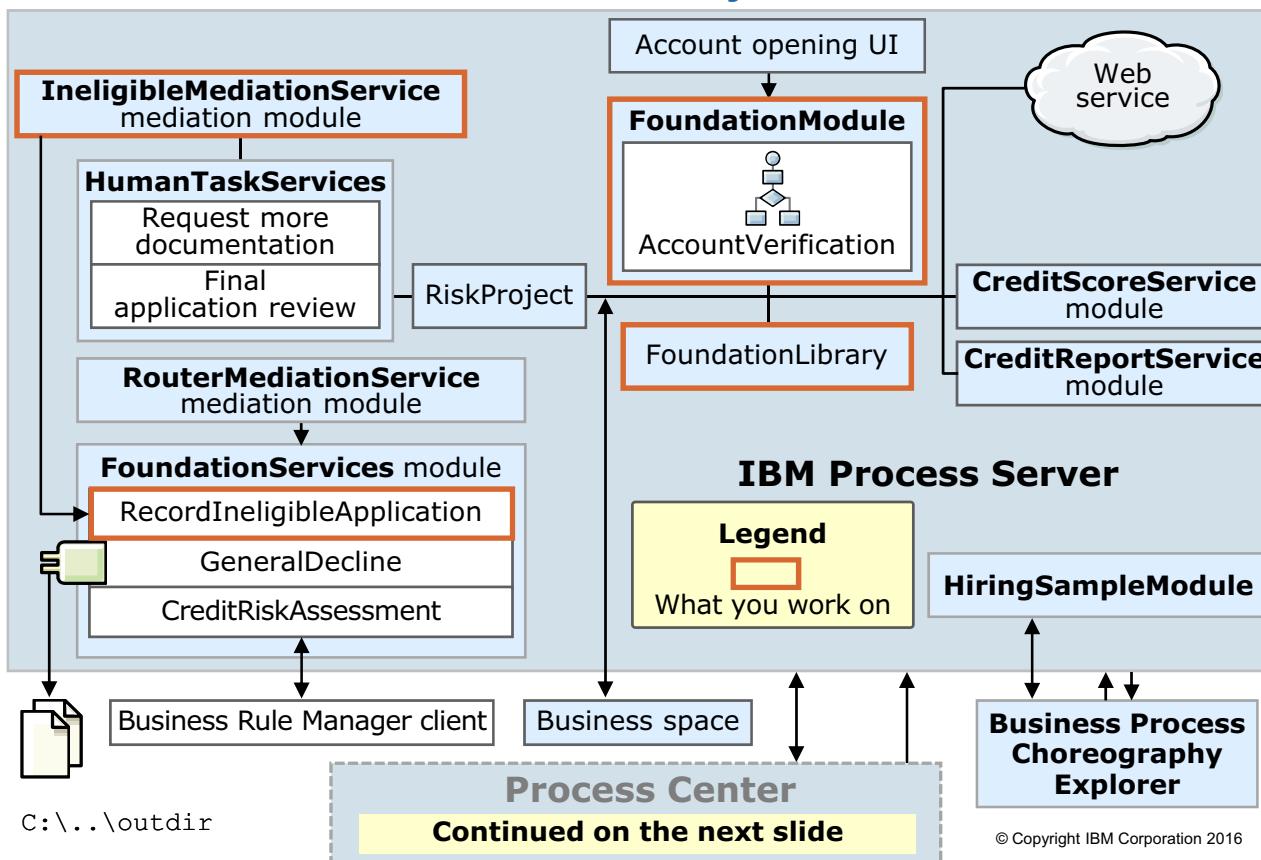


Figure 3-79. Course business scenario: What you work with in Exercise 11

WB8601.2

Notes:

Exercise 12



Creating mediation services, part II

© Copyright IBM Corporation 2016

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

10.1

Figure 3-80. Exercise 12

WB8601.2

Notes:



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

- Create and test a mediation module that contains a Message Filter mediation primitive and an XSL Transformation primitive
- Create an XML data map

© Copyright IBM Corporation 2016

Figure 3-81. What are some key tasks that you do in Exercise 12?

WB8601.2

Notes:

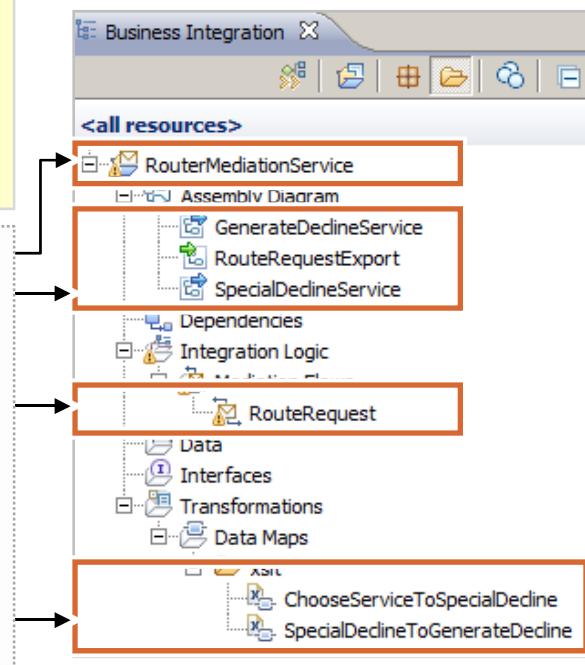
Components that are required for Exercise 12 (1 of 2)

Prebuilt components that are imported in this lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **FoundationServices**
5. **CWYFF_FlatFile**
6. **IneligibleMediationService**

New components that you create in this lab:

1. **RouterMediationService** module
2. **GenerateDeclineService** import
3. **RouteRequestExport** export
4. **SpecialDeclineService** import
5. **RouteRequest** mediation flow
6. **ChooseServiceToSpecialDecline** map
7. **SpecialDeclineToGenerateDecline** map



© Copyright IBM Corporation 2016

Figure 3-82. Components that are required for Exercise 12 (1 of 2)

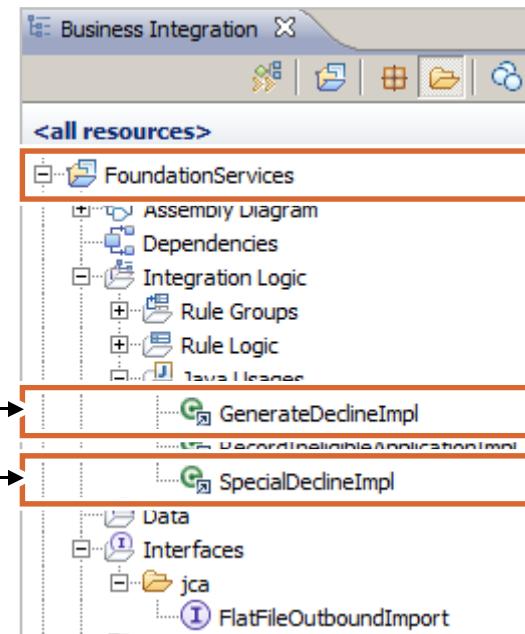
WB8601.2

Notes:

Components that are required for Exercise 12 (2 of 2)

Java components that you add in this lab:

1. **GenerateDeclineImpl** Java implementation
2. **SpecialDeclineImpl** Java implementation



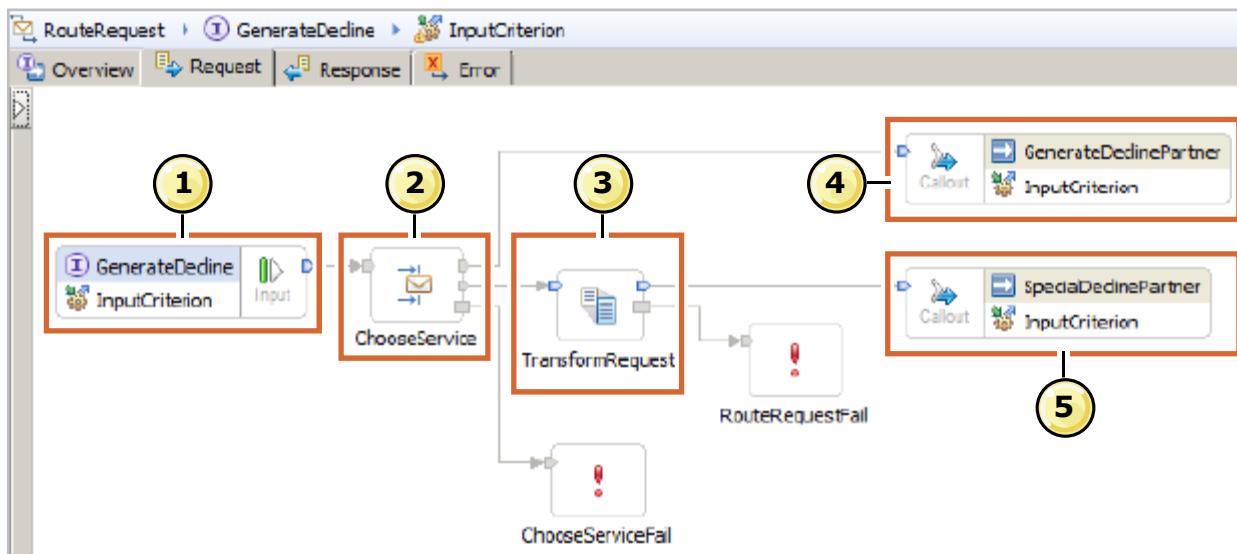
© Copyright IBM Corporation 2016

Figure 3-83. Components that are required for Exercise 12 (2 of 2)

WB8601.2

Notes:

Create the RouteRequest mediation flow in Exercise 12



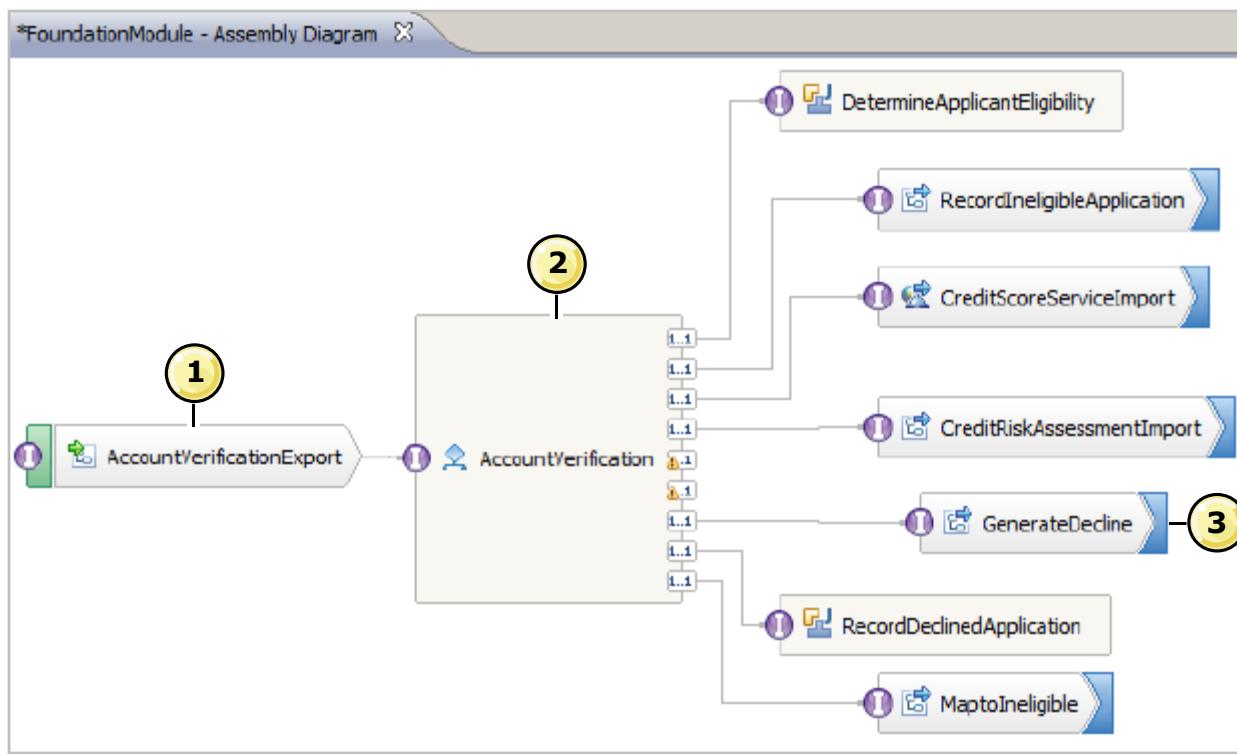
© Copyright IBM Corporation 2016

Figure 3-84. Create the RouteRequest mediation flow in Exercise 12

WB8601.2

Notes:

Assemble SCA components in Exercise 12 (1 of 2)



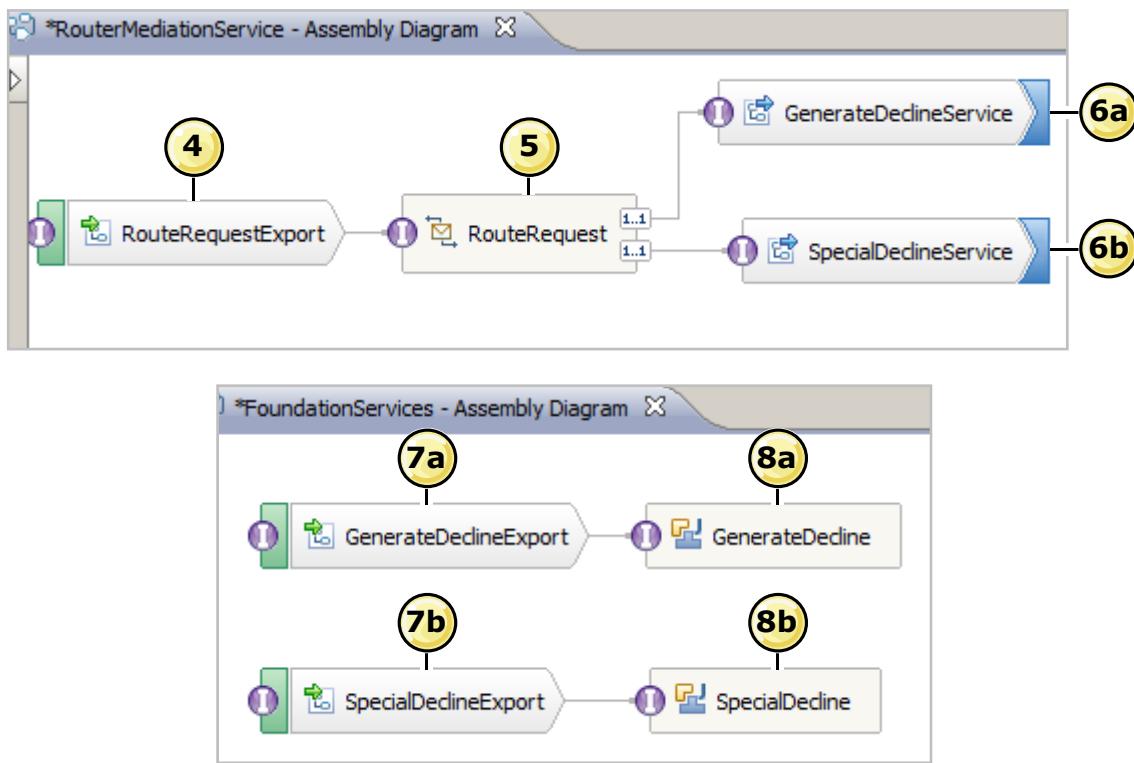
© Copyright IBM Corporation 2016

Figure 3-85. Assemble SCA components in Exercise 12 (1 of 2)

WB8601.2

Notes:

Assemble SCA components in Exercise 12 (2 of 2)



© Copyright IBM Corporation 2016

Figure 3-86. Assemble SCA components in Exercise 12 (2 of 2)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 12

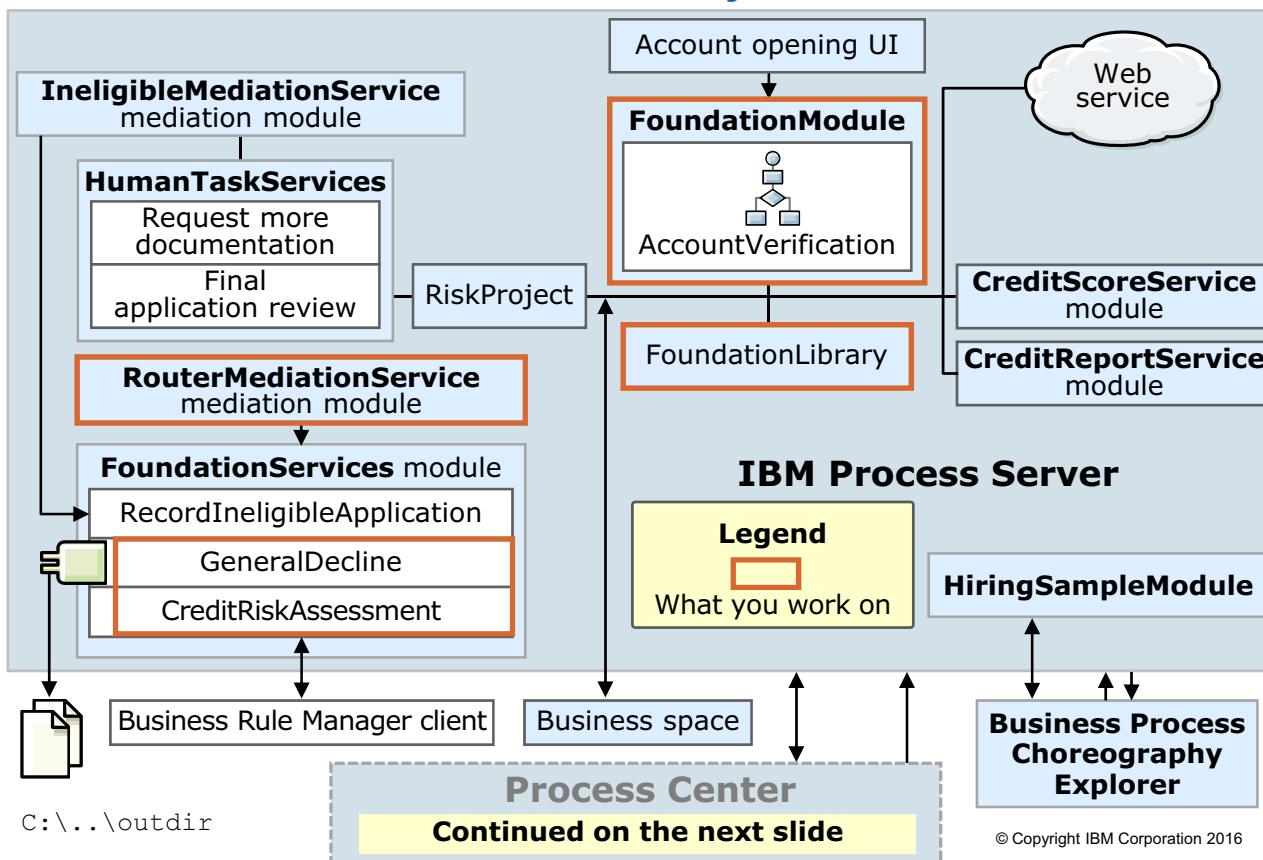


Figure 3-87. Course business scenario: What you work with in Exercise 12

WB8601.2

Notes:

Exercise 13



Exploring Business Space

© Copyright IBM Corporation 2016

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

10.1

Figure 3-88. Exercise 13

WB8601.2

Notes:



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

- Create a business space to work with tasks
- Create a page and display content by using widgets and Business Space templates

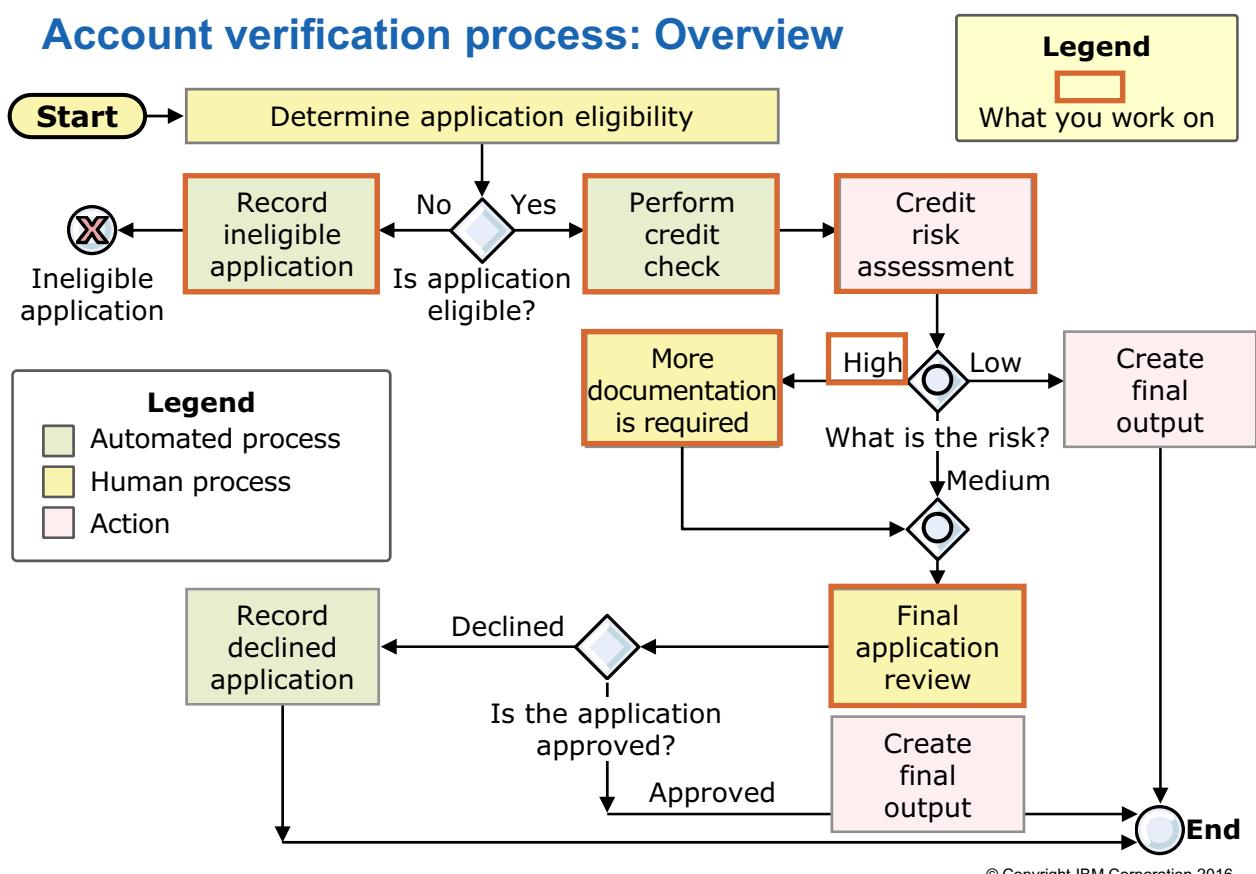
© Copyright IBM Corporation 2016

Figure 3-89. What are some key tasks that you do in Exercise 13?

WB8601.2

Notes:

Account verification process: Overview



© Copyright IBM Corporation 2016

Figure 3-90. Account verification process: Overview

WB8601.2

Notes:

The screenshot shows the WebSphere Education interface. At the top, there's a navigation bar with a logo and the text 'WebSphere Education'. On the right side of the interface, the IBM logo is visible. The main content area has a title 'Create business space in Exercise 13'. Below the title is a 'Tasks' section. A specific task, 'Final Application Review', is highlighted with an orange border. A context menu is open over this task, with 'Accept' being the selected option. The menu also includes 'Edit', 'View', 'Transfer', 'Return', 'Delete', 'Escalate', and 'Postpone'. Below the tasks, there's a 'MyBusinessSpace' dashboard with a sidebar showing 'All Widgets (42)' and a main area displaying various business-related widgets like 'Business Calendars', 'Business Categories', 'Business Rules', 'Create Tasks', etc.

© Copyright IBM Corporation 2016

Figure 3-91. Create business space in Exercise 13

WB8601.2

Notes:

Course business scenario: What you work with in Ex 13

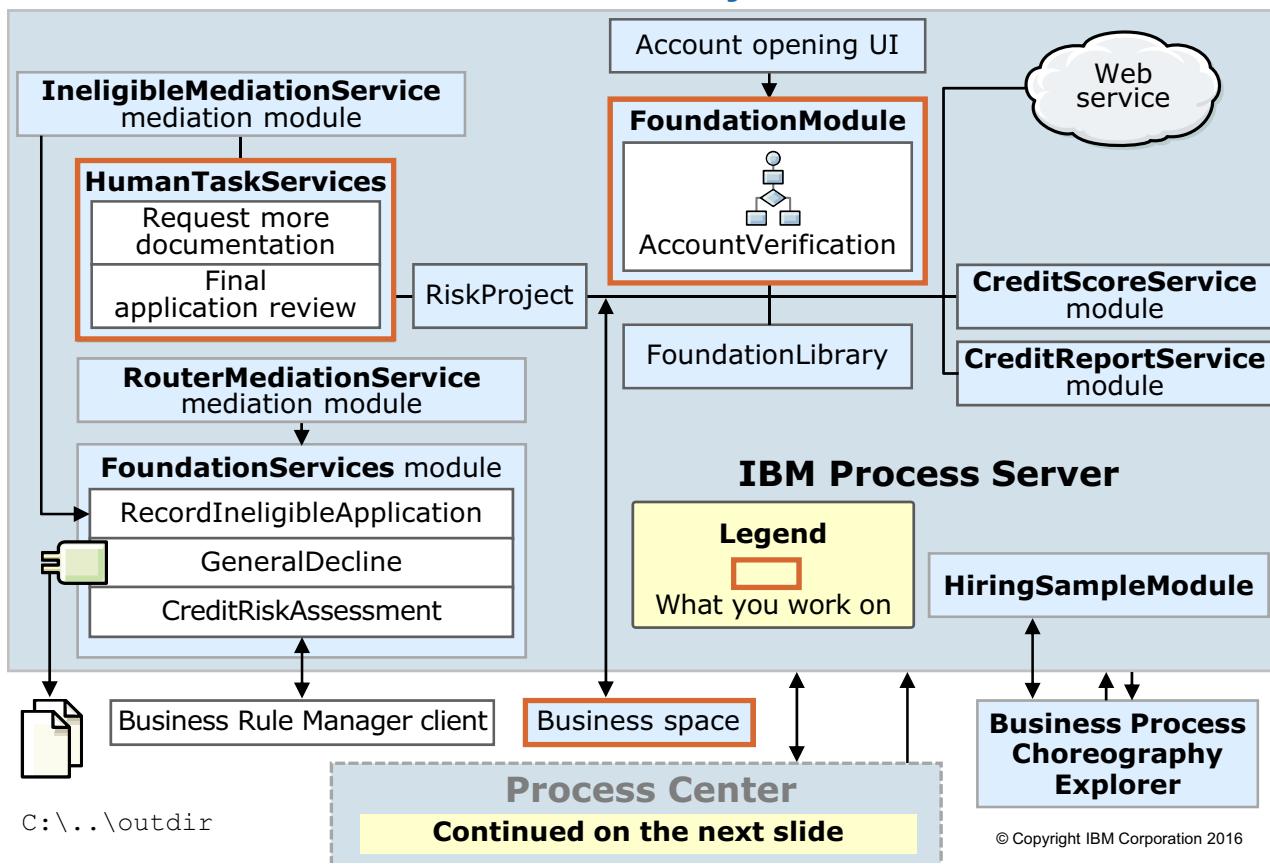


Figure 3-92. Course business scenario: What you work with in Exercise 13

WB8601.2

Notes:

Exercise 14



Using component tests

© Copyright IBM Corporation 2016

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

10.1

Figure 3-93. Exercise 14

WB8601.2

Notes:



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

- You learn how to unit test your SCA applications
- You learn how to use component tests to test SCA components

© Copyright IBM Corporation 2016

Figure 3-94. What are some key tasks that you do in Exercise 14?

WB8601.2

Notes:

WebSphere Education

Components that are required for Exercise 14

Prebuilt components that are imported in this lab:

1. All components from completed Exercise 13

New components that you create in this lab:

1. **FoundationModuleTestProject** and **RiskTestProject** component test projects
2. **AccountVerificationTestSuite** and **RiskTestSuite** test suites
3. **FinalApplicationReviewEmulator** emulator

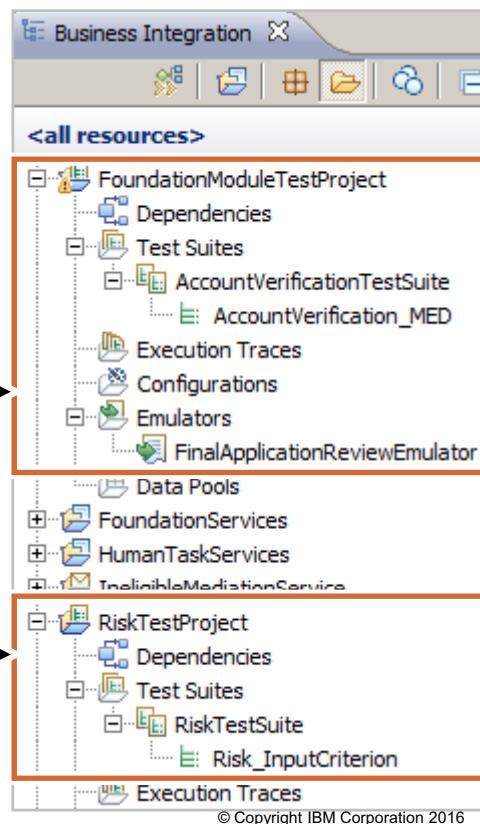


Figure 3-95. Components that are required for Exercise 14

WB8601.2

Notes:

Course business scenario: What you work with in Ex 14

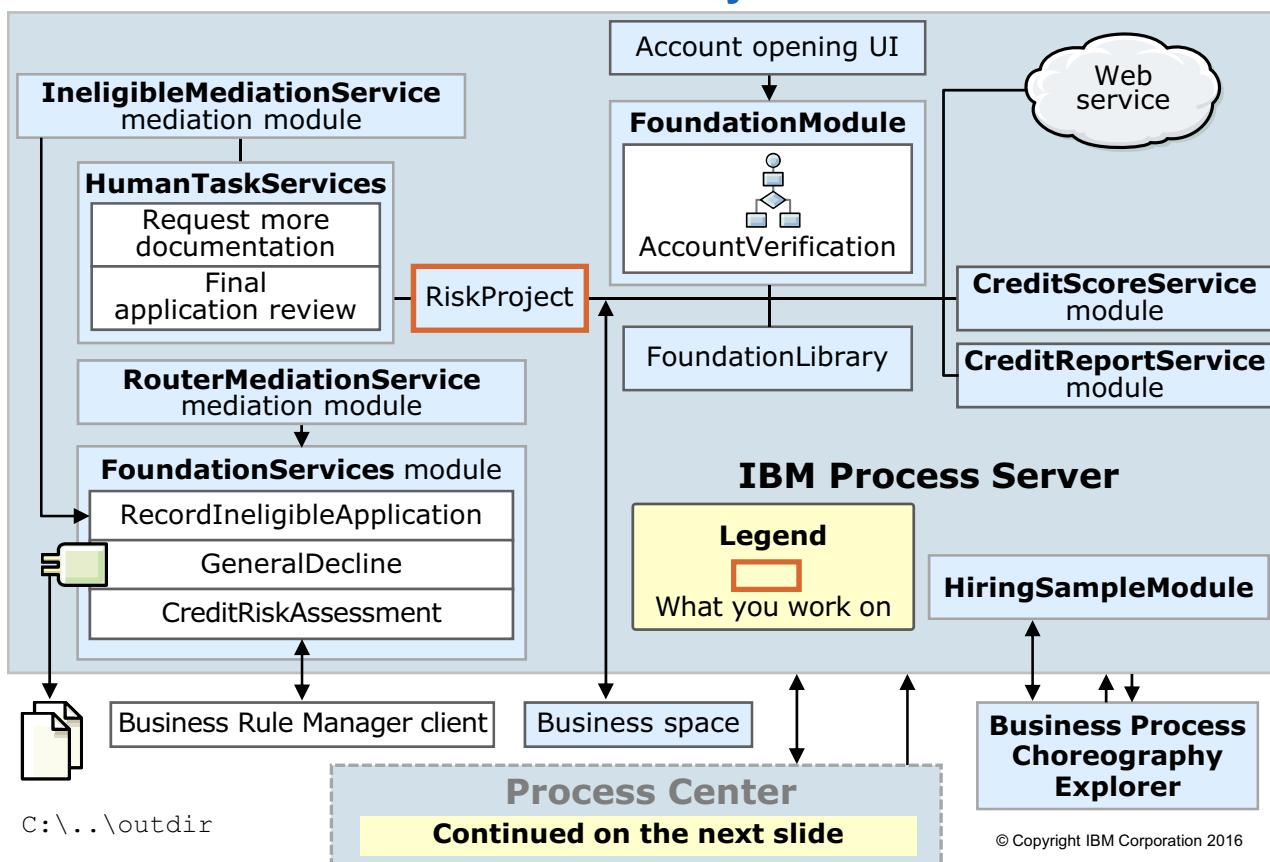


Figure 3-96. Course business scenario: What you work with in Exercise 14

WB8601.2

Notes:

Exercise 15



Bringing the UTE Process Server online

© Copyright IBM Corporation 2016

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

10.1

Figure 3-97. Exercise 15

WB8601.2

Notes:

What are some key tasks that you do in Exercise 15?

- Enable the communication between the Process Center and Process Server
 - This communication is needed for process application deployment from a central Process Center to any connected process server

© Copyright IBM Corporation 2016

Figure 3-98. What are some key tasks that you do in Exercise 15?

WB8601.2

Notes:



Components that are required for Exercise 15

Prebuilt components that are imported in this lab:

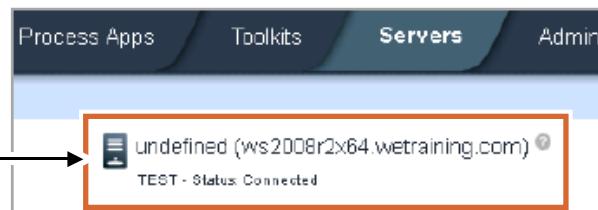
None

New components that you create in this lab:

None

Environment changes that you make in this lab:

1. You do not work with the Integration Designer in this exercise
2. Run the **wsadmin** command to change the connection properties
3. Change the environment through the administrative console



© Copyright IBM Corporation 2016

Figure 3-99. Components that are required for Exercise 15

WB8601.2

Notes:

Course business scenario: What you work with in Ex 15

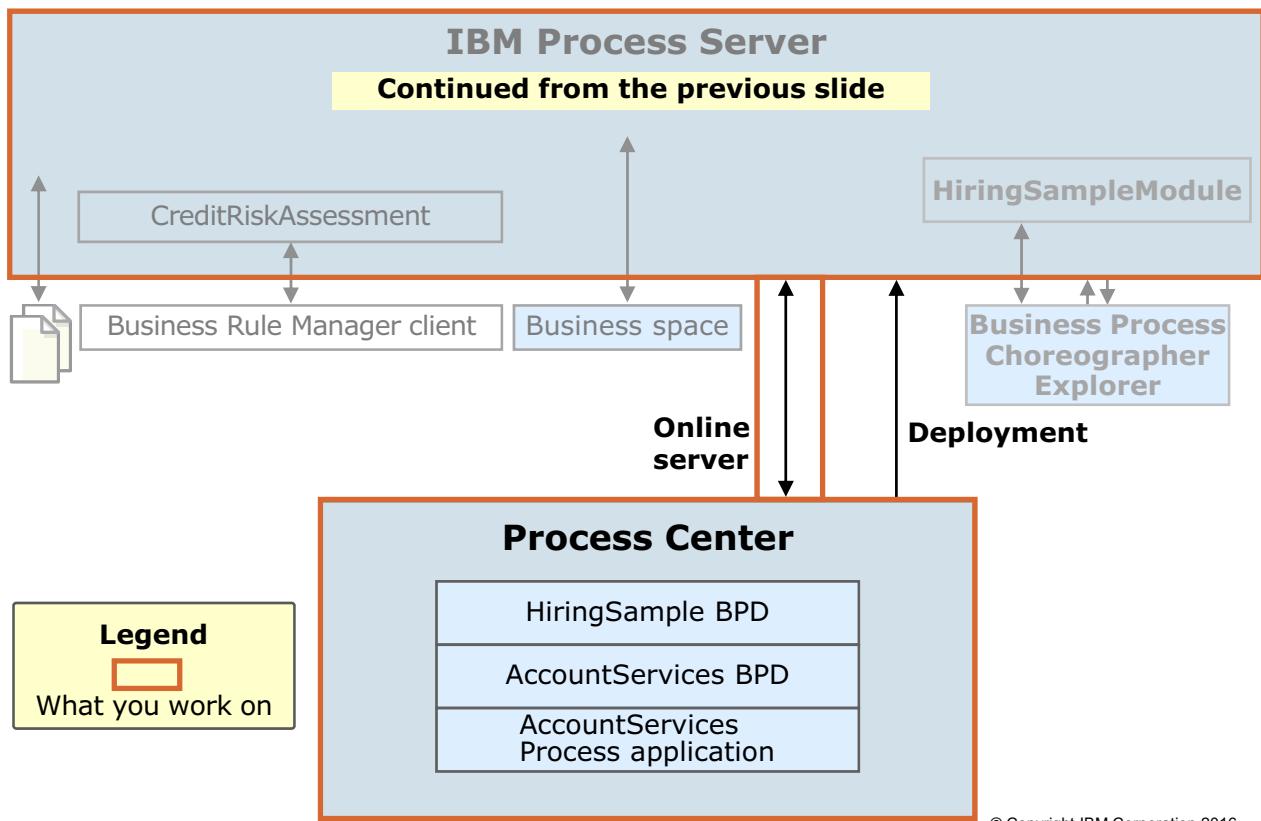


Figure 3-100. Course business scenario: What you work with in Exercise 15

WB8601.2

Notes:



Exercise 16



Exploring IBM Process Center

© Copyright IBM Corporation 2016

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

10.1

Figure 3-101. Exercise 16

WB8601.2

Notes:



What are some key tasks that you do in Exercise 16?

- You explore the Process Center and learn how to manage process applications: installing, updating, deploying, and deleting

© Copyright IBM Corporation 2016

Figure 3-102. What are some key tasks that you do in Exercise 16?

WB8601.2

Notes:



The image shows the WebSphere Education logo on the left and the IBM logo on the right, both set against a blue header bar.

Components that are required for Exercise 16

Prebuilt components that are imported in this lab:

1. All components from the completed Exercise 13

New components that you create in this lab:

1. **Account Verification Services toolkit**
2. **AccountServicesapp** process application



© Copyright IBM Corporation 2016

Figure 3-103. Components that are required for Exercise 16

WB8601.2

Notes:

Course business scenario: What you work with in Ex 16

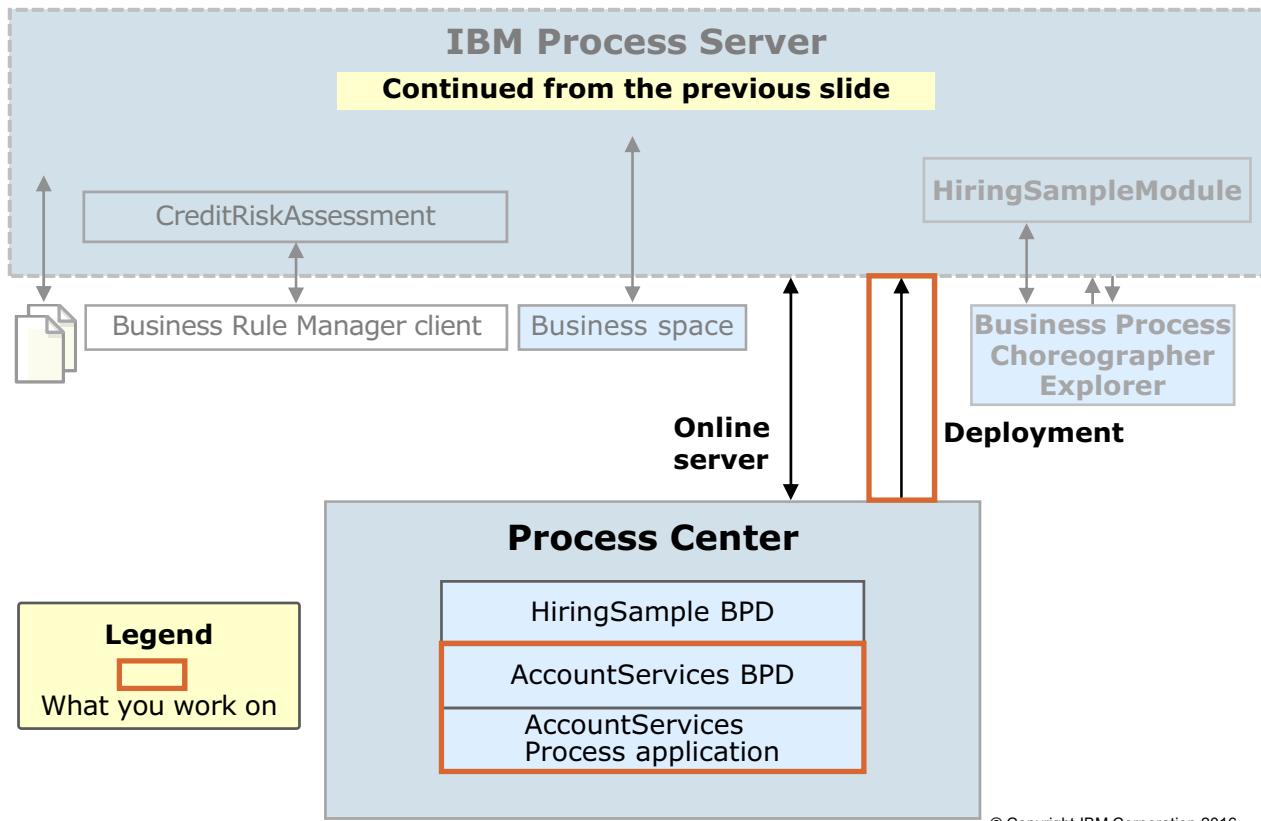


Figure 3-104. Course business scenario: What you work with in Exercise 16

WB8601.2

Notes:

Exercise 17



Implementing Advanced Integration services

© Copyright IBM Corporation 2016

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

10.1

Figure 3-105. Exercise 17

WB8601.2

Notes:



What are some key tasks that you do in Exercise 16?

- In this exercise, you create Advanced Integration service, which IBM Process Designer uses to invoke IBM Integration Designer implementations
 - The capabilities of Advanced Integration with IBM Business Process Manager Advanced open a whole world of service orientation
 - Access to process choreography through BPEL, adapters, easy incorporation of Java, and the tools that are needed to test every part of the solution

© Copyright IBM Corporation 2016

Figure 3-106. What are some key tasks that you do in Exercise 16?

WB8601.2

Notes:

Course business scenario: What you work with in Ex 17

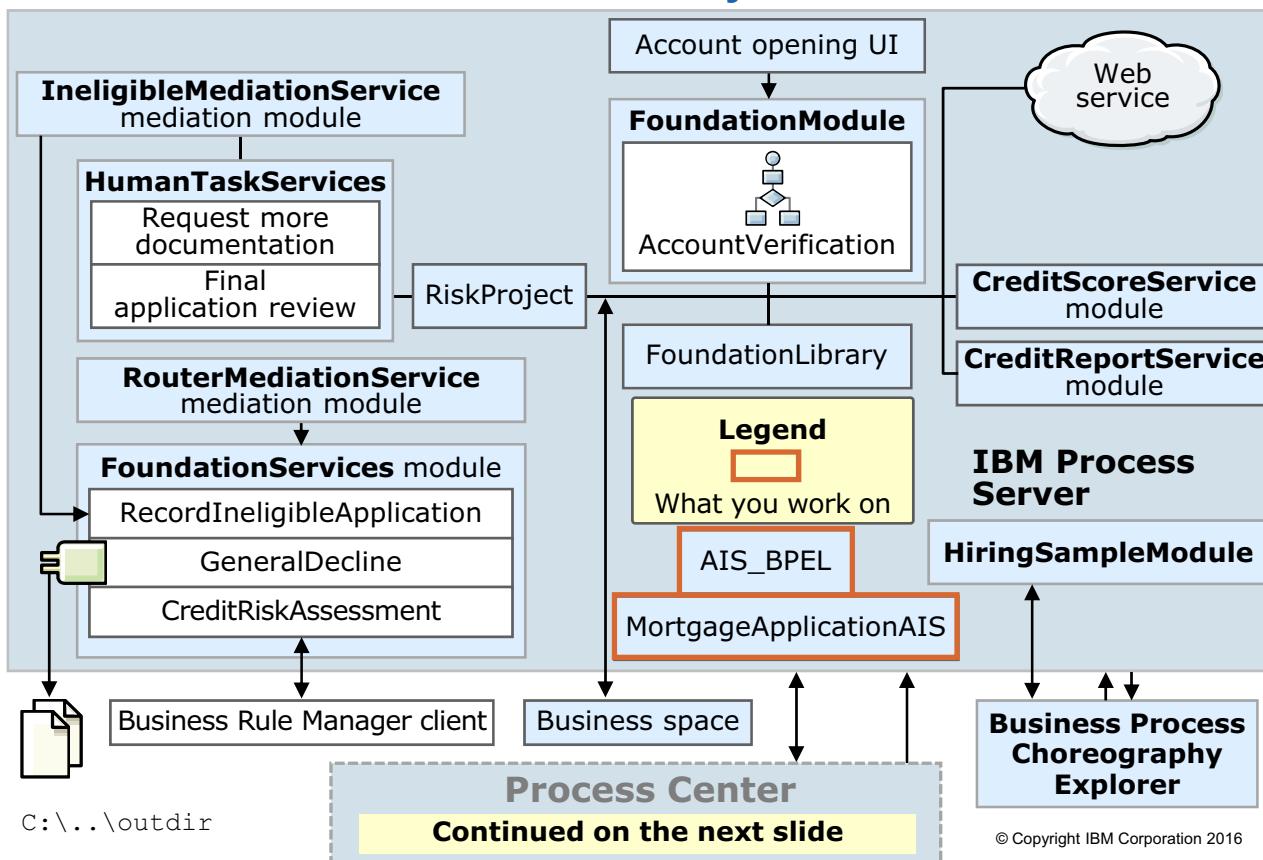


Figure 3-107. Course business scenario: What you work with in Exercise 17

WB8601.2

Notes:

Course business scenario: What you work with in Ex 17

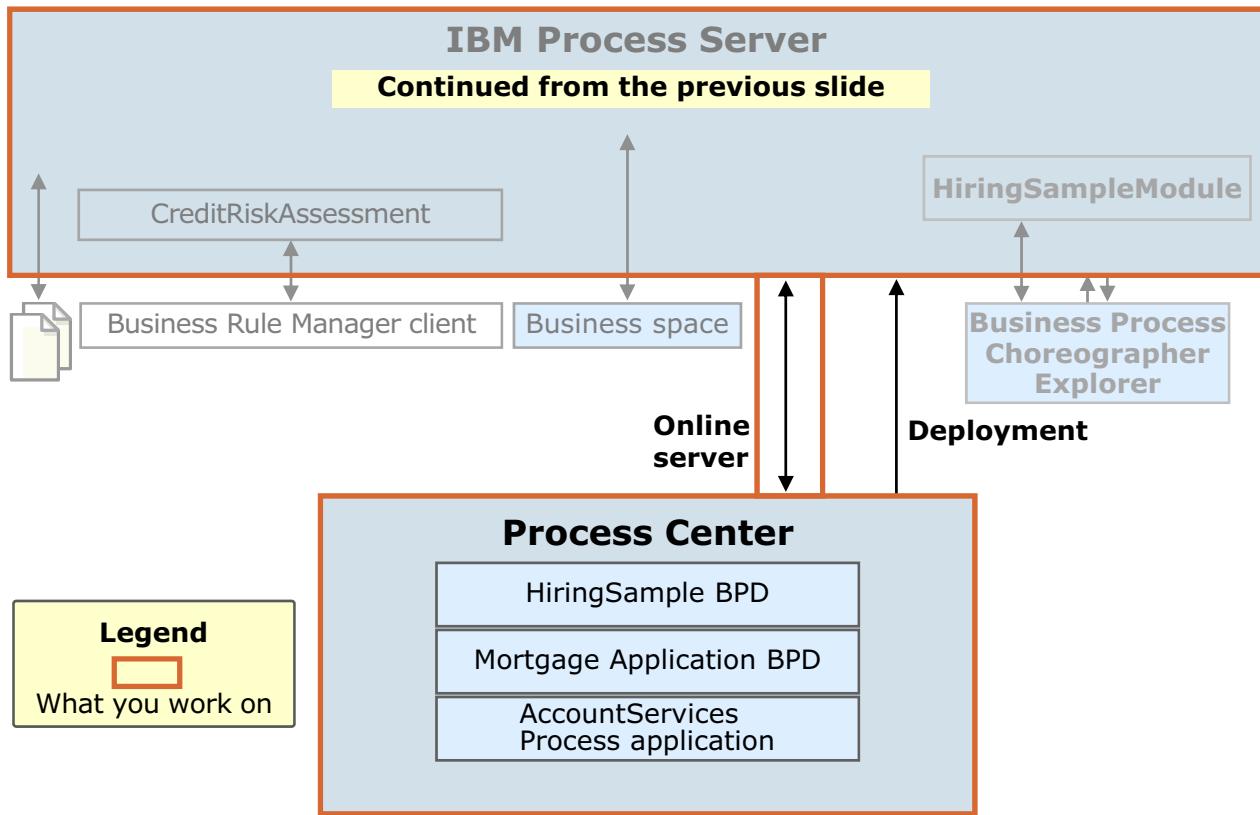


Figure 3-108. Course business scenario: What you work with in Exercise 17

WB8601.2

Notes:



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 2016

Figure 3-109. Unit summary

WB8601.2

Notes:

Unit 4. IBM Integration Designer overview

What this unit is about

In this unit, you learn about the IBM Integration Development environment and how it relates to other IBM Business Process Manager V8.5.7 tools, such as IBM Process Designer and IBM Process Server. The unit also explains how to use the features of IBM Integration Designer.

What you should be able to do

After completing this unit, you should be able to:

- Describe the purpose and business value of IBM Integration Designer
- Define the types of projects that are developed in IBM Integration Designer: modules, mediation modules, and libraries
- Describe how to create and deploy applications in IBM Integration Designer

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the purpose and business value of IBM Integration Designer
- Define the types of projects that are developed in IBM Integration Designer: modules, mediation modules, and libraries
- Describe how to create and deploy applications in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 4-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to IBM Integration Designer
- Application composition and deployment
- Documentation and organization capabilities

© Copyright IBM Corporation 2016

Figure 4-2. Topics

WB8601.2

Notes:

4.1. Introduction to IBM Integration Designer

Introduction to IBM Integration Designer



© Copyright IBM Corporation 2016

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

10.1

Figure 4-3. Introduction to IBM Integration Designer

WB8601.2

Notes:

WebSphere Education

IBM

IBM Integration Designer: Create, assemble, and test applications

IBM Integration Designer is the unified development tool for building integration applications for IBM Process Server and IBM Process Center

- Visual development environment that requires minimal programming skill
- Provides prebuilt mediation functions and BPEL activities
- A comprehensive environment for developing, assembling, testing, deploying, and managing integration modules and mediation modules for run time



The screenshot shows the 'About IBM Integration Designer 8.5.7' window. It displays the following information:

- IBM Integration Designer
- Version: 8.5.7.0
- Build id: 20160211-201551.0.linux
- To determine the version and build level of the IBM Process Server integrated test environment, run the `versionInfo` command in the `<test_environment_install_directory>\bin` directory.
- (c) Copyright IBM Corporation and others 2000, 2016. All rights reserved.

At the bottom, there is a row of twelve small icons representing different tools or components.

Figure 4-4. IBM Integration Designer: Create, assemble, and test applications

WB8601.2

Notes:

IBM Integration Designer is the common tool for building integration solutions across many of the IBM Business Process Manager runtime environments: IBM Process Server, IBM Process Center, and IBM Business Monitor. It simplifies integration with rich features that accelerate the adoption of SOA by rendering existing IT assets as service components, encouraging reuse, and efficiency.

IBM Integration Designer allows integration developers to assemble complex business solutions that require minimal skills, whether they involve processes, mediations, adapters, or code components. Users can construct process and integration solutions by using “drag-and-drop” technology without having a working knowledge of Java.

In addition to providing the tools necessary to build and assemble these artifacts, the product includes a full test framework. This framework allows the tests to be completed in a seamless fashion in an environment identical to a production environment, but without having to do the steps to administer and configure a production environment.

IBM Integration Designer roles

Role	Responsibilities
Integration developer	<ul style="list-style-type: none"> • Focuses on building SOA and EAI solutions <ul style="list-style-type: none"> — Top-down, bottom-up, or meet-in-the-middle • Creates applications from integrated components • Has a basic understanding of business modeling • Expects authoring tools to simplify and abstract advanced implementation details • Is familiar with basic programming concepts <ul style="list-style-type: none"> — Loops, conditions, string manipulation, and other programming concepts • Understands business process choreography, workflow (including human interaction), WSDL, and BPEL • Creates mediation modules to implement connectivity logic • Works with the IBM Process Center repository • Manages and deploys snapshots in the IBM Process Center
Application (IT) developer	<ul style="list-style-type: none"> • Is knowledgeable in one or more application development platforms (Java EE) • Understands SOA, process choreography, workflow, WSDL, and BPEL • Implements application-specific business logic for integrated solutions such as EJBs • Exposes application logic as a service

© Copyright IBM Corporation 2016

Figure 4-5. IBM Integration Designer roles

WB8601.2

Notes:

The primary focal group for IBM Integration Designer is integration developers who are not required to be fluent Java programmers. Therefore, IBM Integration Designer **hides** advanced implementation details as much as possible. However, the base technology is still Rational Application Developer. Application developers can still do EJB development by using IBM Integration Designer when it is necessary (it is necessary to turn on some more capabilities).

Two primary user roles that are associated with IBM Integration Designer, and a description of each, are provided here:

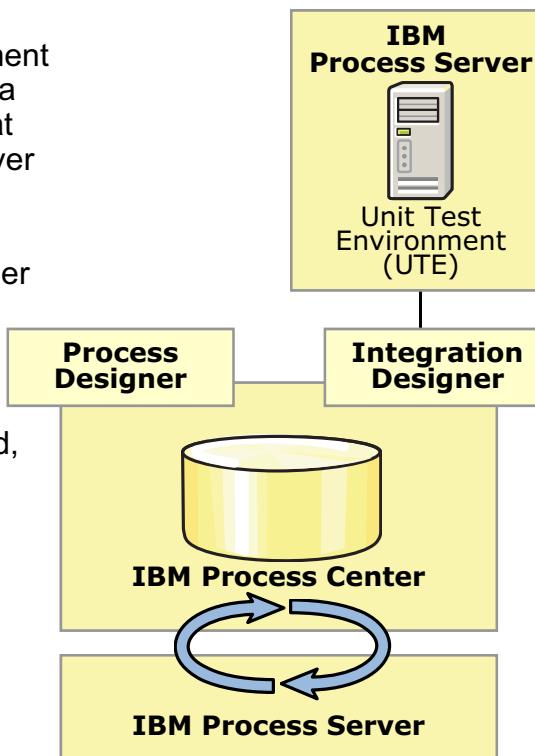
- The **integration developer** is the primary user role that is associated with IBM Integration Designer. This user type is focused on building service-oriented solutions, and expects the authoring tools to simplify many of the advanced implementation details that are associated with building complex business applications. Although the integration developer is typically familiar with basic programming concepts (such as loops, conditions, and string manipulations), this user might not be an expert in a particular implementation platform or programming language. The purpose of the tools that are provided in IBM Integration Designer is to provide the integration developer with a development environment for building complex business applications without extensive knowledge of the underlying implementation technologies. When

those business process applications are built and are ready for deployment, the integration developer is responsible for managing the process application on the IBM Process Center repository. This responsibility includes managing snapshots for the process application, deploying those snapshots, and working with toolkits.

- Another important user role for IBM Integration Designer is the **application developer**. Unlike the integration developer, the application developer is typically knowledgeable in one or more programming languages or application development platforms. The application developer also has a basic understanding of one or more technologies that are associated with building integrated business applications such as SOA, process choreography, workflow, WSDL, or BPEL. Application developers have more extensive knowledge of the implementation details that are associated with building a business application. For this reason, they are typically responsible for implementing the application-specific business logic for the business application that is being built. In addition, with the introduction of the SCA programming model, it is also the application developer who is responsible for exposing specific application logic as an SCA service component.

Installing IBM Business Process Manager Advanced V8.5.7

- Typical installation
 - Install software and configure the deployment manager and managed node profiles and a single cluster deployment environment that consists of a single node and a single server
 - Can install DB2 Express Edition, and it configures required databases
 - Automatically installs IBM Process Designer
- Test environment
 - If you install IBM Business Process Manager Advanced – Process Server or IBM Business Process Manager Advanced, you create a Process Center or Process Server profile during the installation
 - You can use the Process Center or Process Server test environment for deploying your applications
 - Use the BPMConfig command-line utility or Profile Management Tool to create a default profile or to configure an IBM Process Server profile



© Copyright IBM Corporation 2016

Figure 4-6. Installing IBM Business Process Manager Advanced V8.5.7

WB8601.2

Notes:

You can install IBM Business Process Manager by using typical, custom, or silent installation.

You can use a typical installation to install and configure the software by using the product launchpad. The typical installation installs the software, and configures the deployment manager, managed node profiles, and a single cluster deployment environment that consists of a single node and single server.

Or you can choose custom installation and then use the `BPMConfig` command to generate database scripts, configure a deployment manager and one or more managed node profiles, and create a pattern-based network deployment environment.

After installing Business Process Manager, you can install authoring environments, more tools, and add-ons. If you are installing Business Process Manager Advanced, you can install IBM Integration Designer either before or after installing Business Process Manager Advanced.

In Integration Designer, a server in the Servers view for the unit test environment (UTE) can be automatically or manually created. If you install the UTE locally, a new process server is automatically created when you start Integration Designer and use a new workspace. If you are

working with a remote UTE server or if you are using a workspace that was created before the UTE was installed locally, you can manually create a server.

IBM Installation Manager

- Installation Manager is the Eclipse-based tool to manage the update, modification, rollback, and uninstallation of product packages
 - Installation Manager installs automatically before the IBM Integration Designer installation package
 - It also manages licenses
- Use Installation Manager to move the product installation to the most recent update level
 - Skip in-between updates
- Update and modify both the development and test environments through the same tool
 - One tool updates several IBM Business Process Manager components (IBM Business Process Manager Advanced, IBM Integration Designer, IBM Process Designer, and WebSphere Application Server ND)OS

© Copyright IBM Corporation 2016

Figure 4-7. IBM Installation Manager

WB8601.2

Notes:

You can use IBM Installation Manager to install several products simultaneously. In addition, you can install product updates during product installation. When updates are applied, older versions of plug-ins are automatically uninstalled.

On disk, an Eclipse-based product offering (or “package”) is structured as a collection of plug-ins. Each plug-in contains the code that provides some of the function. The code and other files for a plug-in are installed on the local computer, and get activated automatically as required. Plug-ins are grouped into features. A feature is the smallest unit of separately downloadable and installable function.

Large Eclipse-based products can organize their features into trees that start from the root feature that represents the entire product. This root feature then includes smaller units of function all the way down to leaf features that list one or more plug-ins and fragments. The capability of group features hierarchically allows products to be stacked by using a nested approach: a large product can build on top of a smaller one by including it and adding more features.

- **Package groups:** For updating or modifying an existing installation for IBM Business Process Manager or IBM Integration Designer, you must specify a package group. A package group

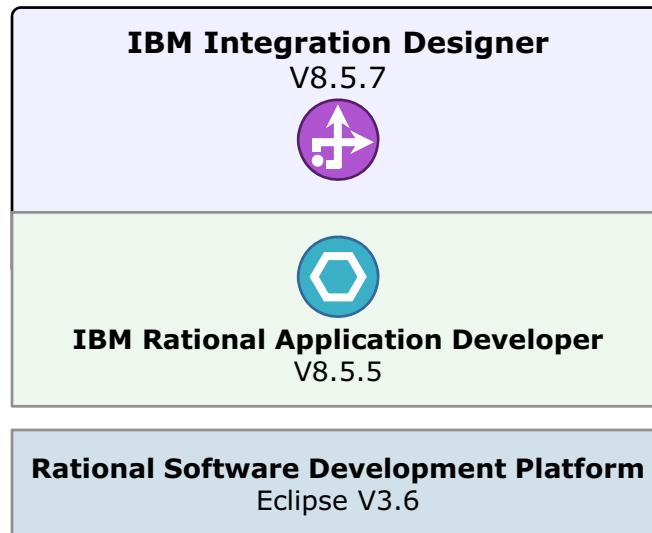
represents a directory in which packages share a common user interface or workbench with other packages in the same group.

When you update or modify the Business Process Manager or Integration Designer package, you can create a package group or install the packages into an existing package group.

- **Shared resources directory:** The shared resources directory is the directory where Eclipse plug-ins are located so that one or more product package groups can use them.

IBM Integration Designer: Platform architecture

- IBM Integration Designer is based on Rational Application Developer
 - Not all Rational Application Developer features are included
- Rational Software Development Platform provides the foundation for IBM Integration Designer and Rational Application Developer
 - Based on Eclipse V3.6
 - Contains the common components for Eclipse-based products
 - Installed one time per system with the first product



© Copyright IBM Corporation 2016

Figure 4-8. IBM Integration Designer: Platform architecture

WB8601.2

Notes:

IBM Integration Designer V8.5.7 is built on top of Rational Application Developer V8.5.5 Eclipse-based technology. IBM Integration Designer is based on the core IBM Rational Application Developer product. However, it is important to note that several features are not included in the core IBM Rational Application Developer product on which IBM Integration Designer is built.

Some of the features that are available in Rational Application Developer but not included in IBM Integration Designer are:

- Crystal Reports tools
- Tools for WebSphere Application Server
- Code review
- Rational Unified Process (RUP)
- Rational RequisitePro integration
- Rational Team Concert
- Rational Team Concert debug extensions

Because IBM Integration Designer is based on a subset of the full IBM Rational Application Developer tools, advanced users who need all the capabilities require a full license for IBM Rational Application Developer.

As the diagram on this slide shows, IBM Integration Designer is built upon the Rational Software Development Platform. The Rational Software Development Platform is based on Eclipse V3 technology, and each IBM product that is built upon this platform coexists and shares plug-ins with other Rational Software Development Platform-based products. The Rational Software Development Platform is installed once for each system with the first product that is installed. When other products that are built on this platform are installed on the system, only the necessary plug-ins are installed.

The current version of Rational Software Architect and IBM Integration Designer are all based on Eclipse V3.6 technology; therefore, shell sharing is supported. Users can switch between different perspectives to develop UML, SCA, or business models.

Installing IBM Integration Designer

- Use Launchpad for the installation
- Use IBM Installation Manager to update or modify an existing installation
- Select the **IBM Integration Designer for IBM Business Process Manager Advanced - Process Server** option in Launchpad to install an IBM Process Server test environment along with IBM Integration Designer
- The following packages are selected for installation:
 - WebSphere Application Server
 - IBM Business Process Manager Advanced - Process Server
 - DB2 Express
 - Integration Designer

© Copyright IBM Corporation 2016

Figure 4-9. Installing IBM Integration Designer

WB8601.2

Notes:

IBM Integration Designer – Process Server installation

If you select the **IBM Integration Designer for IBM Business Process Manager Advanced – Process Server** option in the launchpad, the installation configuration includes an IBM Process Server test environment. The following packages are selected for installation:

- WebSphere Application Server Network Deployment
- IBM Business Process Manager Advanced – Process Server
- DB2 Express
- Integration Designer

A default profile for IBM Business Process Manager Advanced – Process Server is selected on the Features page. You can use this profile to use the test environment immediately. By default, **Start working with the IBM Process Center first** is selected so that you can work with the Process Center. You can use the test environment installation to create an IBM Process Server profile. You can customize the environment to include multiple profiles. If you installed Business Process Manager Advanced – Process Server or Business Process Manager Advanced and created a Process Server profile, you can install Integration Designer only. You can use the Process Server

test environment that is already installed for deploying your applications. The installation process that is used to create the lab environment for this course is included in the exercise guide appendixes. For instructions on installing IBM Integration Designer and re-creating the exercise environment that is used in this course, see the exercise appendixes.

4.2. Application composition and deployment

Application composition and deployment



© Copyright IBM Corporation 2016

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

10.1

Figure 4-10. Application composition and deployment

WB8601.2

Notes:

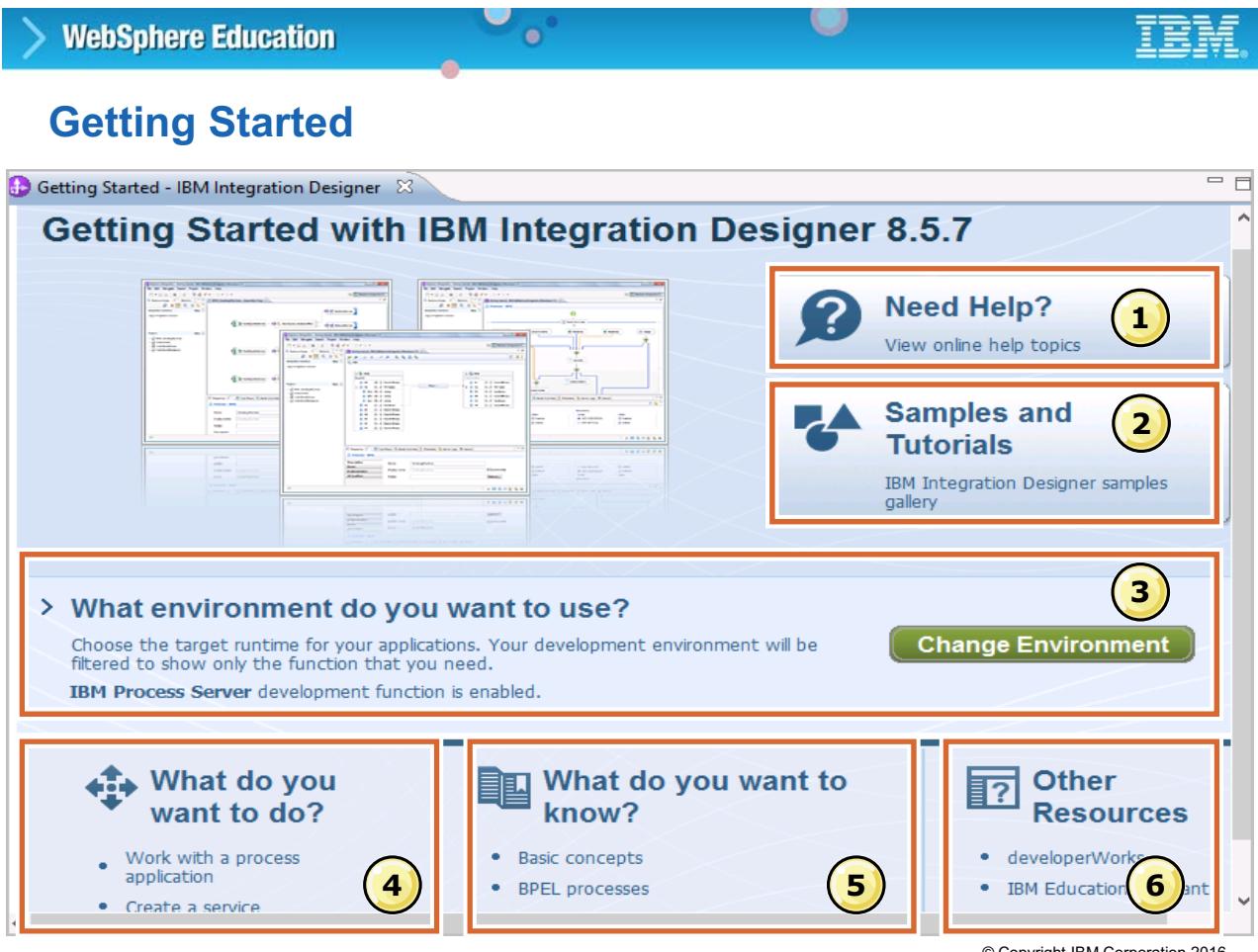


Figure 4-11. Getting Started

WB8601.2

Notes:

The following options are available, as shown in the image:

1. **Need Help:** Opens the IBM Integration Designer window for online help in the IBM Knowledge Center.
2. **Samples and Tutorials:** Opens the sample applications and tutorial instructions page that help you become familiar with IBM Integration Designer.
3. **Change Environment:** Provides the option to change the current target runtime environment. If you have multiple process servers in your environment, you can switch to the server you want to use for development or testing your applications.
4. **What do you want to do:** Provides direction and steps in creating a specific artifact in IBM Integration Designer.
5. **What do you want to know:** Provides definitions and explanations of various terms and components in IBM Integration Designer.
6. **Other Resources:** Opens a web browser with more education resources with links to developerWorks, IBM Education Assistant, IBM support, and FAQ.



Application composition and deployment

- Use the Business Integration perspective to develop business integration projects

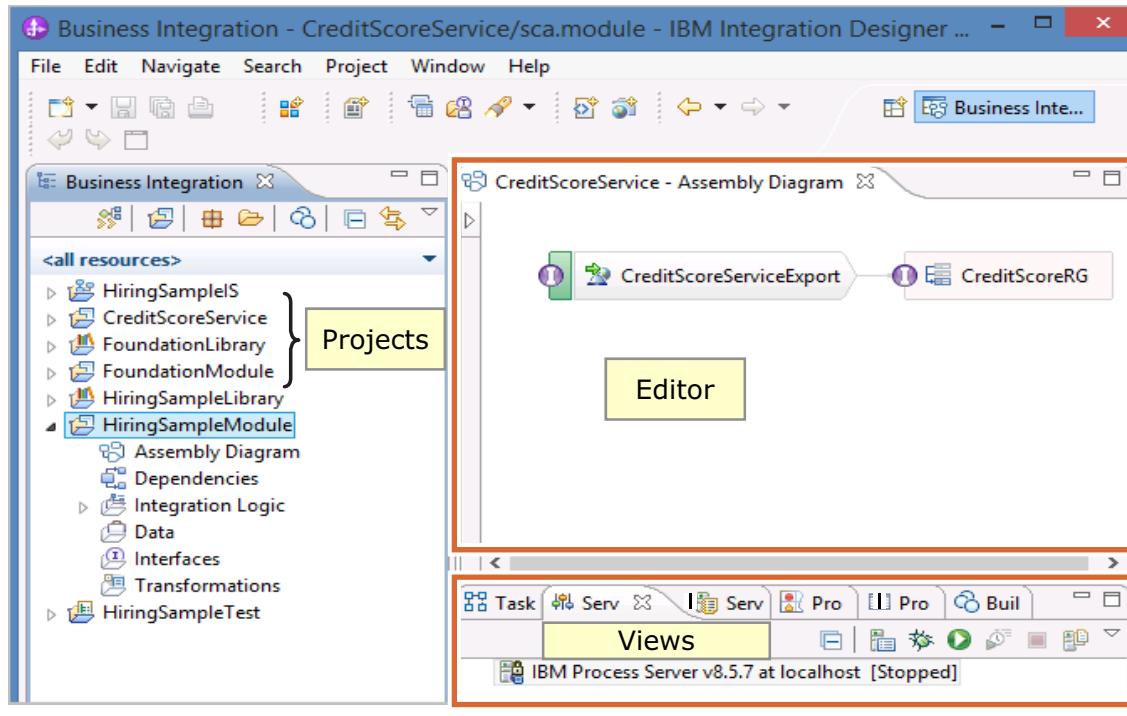


Figure 4-12. Application composition and deployment

WB8601.2

Notes:

The user interface consists of several perspectives, which are groups of views for various tasks. Use the Business Integration perspective to develop business integration projects.



Business Integration perspective

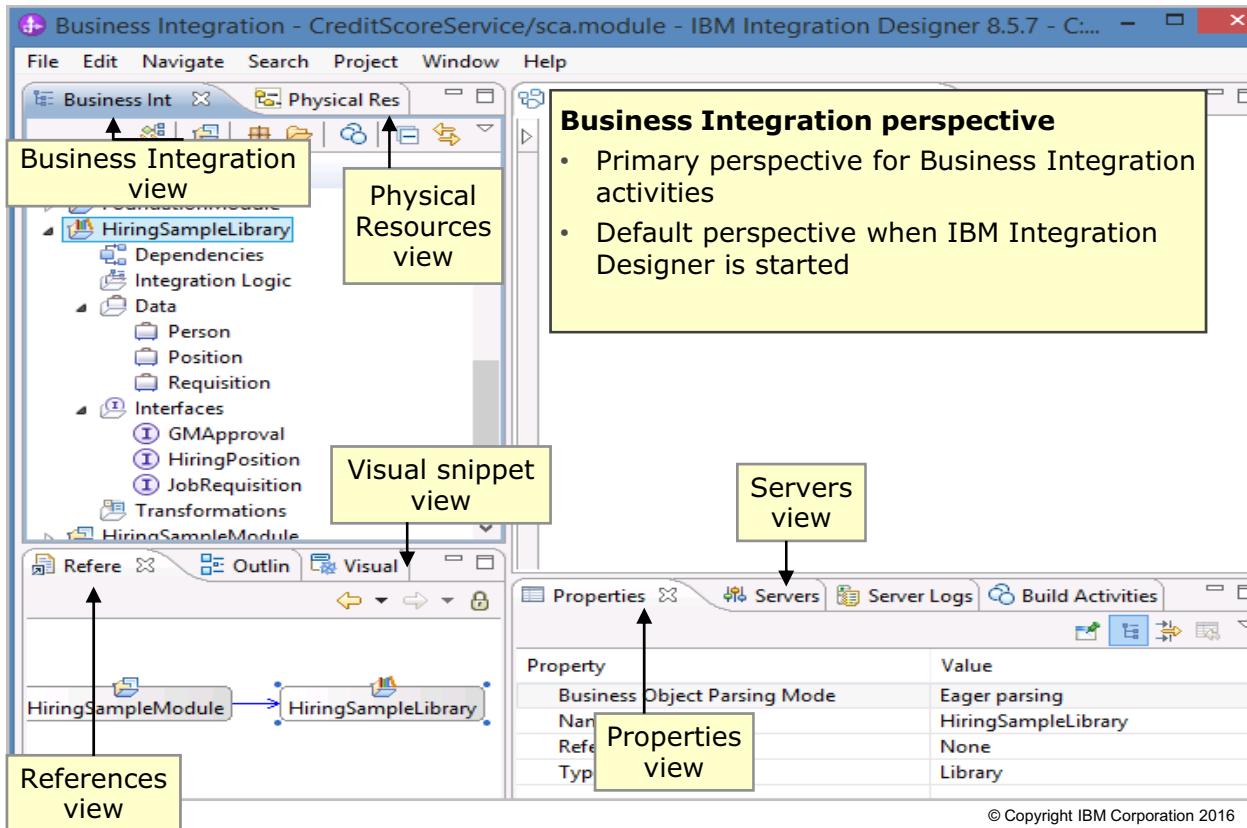


Figure 4-13. Business Integration perspective

WB8601.2

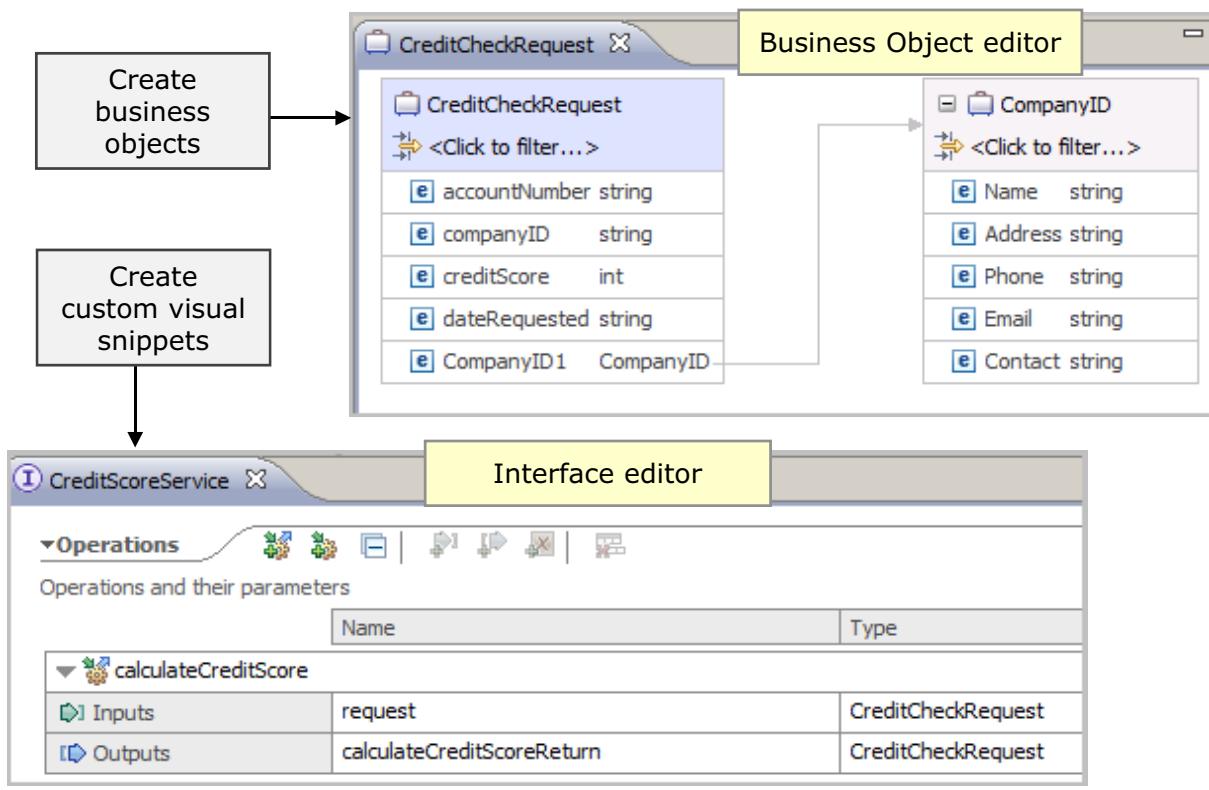
Notes:

The Business Integration perspective is composed of the following default views:

- Business Integration view:** The Business Integration view provides a logical view of the key resources in each module, mediation module, and library. Non-SCA projects (such as Java EE projects) are also shown in the Business Integration view in their natural form: the content and labels of the tree match the ones that are seen in the perspective. By showing non-SCA projects in the Business Integration view, you can edit non-SCA projects without switching perspectives.
- Editor pane:** When you open a resource from the Business Integration view with an editor, the resource is displayed in the editor pane. Each of the components that are presented in the IBM Process Server architectural diagram has an equivalent editor in IBM Integration Designer. These editors are used to create the components that form your integration and mediation modules.
- Task Flows view:** Task flows are provided as a way to learn related concepts and tasks in an interactive manner. Learning task flows briefly describe key concepts to quickly get you up to speed with IBM Integration Designer. Creating task flows can show you how to do key tasks.

- **Build Activities view:** Use the Build Activities view to manage builds and to view the build and server status of projects. In this view, you can also view the operational state of supported servers.
- **Properties view:** Use the Properties view to see detailed information about a selected artifact. When using the Properties view with editors, you are able to modify properties of elements you select.
- **Problems view:** The Problems view helps you debug errors by providing message text. You can see further help for the problem message by selecting a message and pressing F1.
- **Servers view:** Use the Servers view to manage server profiles. This view is also used for deploying applications, for starting and stopping servers (including starting in debugging mode), for creating tables and data sources, and for starting the administrative console and various clients. These clients include the BPEL Process Choreographer Explorer, Business Space, and business rules manager.
- **Server Logs view:** Use the Server Logs view to display the contents of the server console and server log files (the view automatically displays output from the test server JVM log: `SystemOut.log`). The view can also be used to load logs from other servers and to filter log messages by type.

Features overview: Visual tools (1 of 4)



© Copyright IBM Corporation 2016

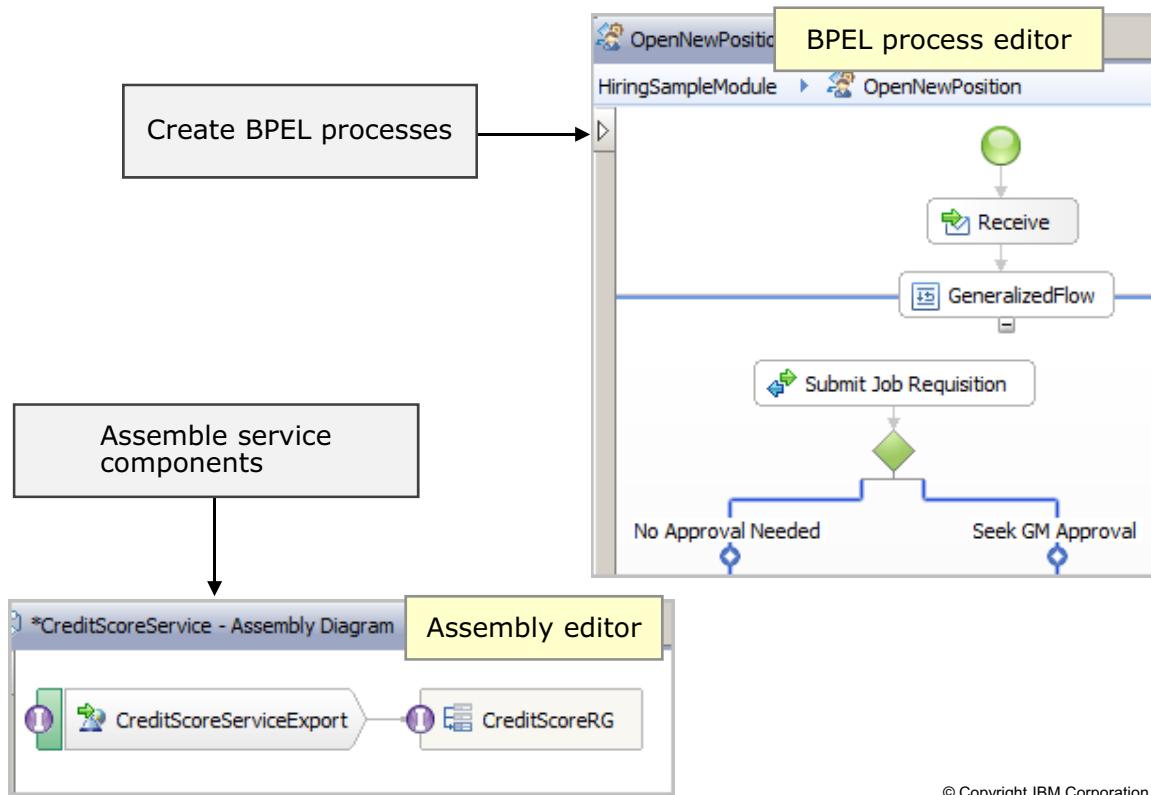
Figure 4-14. Features overview: Visual tools (1 of 4)

WB8601.2

Notes:



Features overview: Visual tools (2 of 4)



© Copyright IBM Corporation 2016

Figure 4-15. Features overview: Visual tools (2 of 4)

WB8601.2

Notes:

Features overview: Visual tools (3 of 4)

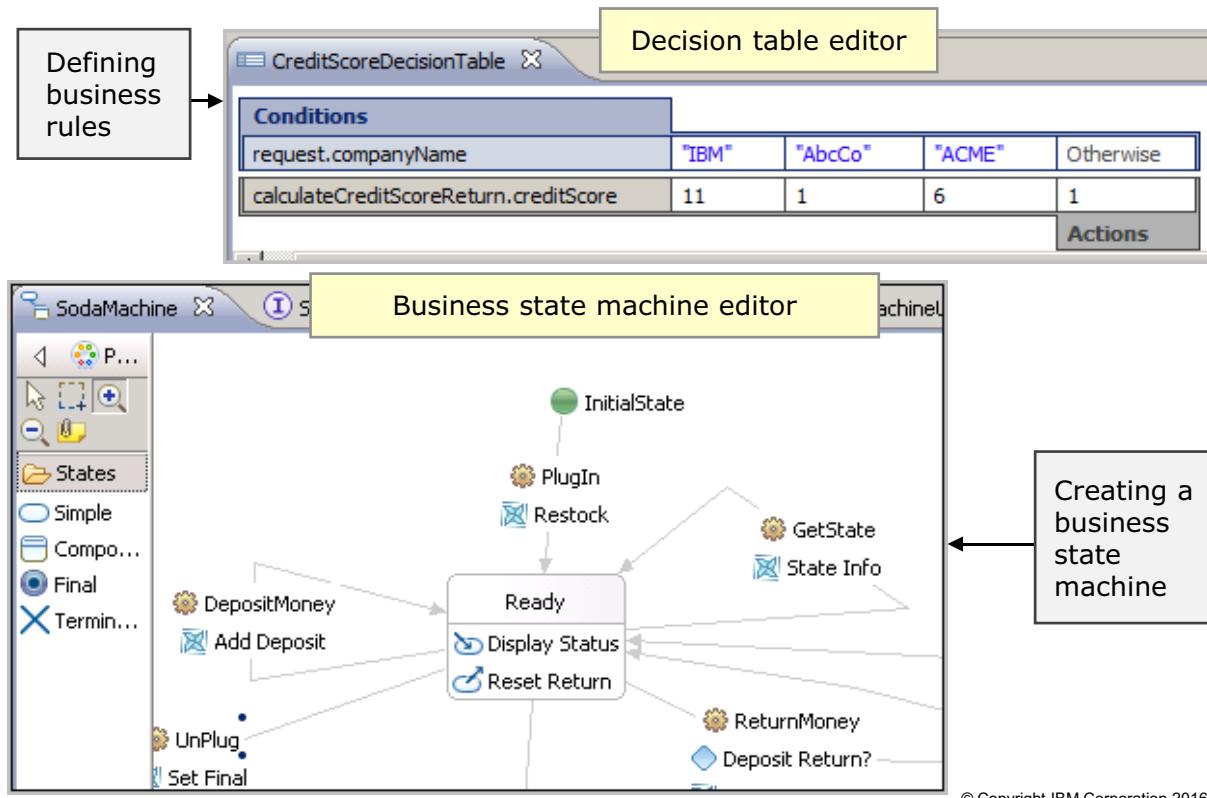


Figure 4-16. Features overview: Visual tools (3 of 4)

WB8601.2

Notes:

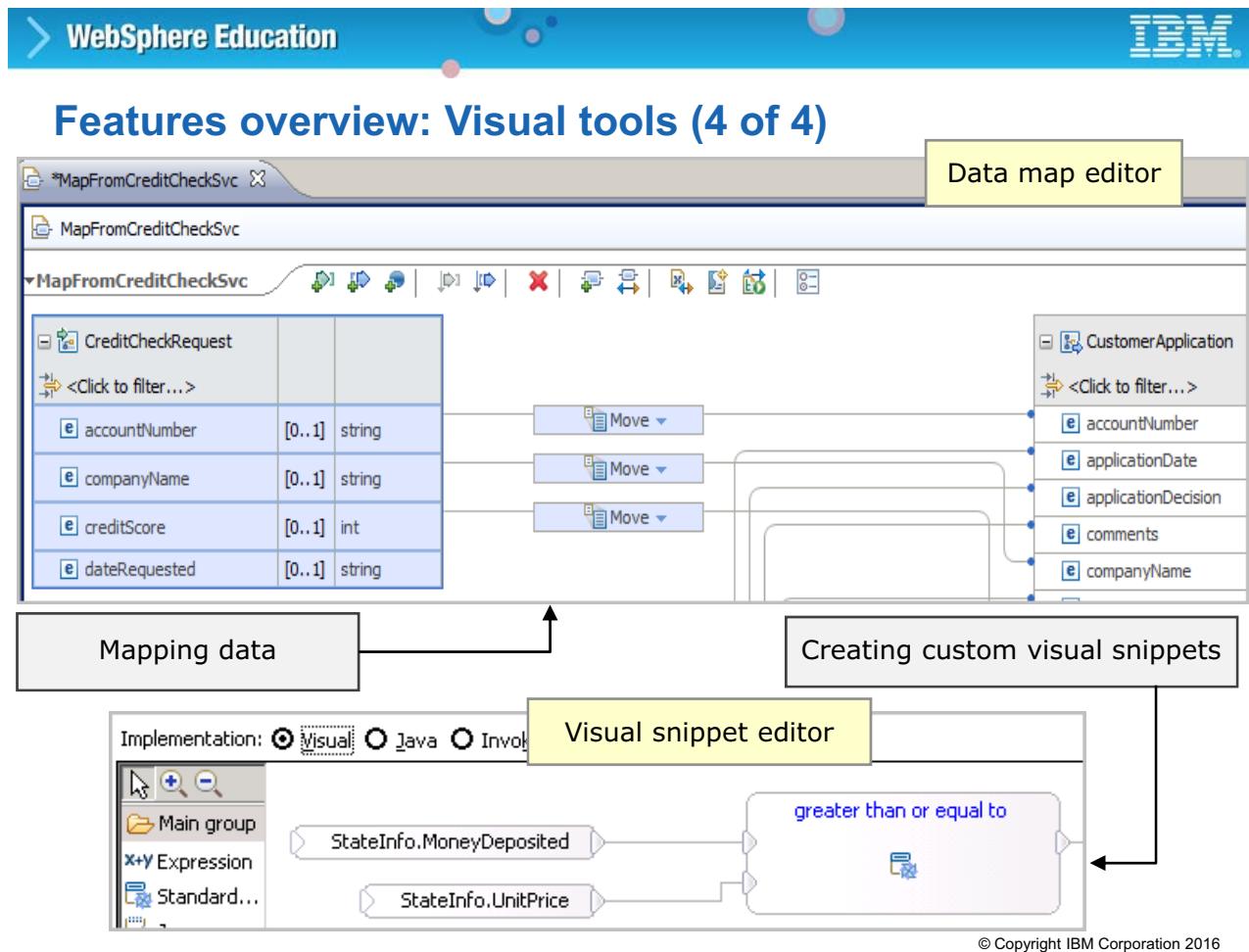


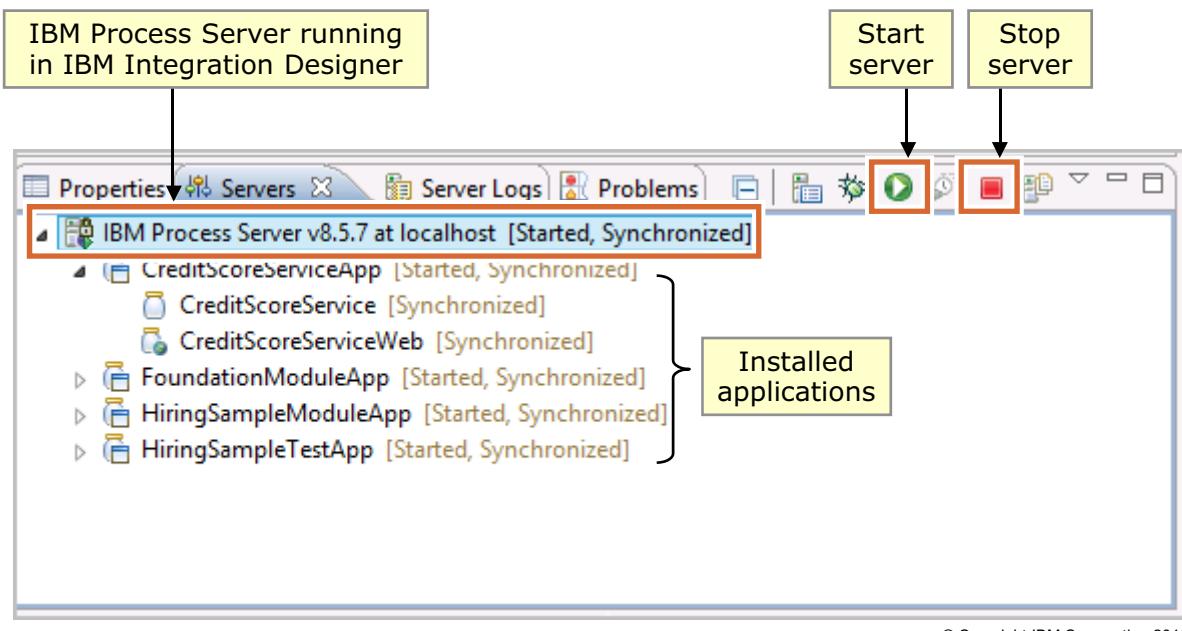
Figure 4-17. Features overview: Visual tools (4 of 4)

WB8601.2

Notes:

IBM Process Server unit test environment (1 of 3)

- In an IBM Integration Designer unit test environment (UTE), the server can be installed locally or on a remote machine.



© Copyright IBM Corporation 2016

Figure 4-18. IBM Process Server unit test environment (1 of 3)

WB8601.2

Notes:

In an IBM Integration Designer unit test environment (UTE), the server can be installed locally or on a remote machine. When Process Server is installed locally, IBM Integration Designer finds it and displays it in the Servers view.

IBM Integration Designer unit test environment has the following characteristics:

- An end-to-end test framework
- A visual tool for starting, testing, viewing, and interacting with modules or components that are being tested
- Ensures that the necessary modules are deployed to the runtime
- Enables testing before complete implementation is available

IBM Integration Designer test environment includes a server configuration that is named IBM Process Server that should be used when you want to test integration and mediation modules. The IBM Process Server product does not support a network deployment environment. Therefore, if you want to create a server to use for testing integration and mediation modules, you must be careful. If Process Server is not already included in the Servers view of the IBM Integration Designer unit test

environment, check the server configuration named IBM Process Server. Make sure that it points to either the IBM Integration Designer unit test environment or an IBM BPM Express server.

IBM Process Server unit test environment (2 of 3)

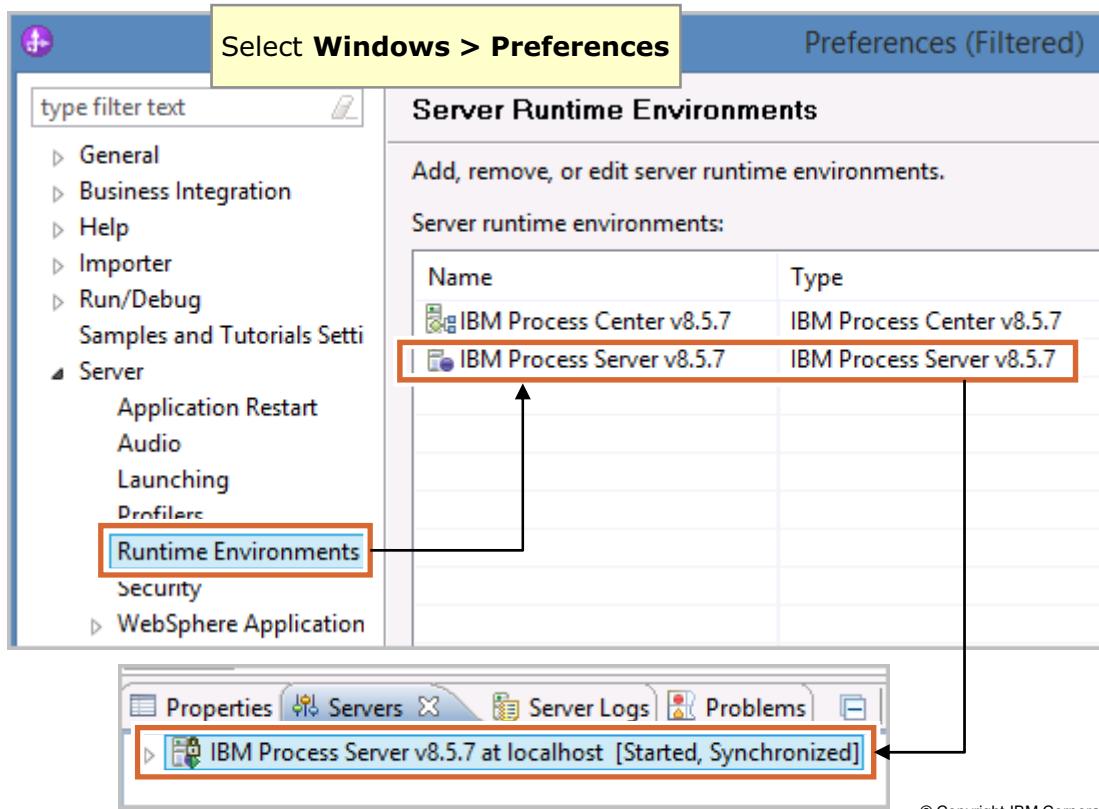


Figure 4-19. IBM Process Server unit test environment (2 of 3)

WB8601.2

Notes:

General Information
Specify the host name and other common settings.

Server name: IBM Process Server v8.5.7 at localhost
Host name: localhost
Runtime Environment: IBM Process Server v8.5.7

Server
Enter settings for the WebSphere Application Server.

Profile name: (with write permission) qbpmaps
Application server name: server1
Update server status interval (in milliseconds): 5000

Server connection types and administrative ports
 Automatically determine connection settings
 Manually provide connection settings

Publishing

Timeouts

Publishing settings for WebSphere

Security

Remote Server Settings

Server configuration editor

Right-click the server and click **Open**

© Copyright IBM Corporation 2016

Figure 4-20. IBM Process Server unit test environment (3 of 3)

WB8601.2

Notes:

Module

- Business Integration project type for developing SCA-based applications
- Basic unit of deployment to the runtime environment
 - A module is packaged in an EAR file

The screenshot shows the 'New Project' dialog box. In the 'Wizards:' list, the 'Module' option is highlighted with a red box and an arrow pointing from the left margin towards it. The 'Module' option is located under the 'Business Integration' category.

Figure 4-21. Module

WB8601.2

Notes:

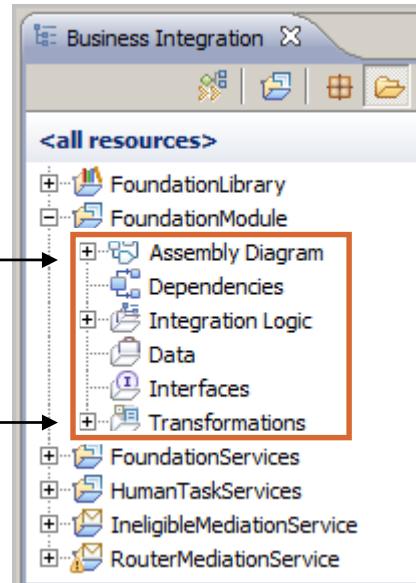
A module is a project that is used for development, version management, organizing business service resources, and deploying to the IBM Business Process Manager. A module is the basic unit of deployment to this runtime environment.

The two kinds of modules are business modules and mediation modules. Both types of modules can be deployed to the IBM Process Server. Modules can include processes, human tasks, mediation flows, state machines, business rules, and Java services. You can add dependent libraries, Java projects, and Java 2 Platform Enterprise Edition projects to a module and choose to deploy them with the module.

WebSphere Education

Module components

- Business integration modules include:
 - Assembly diagrams:** Wire SCA components together to form applications
 - Dependencies:** Include other modules, libraries, Java EE projects, and predefined resources
 - Integration logic:** Artifacts that do specific tasks (business processes, state machines, human tasks, business rules and rule groups, or mediation flows)
 - Data:** Business objects
 - Interfaces:** Service interfaces and their operations
 - Transformations:** Data (XML) maps and relationships



© Copyright IBM Corporation 2016

Figure 4-22. Module components

WB8601.2

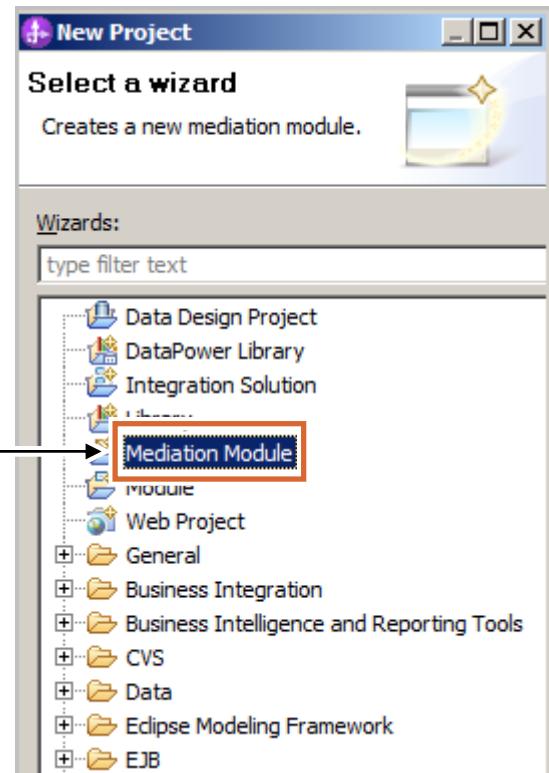
Notes:

The module provides the business services, which are modeled as Service Component Architecture (SCA) components that are wired together in a module assembly. This module can contain all the resources that are used in the service, but these resources are private and can be used only *within* the module. To reuse the logic in a module from other modules, you can export a component's interfaces.

The module assembly contains a diagram of the integrated business application, consisting of components and the wires that connect them. You use the assembly editor to visually compose the integrated application by using elements that you drag from the palette or from the tree in the Business Integration view. The implementations of components that are used in a module's assembly are stored within the module. Components belonging to other modules can be used through imports. Components in different modules can be wired together by publishing the services as exports that have their interfaces, and then dragging the exports into the required assembly diagram to create imports. Modules can export interfaces, but they cannot share resources. Resources that are going to be shared must be stored in a library.

Mediation Module

- Modules that can change the format, content, or target of service requests
- Operate on messages that are in-flight between service requesters and service providers
 - You can route messages to different service providers and to amend message content or form
- Can provide functions such as message logging, and error processing that is tailored to your requirements



© Copyright IBM Corporation 2016

Figure 4-23. Mediation Module

WB8601.2

Notes:

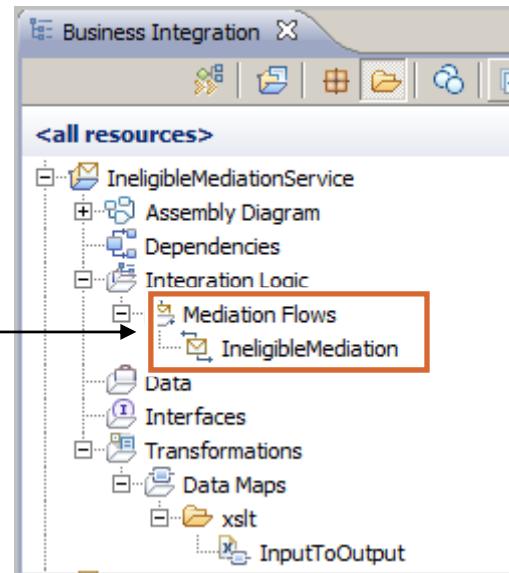
Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

Mediation modules operate on messages that are in-flight between service requesters and service providers. You can route messages to different service providers and to amend message content or form. Mediation modules can provide functions such as message logging, and error processing that is tailored to your requirements.

You can change certain aspects of mediation modules, from the administrative console, without having to redeploy the module.

Mediation module components

- Mediation modules include:
 - Artifacts present in integration modules: assembly diagram, dependencies, data types, interfaces, and transformations
 - **Integration logic:** Only mediation flows and subflows for processing messages that are passed between services



© Copyright IBM Corporation 2016

Figure 4-24. Mediation module components

WB8601.2

Notes:

The purpose of a mediation module that does not contain a mediation flow component is to transform service requests from one protocol to another. For example, a service request might be made by using SOAP/JMS but might need transforming to SOAP/HTTP before sending on.

You can view and make certain changes to mediation modules from IBM Business Process Manager. However, you cannot view or change the SCA components inside a module from IBM Business Process Manager. Use Integration Designer to customize SCA components.

Mediation modules implement mediation flows that intercept and modify messages that are passed between services. For example, mediation flows can be used to find services with specific characteristics that a requester is seeking and to resolve interface differences between requesters and providers. Typical mediations include:

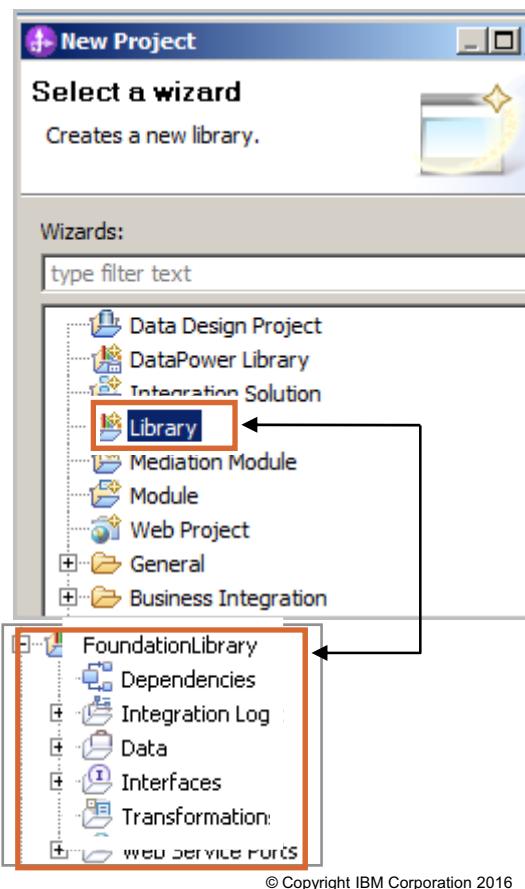
- Transforming a message from one format to another so that the receiving service can accept the message
- Conditionally routing a message to one or more target services based on the contents of the message
- Augmenting a message by adding data from a data source

A mediation module provides a mediation service, which is modeled as Service Component Architecture (SCA) components that are wired together in its module assembly. This module can contain all the resources that are used in the service, but these resources are private and can be used only within the module. To reuse the logic in a module from other modules, you can export the component's interfaces.

WebSphere Education

Library components

- Libraries are project types for storing artifacts that are shared between several modules
- Libraries contain the following artifacts:
 - **Dependencies:** Are used to include other libraries and predefined resources
 - **Integration logic:** Contains artifacts that do specific tasks (mediation subflow and business calendar)
 - **Data:** Business objects and business vocabulary
 - **Interfaces:** Service interfaces and operations
 - **Transformations:** Contains data (XML) maps and relationships
- Libraries are not runnable applications
 - No assembly diagram



© Copyright IBM Corporation 2016

Figure 4-25. Library components

WB8601.2

Notes:

A library project is a business integration project type that is used for storing artifacts that are shared between multiple modules. Unlike a module project, a library project is not a deployable unit to the IBM Process Server runtime environment. Another important difference between a module and library project is the types of artifacts that can be contained by each project type. Specifically, library projects contain interfaces, business objects, and graphs, Extensible Markup Language (XML) maps, business calendars, and relationships. Library projects do not include other types of business integration resources such as SCA components.

A library project is created to store artifacts that are shared between module projects. If a module is dependent upon a particular library project, then add that project to the dependency list for the appropriate module by using the dependency editor.

Creating modules and libraries

- In the Business Integration perspective, you create an SCA library, a module (also called a business integration module), or a mediation module
- Modules and libraries contain multiple SCA artifacts that are grouped according to type
- Libraries are projects that are used to store shared resources
 - To access libraries, add them to module dependencies
- Integration modules provide the business services, and mediation modules provide the connectivity logic
 - Mediation flows and business services are modeled as SCA components
 - SCA components are wired together in the assembly diagram to form applications

© Copyright IBM Corporation 2016

Figure 4-26. Creating modules and libraries

WB8601.2

Notes:

When you are working in the Java development perspective, the first thing you create is a “project.” In the Business Integration perspective, you create a module, mediation module, or library.

Library deployment

- Deploying libraries in modules (default setting)
 - A copy of the library JAR file is included in each module that uses it
 - After deployment, if the library resources change, modules that use the library must be updated
- Deploying libraries globally
 - The library is exported as a deployable JAR file
 - The resource references for a module are configured to use the global library by using the module deployment editor
 - This option saves memory, but you must deploy each library independently

▼ Sharing Across Runtime Environments

Specify how this library will be shared when it is running on the server. [More...](#)

Module A copy of this library will exist on the server for each module that uses it.

Global The library will be shared among all modules that are running. This option will be more memory efficient when many modules need to use this library.

[Instructions for runtime installation.](#)

© Copyright IBM Corporation 2016

Figure 4-27. Library deployment

WB8601.2

Notes:

Libraries are deployed along with the projects that depend on them so that the resources they offer are available at run time. You can deploy a library to the runtime environment in two ways. In most cases, it is best to deploy a library with the modules that have dependencies on it. In this case, a copy of the library is made for each module that uses it. If you have a large library that many modules use, you can save memory use by choosing to create a global library. You deploy global libraries independently, but they allow modules to share their resources without needing to create copies. Choose whether a library must be deployed with the module or as a global library by selecting the appropriate radio button under **Sharing Across Runtime Environments** in the dependency editor.

If you choose to deploy a library without a module, you must configure a shared library. You can find instructions on how to configure a shared library from the technical notes available on the support website.

WebSphere Education

IBM.

Business Integration view

- Primary view for managing and viewing business integration resources
- Provides logical grouping of resources
 - Artifacts not essential for business integration development are not visible
- Logical resources do not necessarily have a one-to-one relationship with a physical resource

Figure 4-28. Business Integration view

WB8601.2

Notes:

WebSphere Education

IBM

Dependency editor

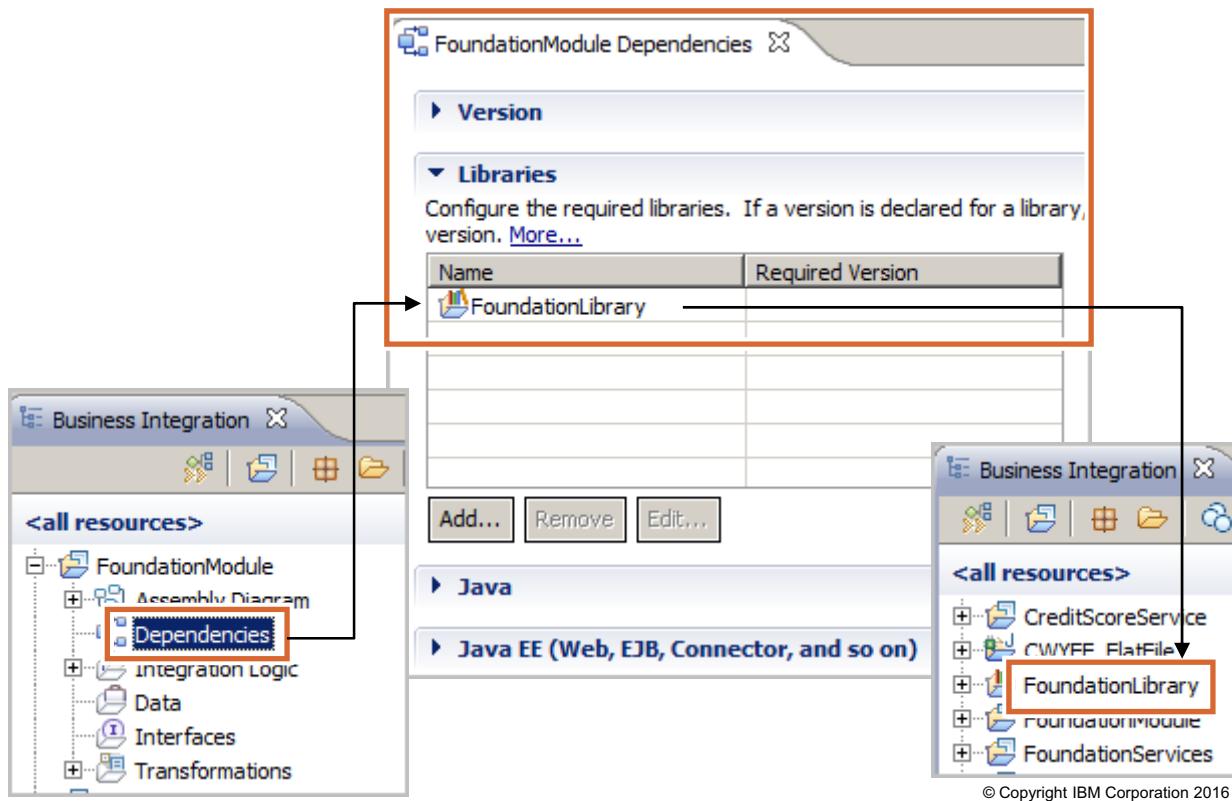


Figure 4-29. Dependency editor

WB8601.2

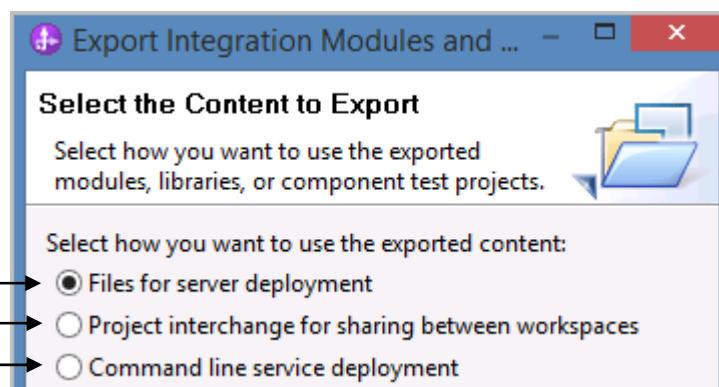
Notes:

Libraries offer a place to store resources that more than one module can share. Libraries are associated with modules through dependencies. You use the dependency editor to manage these dependencies.

Packaging modules

- In IBM Integration Designer, publishing modules to the test runtime environment automatically packages them in enterprise archive (EAR) files and installs them as enterprise applications
- Modules that are created in IBM Integration Designer can be packaged for use outside the test environment:
 - You can export a module as an EAR file for remote deployment
 - You can use the `serviceDeploy` command-line tool to create an installable EAR file that includes components outside your workspace (`serviceDeploy` is covered in a later unit)
 - You can export project interchange files for exchange between workspaces

- Exports as an EAR file
- Exports as a project interchange file
- Exports as a compressed file



© Copyright IBM Corporation 2016

Figure 4-30. Packaging modules

WB8601.2

Notes:

You can export mediation modules as EAR files for deployment or serviceDeploy for deploying files from the command line.

In IBM Integration Designer, from the taskbar, click **File > Export > Business Integration > Integration Modules and Libraries**. The export dialog box provides the steps to complete the export.

Module deployment

- When a module is built and packaged, it is the basic unit of deployment
 - Integration modules are packaged in EAR files as SCA modules
 - The EAR file is deployed to the runtime environment
- The EAR file can contain other components:
 - Java EE projects (EJB modules, and web modules)
 - Java projects
 - Dependent libraries
- A library might be included as a JAR file (if not deployed globally)

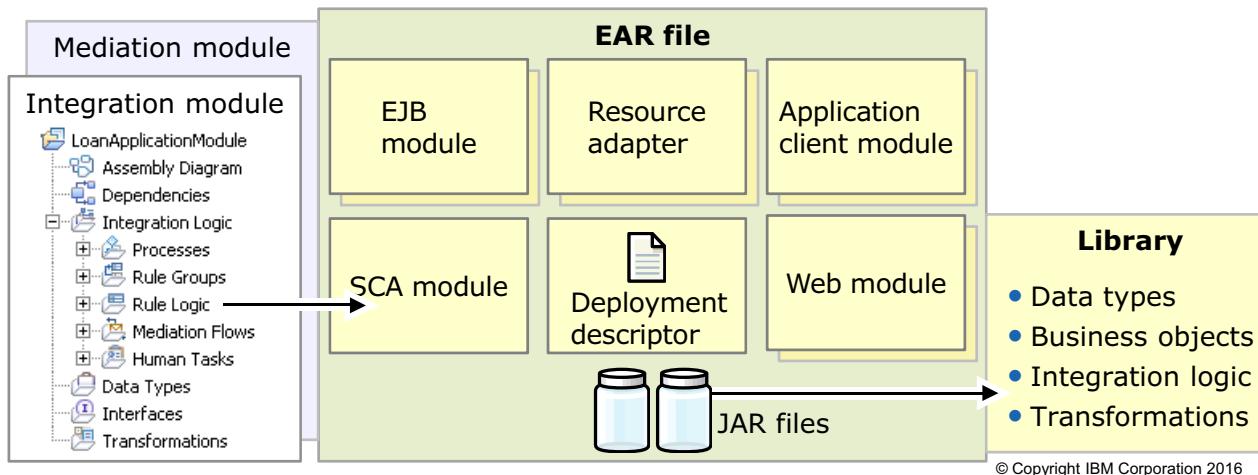


Figure 4-31. Module deployment

WB8601.2

Notes:

Enterprise archive (EAR) files are compressed archive files that contain Java EE application components such as web applications (packaged in WAR files) and EJB beans. Java EE application containers import EAR files into the application environment at run time and deploy the components within each EAR file as separate web applications.

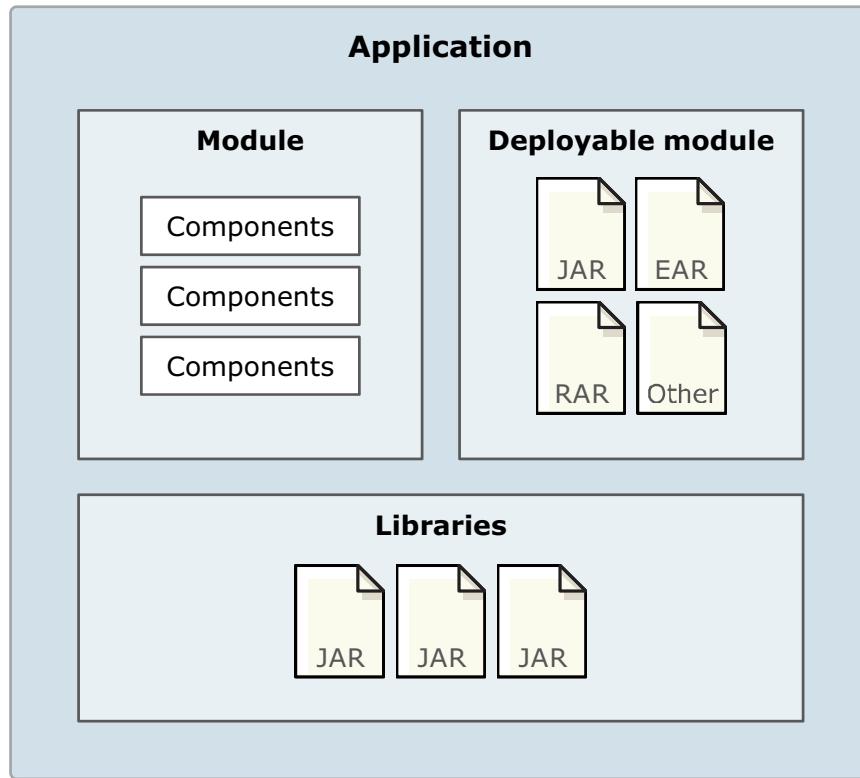
A module is a business integration project type for developing SCA-based applications. A module is a basic unit of deployment to the IBM Process Server runtime environment, and as such, is packaged in an EAR file. The EAR file contains all the SCA-based artifacts (packaged in a JAR file), Java EE projects (web modules and EJB modules), dependent Java projects, and any dependent libraries (packaged as JAR files). This module can contain all the resources that are used in the service, but these resources are private and can be used within the module only.

Two important project types are available to you when doing business integration development in IBM Integration Designer. A module project represents a basic unit of deployment, and encapsulates SCA resources (SCA module), Java EE projects, Java projects, and dependent libraries.

A library project is another type of business integration project. The library project holds artifacts that can be shared between multiple modules. Unlike the module project, a library project is not a

deployable unit. At deployment time, Integration Designer libraries are transformed into utility JAR files and packaged in the applications to be run. At run time, the library is deployed individually with each module that is dependent upon it. At run time, the library is not shared unless it is deployed globally.

Application, module, and library relationship



© Copyright IBM Corporation 2016

Figure 4-32. Application, module, and library relationship

WB8601.2

Notes:

When a module needs an artifact, the server locates the artifact from the EAR class path and loads the artifact unless it is already loaded into memory. The slide shows how an application contains components and related libraries.



Using the process center perspective

- Use the repository in IBM Process Center to share artifacts with other developers
- When an application is built, it can be placed in the repository
- The Process Center perspective in IBM Integration Designer might be used to:
 - Access the repository
 - Import process applications and toolkits
 - Get updates from the repository and send updates to the repository

The screenshot shows the 'Toolkits' tab selected in the top navigation bar of the IBM Integration Designer interface. Below the tabs, there is a search bar with the placeholder 'Sort By: Recently Updated' and a dropdown menu set to 'All'. The main area displays a list of four applications:

	Application Name	Last updated by
	Account Verification Services (AVS101)	Last updated on 7/2/13 by depadmin
	Coaches (SYSC)	Last updated on 6/20/13 by depadmin
	System Data (TWSYS)	Last updated on 6/20/13 by depadmin
	Content Management (SYSCM)	Last updated on 6/20/13 by depadmin

© Copyright IBM Corporation 2016

Figure 4-33. Using the process center perspective

WB8601.2

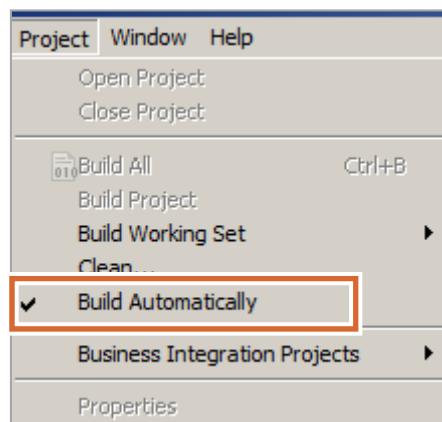
Notes:

IBM Process Center is presented in a later unit.



Building projects

- When artifacts are created and assembled, you build the projects for testing and deployment
- IBM Integration Designer projects are automatically built (compiled) by default when they are saved in the workspace
 - Click **Project > Build Automatically** to toggle automatic builds



© Copyright IBM Corporation 2016

Figure 4-34. Building projects

WB8601.2

Notes:

Artifacts in IBM Integration Designer and Process Designer

	IBM Integration Designer	IBM Process Designer
Container for integration artifacts	Module , which includes: <ul style="list-style-type: none"> Integration logic (BPEL processes, human tasks, and business rules) Data and interfaces Transformations 	Process application , which includes: <ul style="list-style-type: none"> Processes (BPD, human tasks, and rules) Data and services
Container for shareable artifacts	Library , which includes: <ul style="list-style-type: none"> Integration logic Data and interfaces Transformations Web service ports 	Toolkit , which includes: <ul style="list-style-type: none"> Processes Data and services
Container for mediation services	Mediation module , which includes: <ul style="list-style-type: none"> Mediation flows 	N/A

© Copyright IBM Corporation 2016

Figure 4-35. Artifacts in IBM Integration Designer and Process Designer

WB8601.2

Notes:

Process applications and toolkits in IBM Process Designer are analogous to modules and libraries in IBM Integration Designer. Some similarities include:

- Process applications (like modules) are deployed to the server.
- Toolkits (like libraries) are not deployed to the server.
- Process applications contain business process modeling artifacts.
- Process applications have dependencies on any number of toolkits.
- Toolkits have dependencies on other toolkits.

Modules and libraries contain multiple SCA artifacts that are grouped by type. Libraries are projects that are used to store shared resources and are accessed by adding them to module dependencies. Integration modules provide the business services, and mediation modules provide connectivity logic. Mediation flows and business services are modeled as SCA components. SCA components are wired together in the assembly diagram to form applications.

4.3. Documentation and organization capabilities

Documentation and organization capabilities



© Copyright IBM Corporation 2016

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

10.1

Figure 4-36. Documentation and organization capabilities

WB8601.2

Notes:



Documentation and organization capabilities

- Document generation creates a project resource summary
 - Generates reference documentation for your modules and libraries
- Documentation generation also documents selected resources and the resources that they reference
- Artifacts can be exported as images
 - Right-click the artifact, and then export as an image
 - JPEG, PNG, SVG, or TIFF formats
 - Image file name is generated from the project file name
 - Share a picture of the editor without exchanging project interchange files
 - Facilitates documentation

<p>Documentation Generation</p> <p>Specify the properties for the documentation.</p> <hr/> <p>Author: <input type="text" value="John Doe"/></p> <p>Title: <input type="text" value="reportFile"/></p> <p>Location: <input type="text" value="C:\OpeningAccountService_ReportFile.pdf"/></p>	<p>Export as Image to File System</p> <p>Export artifact as an image to the file system</p> <hr/> <p>Format: <input type="text" value="JPEG Image"/></p> <p>Location: <input type="text" value="C:\Documents and Settings\Administrator\Desktop"/></p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

© Copyright IBM Corporation 2016

Figure 4-37. Documentation and organization capabilities

WB8601.2

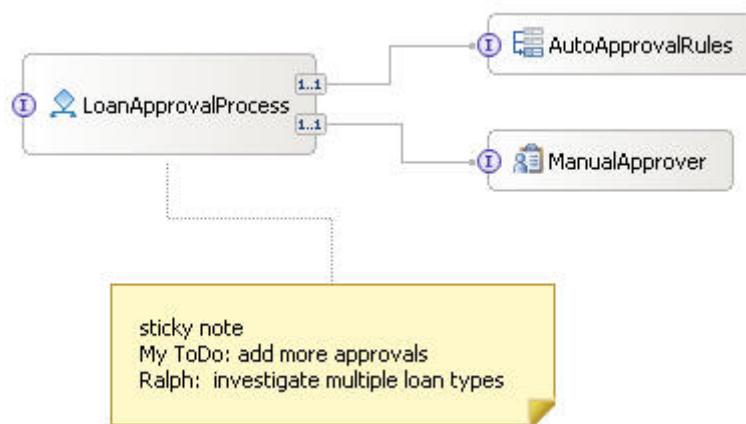
Notes:

You can use the report generation feature to create a summary of the resources in your project by right-clicking the project in the Business Integration view and clicking **Generate documentation**. With the report generation process, you can customize the types of resources in your module or library that are documented in the final report. The report provides information about resources such as business objects, interfaces, human tasks, and business processes that are included in the target project.

You can export many artifacts as images, including the assembly diagram, business objects, business processes, human tasks, and others. If **Export as image** is enabled on the menu, you can capture the artifact as an image.

Documentation: Sticky notes

- Common mechanism to share, and capture text
 - Ideas, reminders, tasks, notes, and URLs
- Available in:
 - Assembly editor, Integration Solution editor, BPEL editor, the mediation flow editor, and the business state machine editor



© Copyright IBM Corporation 2016

Figure 4-38. Documentation: Sticky notes

WB8601.2

Notes:

Sticky note information is persisted in a `.stickyboard` extension file. Sticky notes are saved and exported along with the project interchange file. Dotted lines associate a sticky note with a component. You cannot connect a note to an interface, a reference, or a wire.

Sticky notes do have some limitations:

- Spaces in hyperlink paths are not supported.
- Notes from IBM Process Designer do not import into IBM Integration Designer.
- Notes do not move when arranging contents automatically.
- Sticky notes are used in four editors of IBM Integration Designer.

Another useful feature in sticky notes is the ability to have four different types of hyperlinks: web URL, FTP, mailto, and local file system. The web URL and FTP hyperlinks start your default browser client, and the mailto hyperlink brings a new email into your default mail client.



Documentation: Java task tags

- Highlighting for Javadoc task tag
 - Organizes work for developers
 - Defined in the Java compiler task tag preferences
 - Defaults: FIXME, and TODO
 - Can add a custom task tag (URGENT)
- Use the Tasks view or the Markers view to see tags in the list

Default task tag examples:
TODO: Add code to the loan approval
FIXME 1. Do this task first
FIXME 2. Do this task next

Custom task tag example:
URGENT: Important announcement

The screenshot shows the 'Task Tags' configuration dialog and the 'Tasks' view in the IDE.

Task Tags Dialog:

Tag	Priority
FIXME	High
TODO (default)	Normal
URGENT	High

Tasks View:

	Description	Resource	Path	Location
!	FIXME 1. Do this first	sca.module	LoanApplicationM...	Unknownr
!	FIXME 2. Do this next	sca.module	LoanApplicationM...	Unknownr
!	URGENT very important announcement.....	sca.module	LoanApplicationM...	Unknownr
	TODO add code to loan approval	sca.module	LoanApplicationM...	Unknownr
	ToDo Create	styles-IBM.css	CustomerUI/Web...	line 883

Figure 4-39. Documentation: Java task tags

WB8601.2

Notes:

Task tag highlighting is added to provide automatic highlighting when typing a predefined task tag in a sticky note. Custom task tags can be added by browsing to **Window > Preferences > General > Editors > Structured Text Editors > Task Tags** in the menu bar. The TODO task tag is a default.

Another default task tag is XXX, which is intended as a placeholder for a custom tag.

Organization: Integration solution (1 of 2)

- Diagram to display how modules, mediation modules, libraries, and other dependent projects connect
 - Objects can be in more than one solution
 - Workspace can contain more than one integration solution
 - Integration solutions can be focused and hidden
- Development time concept only
 - Not deployed (referenced applications are deployed)
 - Acts like a viewer by using Adobe Flash Player to display projects



© Copyright IBM Corporation 2016

Figure 4-40. Organization: Integration solution (1 of 2)

WB8601.2

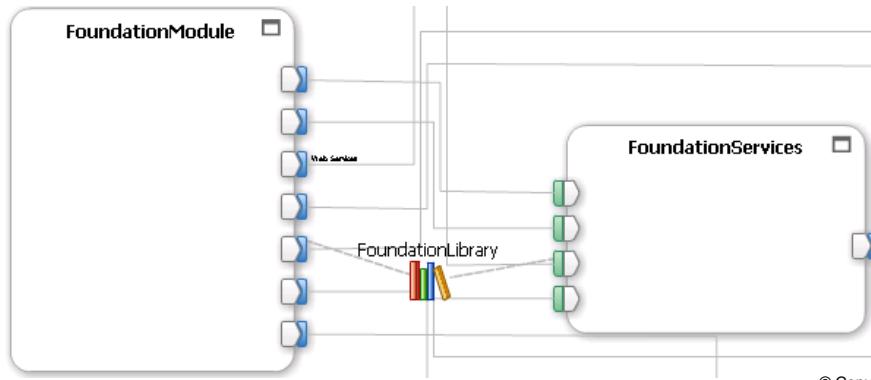
Notes:

An integration solution is used to organize and display how modules, mediation modules, libraries, and other dependent projects connect. Integration solutions are presented in IBM Integration Designer as though they were another type of project. The integration solutions section helps users organize multiple modules, mediation modules, libraries, and other dependent projects into one scope. This concept is valid only during development. The servers are not aware of the integration solution. The servers still view an application as a set of .ear files.

The integration solution has connections to team environments, like CVS or ClearCase, where a developer can check out all the projects that are associated with an integration solution in one action. You can use the integration solution diagram to visualize how modules, mediation modules, libraries, and other dependent projects connect. Think of the integration solution diagram more as a viewer than as an editor. However, it can open editors, can display transaction highlighting, updates automatically to changes in projects that are encapsulated in the integration solution, and has several other editor-like functions.

Organization: Integration solution (2 of 2)

- Transaction highlighting
 - Like an assembly diagram
 - Shows transactions in and between modules (in green)
- Menu
 - Test a module by right-clicking the diagram
 - Open editors from a solution diagram
- Export projects that are included in a solution as an integration module
 - .EAR or a project interchange file
 - Export as an image for documentation purposes (.PNG only)
- Add and remove integration solution projects from the server
 - Publishes all referenced applications



© Copyright IBM Corporation 2016

Figure 4-41. Organization: Integration solution (2 of 2)

WB8601.2

Notes:

In the Integration Solution editor, you can display 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 can also display and highlight the transaction path from a specific component to the other components with which it interacts. If you choose, you can remove the highlighting from the transaction path.

You can also choose to display an integration overview, which is a small image of the current and entire integration solution. The overview is displayed in the lower-right corner of the integration solution editor.



Unit summary

- Describe the purpose and business value of IBM Integration Designer
- Define the types of projects that are developed in IBM Integration Designer: modules, mediation modules, and libraries
- Describe how to create and deploy applications in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 4-42. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. True or False. If you install a unit test environment (UTE) locally, a new process server instance is automatically created.
2. True or False. IBM Installation Manager is used to update IBM Integration Designer and the server runtime test environment.
3. True or False. An integration module is a project that is used to store resources that two or more modules share.

© Copyright IBM Corporation 2016

Figure 4-43. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. True.
2. True.
3. False. A library (not integration module) is a project that is used to store resources that two or more modules share. An integration module is a project that consists of SCA components that are wired together in an assembly diagram to form applications.

© Copyright IBM Corporation 2016

Figure 4-44. Checkpoint answers

WB8601.2

Notes:

Exercise 2



Exploring IBM Integration Designer,
part I

© Copyright IBM Corporation 2016

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

10.1

Figure 4-45. Exercise 2

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Describe IBM Integration Designer capabilities and preferences
- Use the help menu to search for assistance
- Use the Installation Manager to find product updates
- Browse the Business Integration perspective and views
- Examine the modules and libraries of a business integration project
- Browse the IBM Integration Designer graphical editors
- Use the IBM Integration Designer documentation capabilities
- Use the Task Flows view

© Copyright IBM Corporation 2016

Figure 4-46. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 2

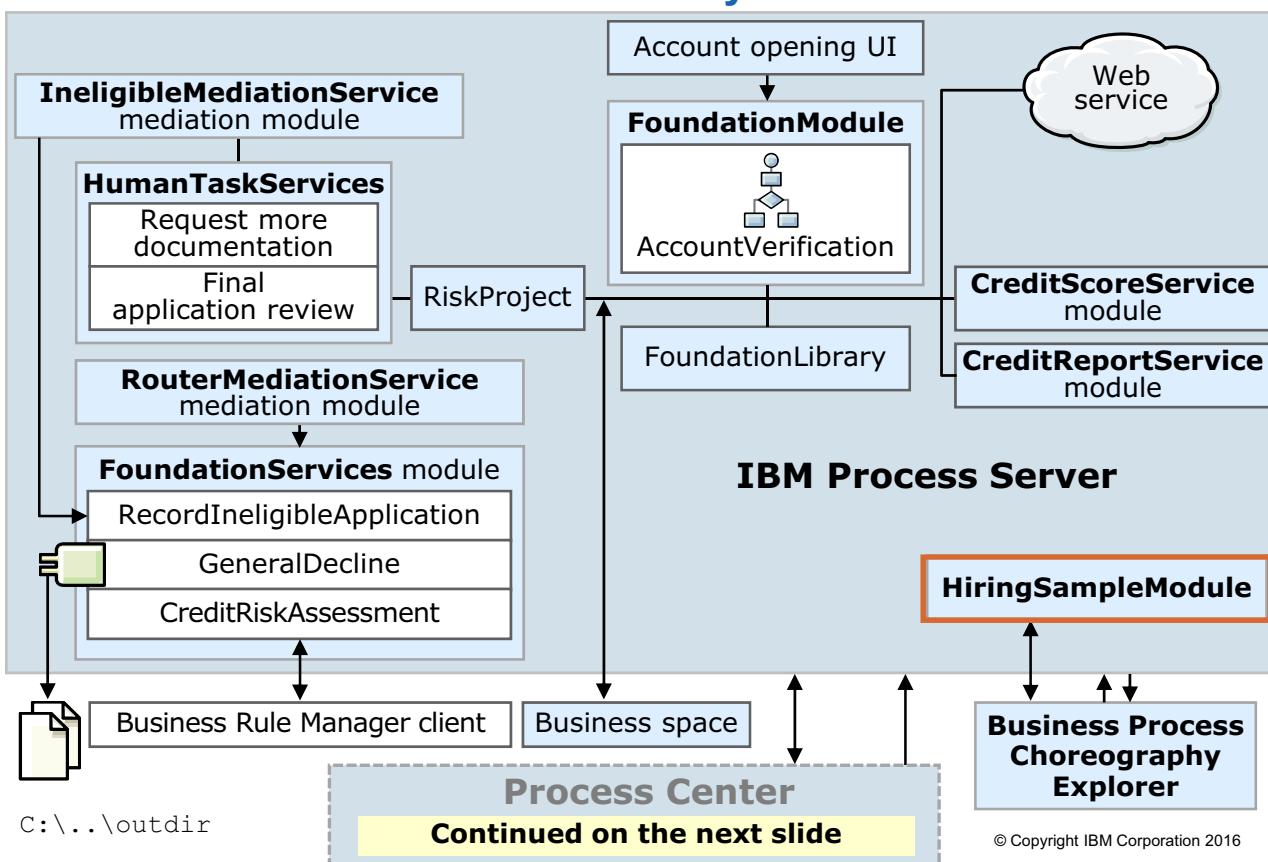


Figure 4-47. Course business scenario: What you work with in Exercise 2

WB8601.2

Notes:

In this exercise, you explore HiringSampleModule.



Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

- 1. HiringSampleLibrary
- 2. HiringSampleModule

<all resources>

- HiringSampleLibrary
- HiringSampleModule
 - Assembly Diagram
 - Dependencies
 - Integration Logic
 - BPEL Processes
 - OpenNewPosition
 - Human Tasks
 - Data
 - Interfaces
 - Transformations

© Copyright IBM Corporation 2016

Figure 4-48. Components that are required for Exercise 2

WB8601.2

Notes:

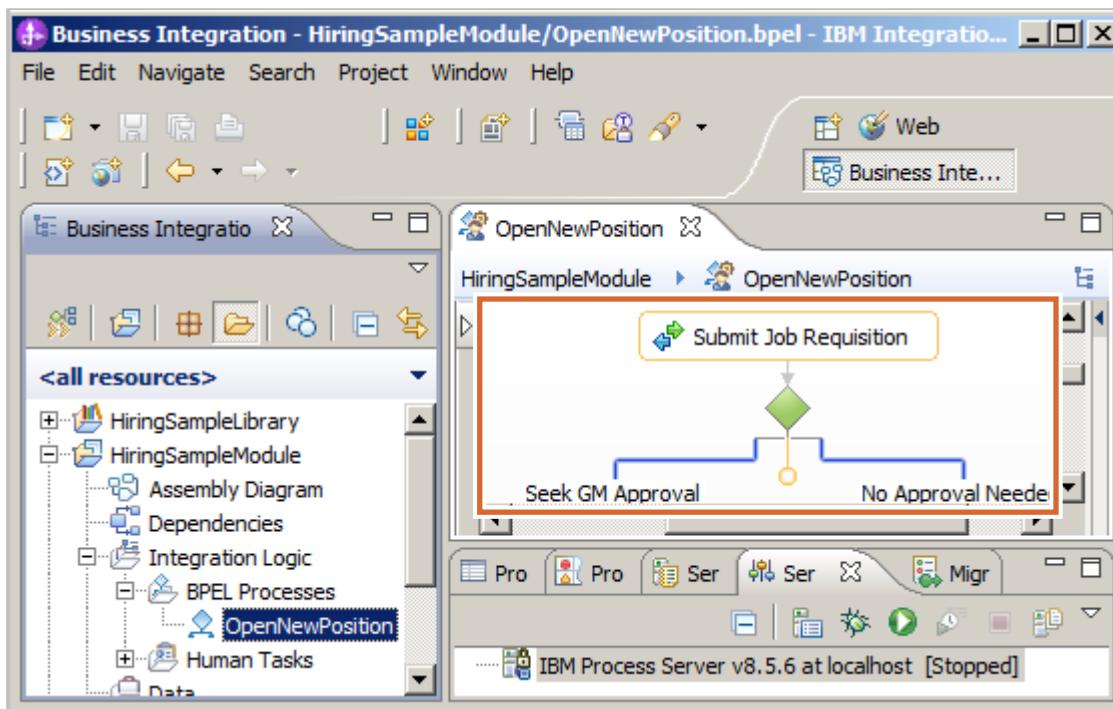
HiringSampleLibrary and HiringSampleModule are two of the artifacts in IBM Integration Designer that are already built for you. In this exercise, you explore these artifacts.

4-64 Developing Applications in IBM BPM Advanced V8.5.7 - I

© Copyright IBM Corp. 2016

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

Explore IBM Integration Designer in Exercise 2



© Copyright IBM Corporation 2016

Figure 4-49. Explore IBM Integration Designer in Exercise 2

WB8601.2

Notes:

You view the HiringSample process in IBM Integration Designer. This process is the same BPD that you explored in Exercise 1 by using the Process Center. The difference is that in this exercise you use the BPEL version of the HiringSample process. For differences between BPDs and BPEL, see the earlier unit.



References

- IBM Integration Designer home page:
 - <http://www.ibm.com/software/products/us/en/integration-designer>
- Eclipse home page:
 - <http://www.eclipse.org>
- Eclipse V3.6.2 (“Helios”) help system:
 - <http://help.eclipse.org/helios/index.jsp>
- IBM Rational Application Developer overview:
 - https://www.ibm.com/support/knowledgecenter/SSRTLW_8.5.5/com.ibm.rad.nav.doc/helpindex_rad.html
- IBM Knowledge Center, Business Process Manager documentation:
 - http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbpm.main.doc/kc-homepage-bpm.html

© Copyright IBM Corporation 2016

Figure 4-50. References

WB8601.2

Notes:

Unit 5. Service Component Architecture overview

What this unit is about

In this unit, you learn about the service component architecture (SCA), its constituent components, and its relevance to SOA, BPM, and IBM Integration Designer.

What you should be able to do

After completing this unit, you should be able to:

- Identify the purpose and business value of using SCA
- Identify and explain the function of the individual parts of an SCA component
- Define the concept of a service module and explain each of the service module components
- Describe how to test SCA applications in IBM Integration Designer
- Describe the features that are available in the IBM Integration Designer test environment

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Identify the purpose and business value of using SCA
- Identify and explain the function of the individual parts of an SCA component
- Define the concept of a service module and explain each of the service module components
- Describe how to test SCA applications in IBM Integration Designer
- Describe the features that are available in the IBM Integration Designer test environment

© Copyright IBM Corporation 2016

Figure 5-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to Service Component Architecture components
- Overview of SCA modules
- Testing SCA applications

© Copyright IBM Corporation 2016

Figure 5-2. Topics

WB8601.2

Notes:

5.1. Overview of SCA modules

Overview of SCA modules



© Copyright IBM Corporation 2016

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

10.1

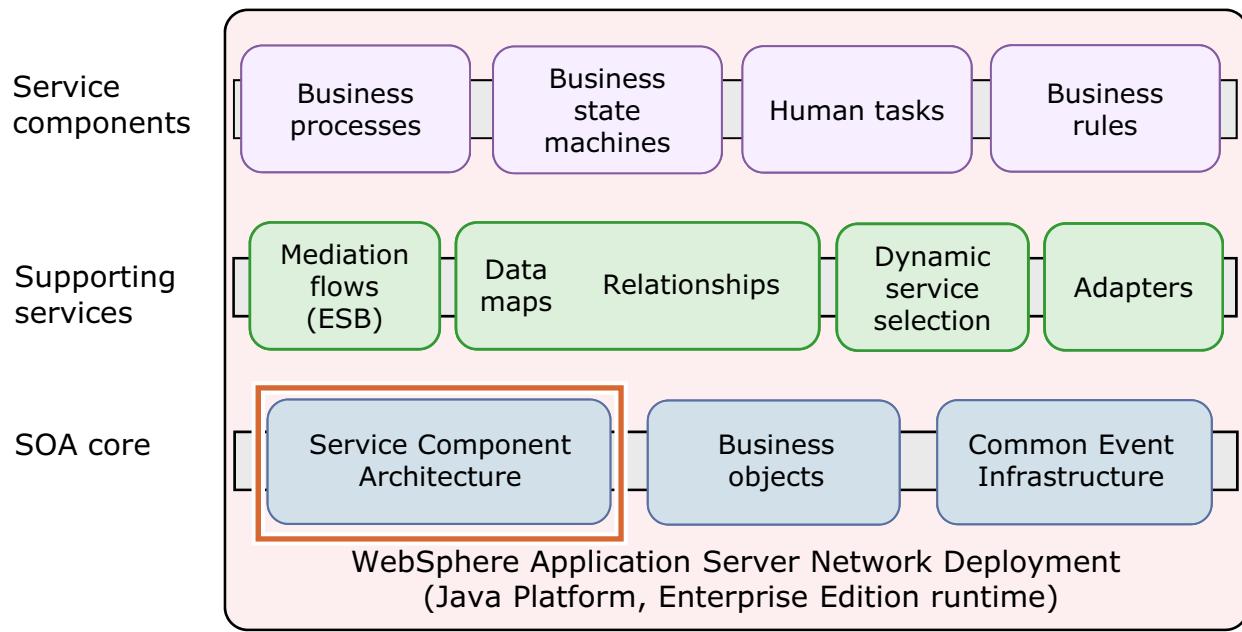
Figure 5-3. Overview of SCA modules

WB8601.2

Notes:

Service Component Architecture

- Service Component Architecture is an SOA core component
- SCA provides the programming model for business integration modules and mediation modules



© Copyright IBM Corporation 2016

Figure 5-4. Service Component Architecture

WB8601.2

Notes:

Beyond the basic WebSphere Application Server functions, the power of IBM Process Server builds on a set of logical services that IBM terms the SOA core. This set of services provides the core foundation on which SOA applications are based. Within the SOA core, the three major components are Service Component Architecture, business objects, and Common Event Infrastructure. Service Component Architecture (SCA) provides a universal invocation model with well-defined interfaces and encapsulation into modules.

Overview and business value of SCA

- SCA is a service-oriented component model that provides a declarative, high-level method of creating relationships between services
 - Service implementation details are not considered
 - When services change, only the declaration is changed, not the mechanical steps in the application code
- SCA provides a single service component abstraction for services that might already be implemented as business processes, Java classes, and mediation flows
 - Abstraction separates “business logic” from “infrastructure logic,” allowing developers to focus on business problems, not infrastructure code
- Without SCA, you must write code to communicate with the services in your SOA-based application
 - This application is not loosely coupled, and is not easy to change
 - You must change application code to respond to infrastructure changes

© Copyright IBM Corporation 2016

Figure 5-5. Overview and business value of SCA

WB8601.2

Notes:

SCA is a service-oriented component model for defining and invoking business services that publish or operate on business data. SCA is aimed at providing a simplified programming model for writing applications that run in a Java EE runtime environment. SCA is based on concepts and techniques that are refinements of existing Java EE technology. SCA is an open standard that OSOA (open service-oriented architecture) manages and includes contributions from companies such as IBM, Oracle, SAP, and Sun.

One of the important aspects of SCA is to provide a separation between application business logic and implementation details. To this end, SCA provides a single abstraction for service types that are already expressed as session beans, web services, Java classes, or business process applications that are written in Business Process Execution Language (BPEL). The ability to separate business logic from infrastructure logic is important to help reduce the IT resources that are needed to build an enterprise application. It gives developers more time to work on solving business problems rather than focusing on the details of which implementation technology to use.

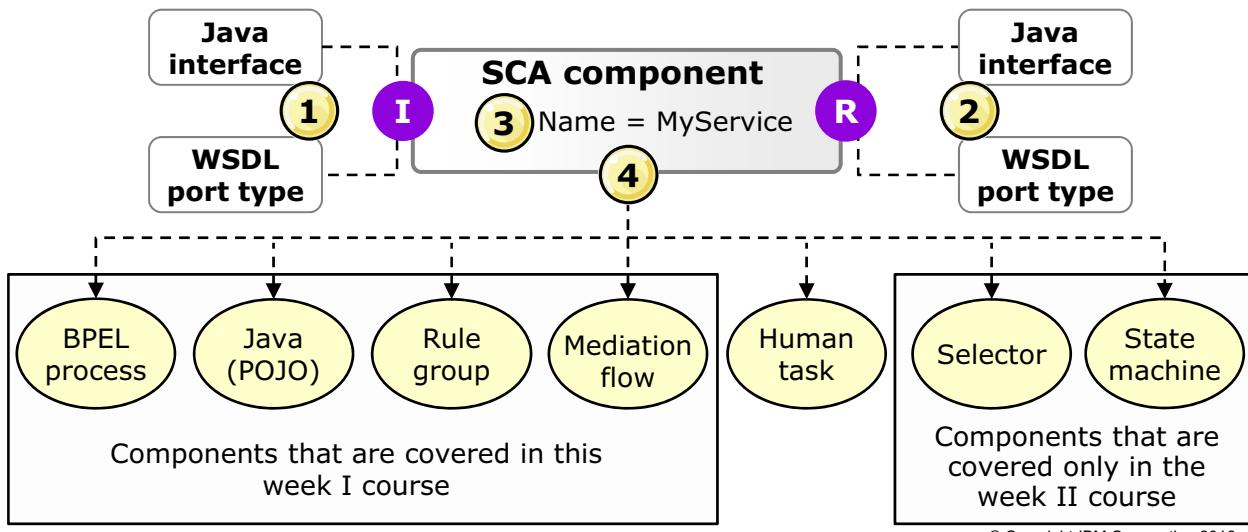
The Service Component Definition Language (SCDL) provides the basis of SCA. SCDL is an XML-based definition language, and is used to define all SCA artifacts in a project. IBM Integration Designer automatically generates the appropriate SCDL definitions when building an SCA-based

application. 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.

It is also important to understand the different types of artifacts that can be defined by using SCDL. The various artifact types that exist in SCA were designed to support some of the basic requirements of a service-oriented architecture. First, SCA needs a mechanism for defining a basic service component. When you have a mechanism for defining service components, it is important to make these services available to clients that are either inside or outside the current SCA module. In addition, a construct must be designed to import and reference services external to the current SCA module. Finally, SCA provides constructs for composing services and modules into larger applications. In the remainder of this unit, you learn about each of these SCA artifacts and how they can be composed into larger applications.

SCA components

- SCA components are discrete units of business logic that contain:
 1. **Interfaces:** Are used to call the component
 2. **References:** Are used to call other components
 3. **Name:** Unique in the SCA module
 4. **Implementation:** A representation of the service type (the physical implementation is separate from the SCA component)



© Copyright IBM Corporation 2016

Figure 5-6. SCA components

WB8601.2

Notes:

Each service component must have a unique name within the SCA module, and it must match the file path relative to the module root. In Windows environments, keep component names as short as possible to avoid path limits. Component names also must follow a standard naming convention to allow reuse.

The service component definition is included in a file that is called `<SERVICE_NAME>.component`. Each service component can have zero or more interfaces that are associated with it; these interfaces can either be Java or WSDL `portType` interface definitions. The interfaces that are associated with a service component can support either a synchronous or an asynchronous interaction style with clients that call the service.

Each service component can be implemented in various ways, which the implementation definition specifies. Service components can invoke other service components or imports that are defined in the current service module. When a component invokes another component or import, the appropriate reference must be defined to indicate which service is used. Often this type of reference is "inline" in the service component definition (although it might alternatively be placed in a stand-alone reference file). Each service component definition can have zero or more references to other services called by the service component that is being defined.

- **SCA** is a universal model for business services that publish or operate on business objects.
- **SCA services** are represented as an **SCA component**.
- A **service** can have many different implementations, for example: Java, BPEL, and state machines.
- Services focus on the business purpose, rather than on the infrastructure technology.
- The same implementation artifact can be reused many times:
 - The **implementation** is the **template**.
 - The **component** is the **configuration**.
- A **service interface** is defined through either a **Java interface** or a **WSDL portType**.
 - Parameters and return values are described by using **Java classes**, **simple Java types**, or **XML schema**.
 - Parameters that are described in **XML schema** are exposed as **business objects**.
- **Services** can be *composed* of other services and components, allowing for structured reuse.
 - Services reference other services through symbolic links that are called **service references**.
- **Service references** are resolved with **wires**; you use SCA wiring to combine SCA modules in the component assembly editor. Using the assembly editor, the interaction between services can be visually defined.

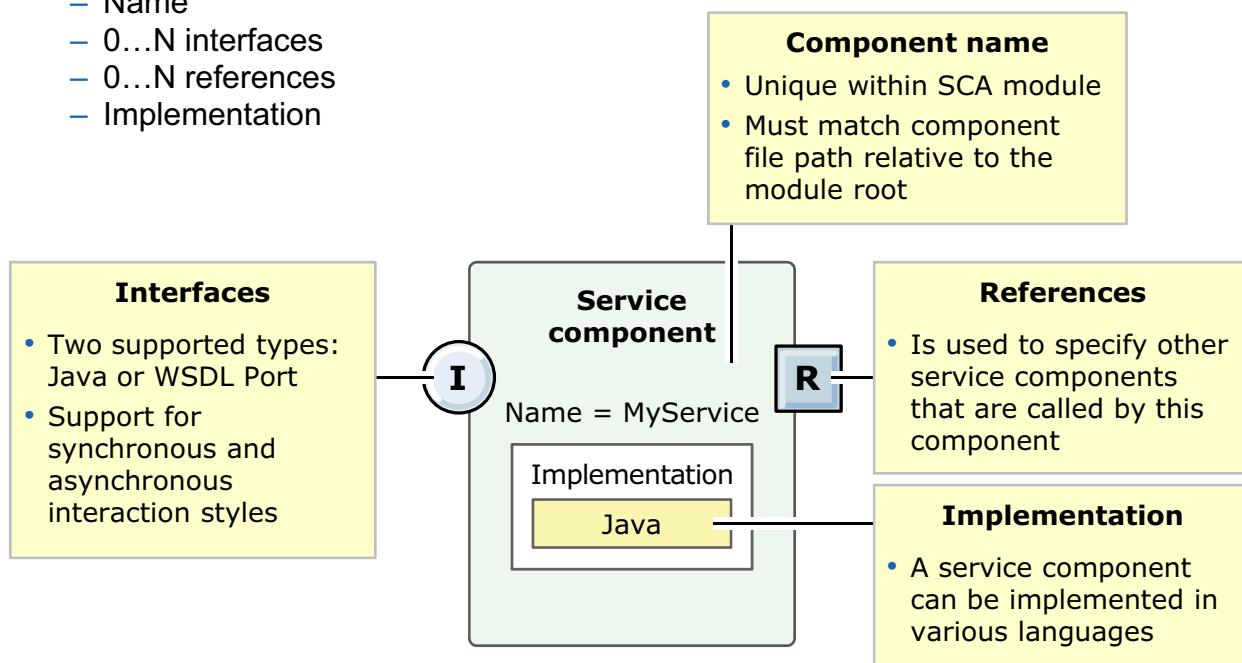
For more information about the current progress of the SCA specification and proposals for other implementation types, see the Apache Tuscany specification at:

<http://incubator.apache.org/tuscany>

SCA component definition

- Component definitions include:

- Name
- 0...N interfaces
- 0...N references
- Implementation



© Copyright IBM Corporation 2016

Figure 5-7. SCA component definition

WB8601.2

Notes:

This slide provides a more detailed look at the service component definition that was introduced earlier in the presentation. Each service component must have a unique name within the SCA module, and it must match the file path relative to the module root. As noted on the previous slide, the service component definition is included in a file called `<SERVICE_NAME>.component`. Next, each service component can have zero or more interfaces that are associated with it, and these interfaces can be either Java or WSDL port type interface definitions.

The interfaces that are associated with a service component can support either a synchronous or an asynchronous interaction style with clients that call the service. This feature is discussed in greater detail in upcoming slides in this presentation. As noted earlier, each service component can be implemented in various ways, and this specification is in the implementation definition. Finally, service components can invoke other service components or imports that are defined in the current service module. In this case, the reference must be defined to indicate which service is used. Often this type of reference is in-lined in the service component definition (although it can alternatively be placed in the stand-alone references file). Each service component definition can have zero or more references to other services called by the service component that is being defined.



SCA component interfaces

- Web Services Description Language (WSDL) provides an industry standard way of describing services in a service-oriented architecture
- WSDL is a way for service providers to describe the basic format of requests to their systems, regardless of the underlying implementation
- WSDL allows a provider to specify the characteristics of a service:
 - The name of the service and the addressing information
 - The protocol and encoding style that is used to access the public operations
 - The operations, parameters, and data types in the service interface
 - The preferred interaction style: synchronous or asynchronous
- WSDL documents are defined in XML
 - An industry standard language
 - Platform and technology independent
 - Capable of describing a wide range of services
 - Service definitions are flexible and extensible
- Interface elements can be structured in one or more WSDL files

© Copyright IBM Corporation 2016

Figure 5-8. SCA component interfaces

WB8601.2

Notes:

WSDL is the key to realizing the concept of services. WSDL is a standard language for describing the capabilities of a component, the operations it can do, the data for those operations, and how to invoke the component. A WSDL document provides a contract that others can use and rely upon. When you have the WSDL document, you can use the capabilities that the service supplies without knowing about the physical implementation of that service.

A WSDL file can be a single file, or it can be split into multiple files that contain **portType**, **binding**, and **service definitions**.

A WSDL document contains the following definitions:

- Service implementation definition:
 - **Port**: A single endpoint that is defined as a combination of a binding and a network address
 - **Service**: A collection of related endpoints
- Service interface definition:
 - **Bindings**: A concrete protocol and data format specification for a particular `portType`
 - **portType**: An abstract set of operations and their associated messages

- **Types:** A container for data type definitions (which are used to describe the messages that are exchanged); uses a type system like XML Schema Definition (XSD)
- **Operation:** An abstract description of an action that the service supports
- **Message:** An abstract, typed definition of the data that is being communicated (a message consists of logical parts, each of which is associated with a definition within a type system)

SCA component references

- A reference is used to specify the target of a service invocation
 - Defined on the calling component or in a stand-alone reference
- Reference definitions include:
 - Name: Is used to look up the appropriate service
 - Multiplicity: Number of services that can be wired to the reference
 - Interface: Interface that is used to invoke the target component
 - Wire: Used in a module assembly to identify the target service component that resolves the reference (a component or import)

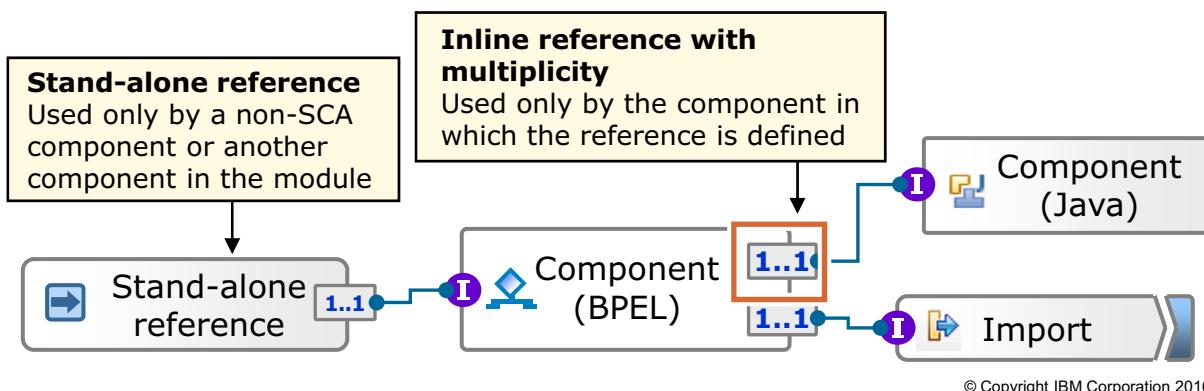


Figure 5-9. SCA component references

WB8601.2

Notes:

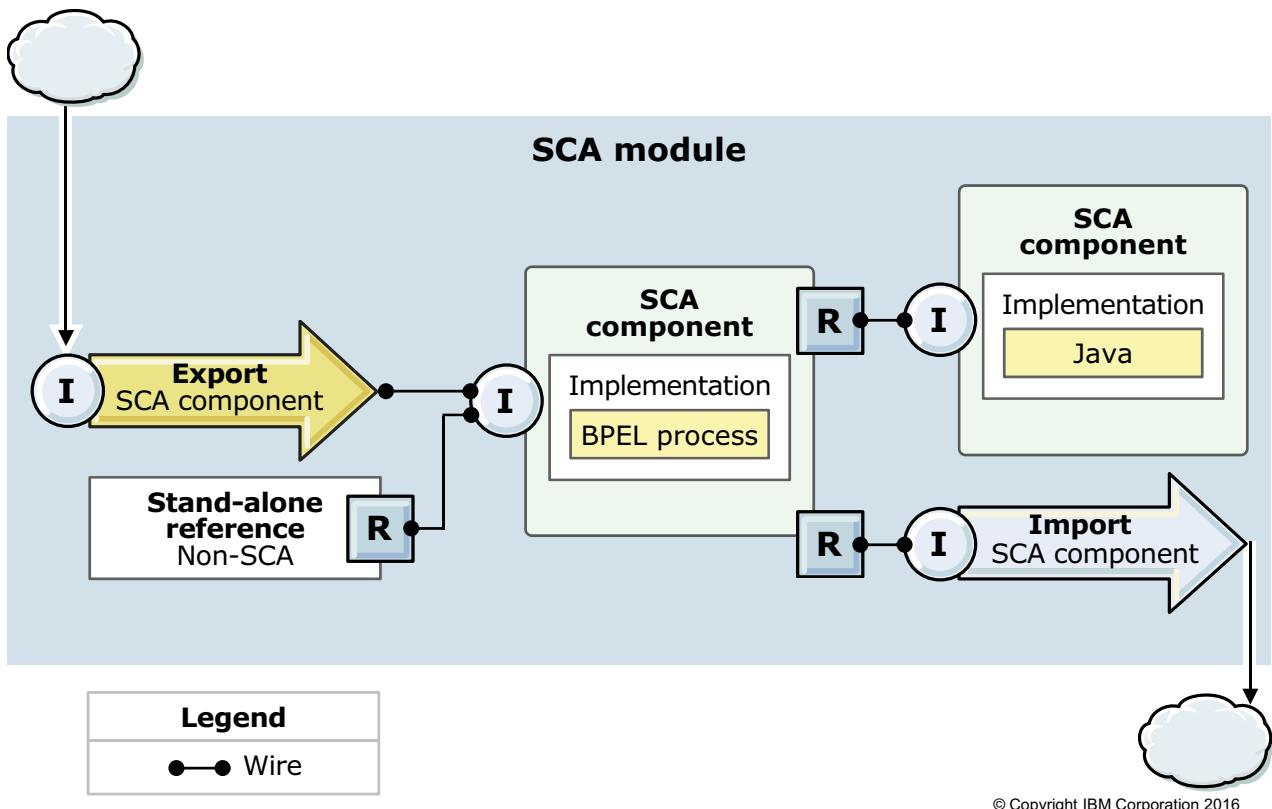
To call a service component, SCA and non-SCA clients need a reference to that service to invoke it. Each reference has a name, and a client uses this name to look up the appropriate service by using the client programming model. In addition to the name, a reference also includes an interface element. The multiplicity for a reference indicates how many wire definitions can name this reference as the source. The wire definition specifies the name of the target service component or import that resolves the reference.

References can be defined in two ways. The first way is to include the reference in the service component definition. In this approach, the references are available only to the service component in which the references are included. Another approach is to include reference definitions in a stand-alone references file. In this approach, a non-SCA client or another component within the module can use the references. An example of a non-SCA component that might use a reference in the stand-alone references file is a user interface component such as a JSP file that needs the ability to invoke a particular service. To invoke the service, the client needs a reference so that it can use the SCA runtime environment to look up the appropriate target.

Multiplicity defines the number of services that can be wired to the reference:

- **1..1** is the simplest and most common case, where the implementation expects that exactly one service is wired. If no service is wired to the reference, the implementation fails at run time. The SCA runtime environment generates a warning during deployment in this case.
- **0..1** caters to an implementation that can tolerate the case where no service is wired to the reference. This case might occur if the service that is being invoked is a logging service, where correct operation of the component occurs even if no logging service is wired.
- **0..n** and **1..n** cater to an implementation that is prepared to handle multiple target services. The two cases differ according to whether the implementation can tolerate having no service that is wired to the reference. **0..n** allows for no service to be wired while **1..n** requires at least one. In both of these cases, the implementation receives a **collection of** service references rather than a single one. It is up to the implementation to invoke each service in turn.
 - **0..n** is typically used in publish/subscribe situations where many services can receive a message from an implementation, but the implementation itself can operate successfully even if none exists.
 - **1..n** might be used in situations where the implementation wants to invoke the same service interface against many services, perhaps to compare the results. An example might be a price comparison for widgets from multiple suppliers. In this example, each wire represents a connection to the widget price query service of a different supplier.

SCA module: Overview



© Copyright IBM Corporation 2016

Figure 5-10. SCA module: Overview

WB8601.2

Notes:

A service module can have zero or more service components included with it. To access these services by a client (SCA or non-SCA), there must exist at least one reference to the service, or the service needs to be exposed with an export.

Imports

A service module can have zero or more imports included with it. An import is used to access services that are outside the current SCA module. When an import is defined, other services from within the module can reference the imported service as if it were a regular service component that was defined in the module.

Exports

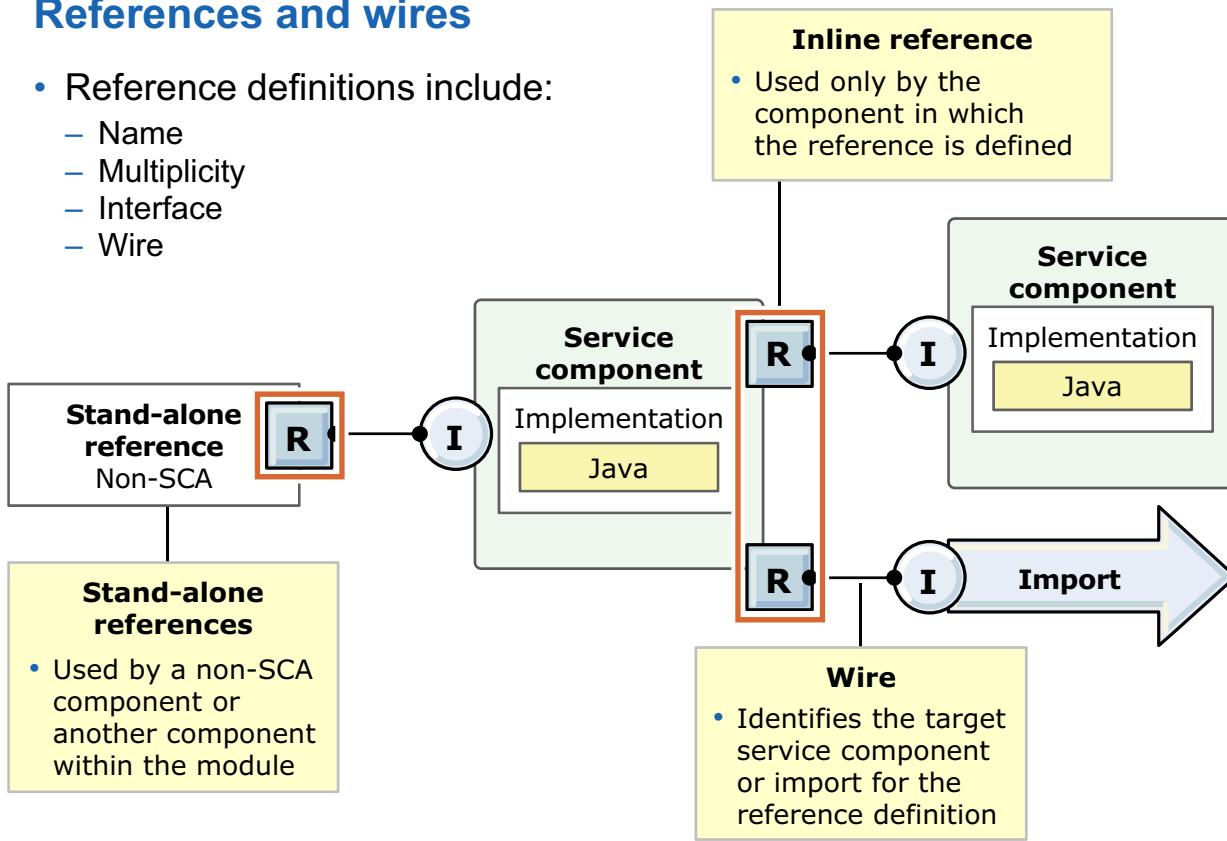
A service module can have zero or more exports included with it. An export is used to expose a particular service to clients outside the current SCA module.

Stand-alone references

A service can include a stand-alone references file that includes references to a service in the module that SCA and non-SCA services can use.

References and wires

- Reference definitions include:
 - Name
 - Multiplicity
 - Interface
 - Wire



© Copyright IBM Corporation 2016

Figure 5-11. References and wires

WB8601.2

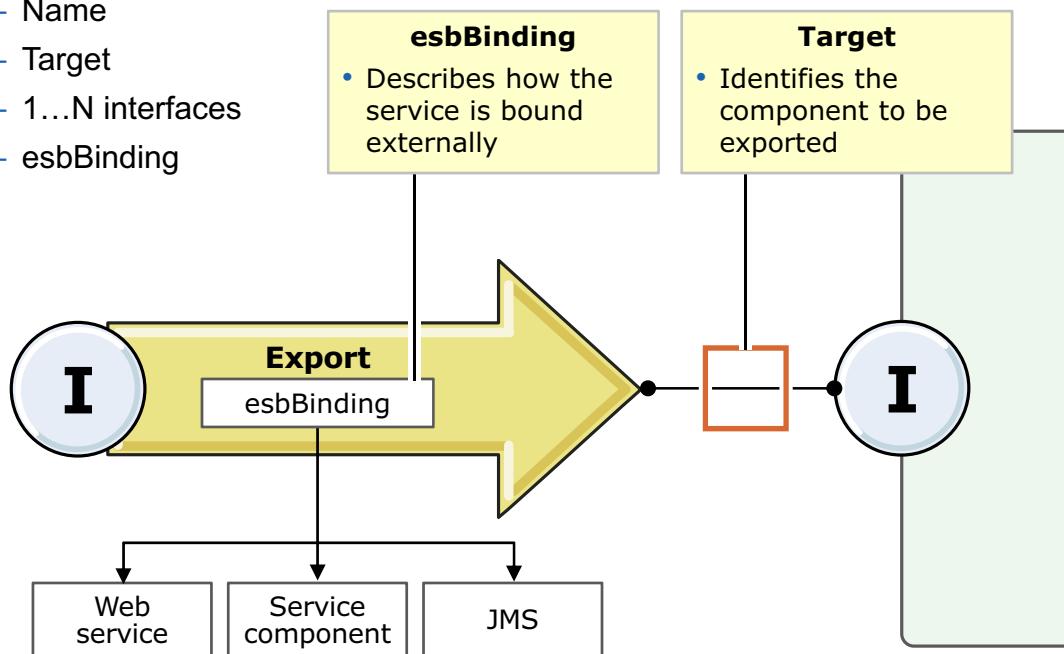
Notes:

This slide reveals some of the details of defining references. Each reference has a name, and this name is used to look up the appropriate service by a client that uses the client programming model. In addition to the name, a reference also includes an interface element. The multiplicity for a reference indicates how many wire definitions can name this reference as the source. Finally, the wire definition specifies the name of the target service component or import that resolves the reference.

References can be defined in two ways. The first way is to inline the reference in the service component definition. In this approach, the references are available only to the service component in which the references are included. Another approach is to include reference definitions within the stand-alone references file. In this approach, a non-SCA client or another component within the module can use the references. An example of a non-SCA component that might use a reference in the stand-alone references file is a user interface component such as a JSP that needs the ability to invoke a particular service. To invoke, the client needs a reference so that it can use the SCA runtime to look up the appropriate service to invoke.

Exports

- An export allows access to services for use outside the current SCA module
- Export definitions include:
 - Name
 - Target
 - 1...N interfaces
 - esbBinding



© Copyright IBM Corporation 2016

Figure 5-12. Exports

WB8601.2

Notes:

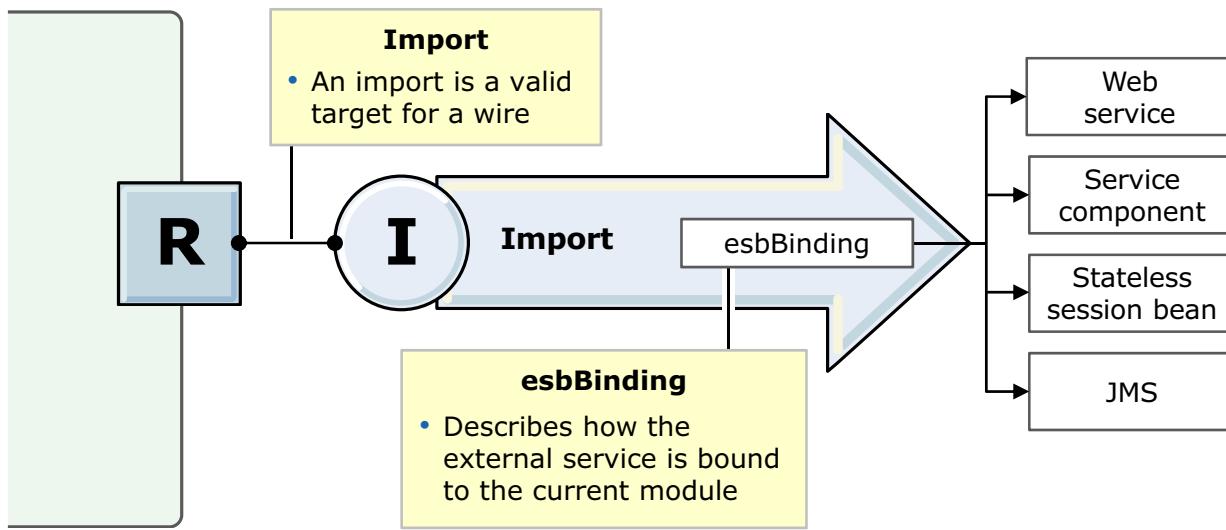
Exports define interactions between SCA modules and service requesters. SCA modules use exports to offer services to others.

An SCA module export needs at least one interface. Export interfaces are abstract definitions that define a set of operations by using Web Services Description Language (WSDL), an XML language for describing web services. An SCA module can have many export interfaces.

Export bindings define the specific way in which service requesters can access an SCA module. Export bindings are concrete definitions that specify the physical mechanism that service requesters use to access a service. Usually, an SCA module export has one binding that is specified. An export with no binding specified is used by the runtime environment as an export with an SCA binding.

Imports

- An import provides access to services outside the current SCA module
- Import definitions include:
 - Name
 - 1...N interfaces
 - esbBinding



© Copyright IBM Corporation 2016

Figure 5-13. Imports

WB8601.2

Notes:

Imports define interactions between SCA modules and service providers. SCA modules use import components to access external services (services that are outside the SCA module) by using a local representation. Import bindings define the specific way that an external service is accessed.

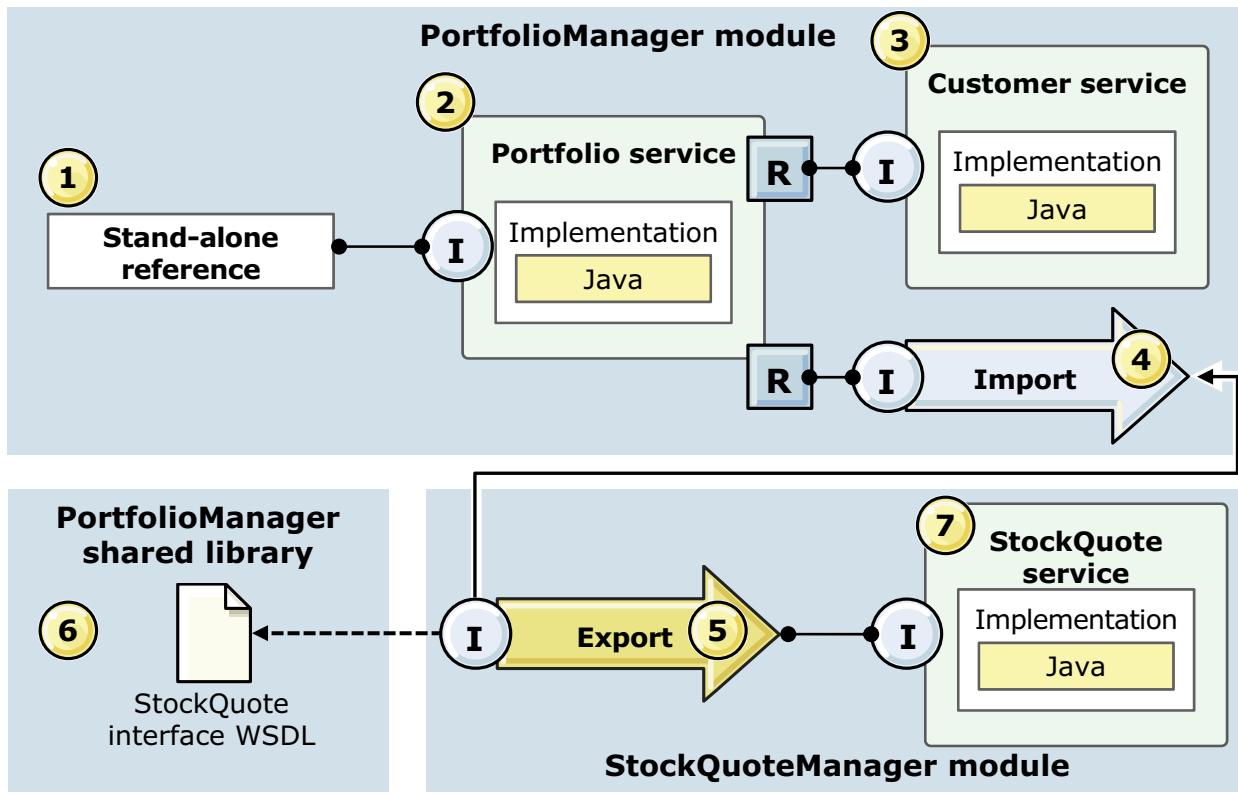
If SCA modules do not need to access external services, they are not required to have imports. Mediation modules usually have one or more imports that are used to pass messages or requests on to their intended targets.

By using an import, you can use functions that are not a part of the module that you are assembling. Imports are used in an application in the same way as local components. Imports have interfaces that are the same as, or a subset of, the interfaces of the remote service that they are associated with so that those remote services can be called.

An SCA module import needs at least one interface, and an SCA module import has a single binding. Import interfaces are abstract definitions that define a set of operations by using Web Services Description Language (WSDL), an XML language for describing web services. An SCA module can have many import interfaces.

Import bindings are concrete definitions that specify the physical mechanism that SCA modules use to access an external service. An import binding describes the specific way that an external service is bound to an import component. For an import that is generated from an export, the binding type of the import is specified for you. If you are using the palette in the assembly editor to create the import, you must specify a binding type for the external service to test it.

Example: Using imports and exports



© Copyright IBM Corporation 2016

Figure 5-14. Example: Using imports and exports

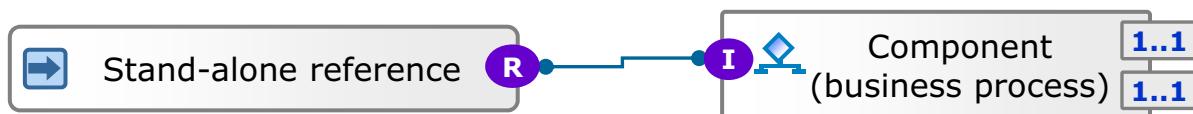
WB8601.2

Notes:

1. The stand-alone reference is a non-SCA component, such as a JavaServer Page (JSP) that can interact with the SCA components. In this scenario, the stand-alone reference invokes the portfolio service by calling its interface.
2. The **PortfolioManager** module has a service component, **portfolio service**, which is implemented in Java. The portfolio service references another service component, **customer service**, in the same module. It also references an external service in the **StockQuoteManager** module through an import component.
3. The customer service is an SCA component that is implemented in Java and gets invoked by the portfolio service.
4. The import invokes the **StockQuote** service, which is in a separate module through the export.
5. The **StockQuote** service exposes itself through an export component.
6. The **PortfolioManager** library stores the shared artifacts. In this case, it stores the **StockQuote** WSDL that the export uses to call the **StockQuote** service interface.
7. Finally, the **StockQuote** service returns the value of the **StockQuote** request.

SCA module components: Stand-alone references

- A non-SCA component can use a stand-alone reference to invoke an SCA component
 - You can use *Stand-alone references* in a Java program, for example, a JavaServer Page (JSP) to invoke SCA components or imports
- Stand-alone references do not have any implementation or interface
- A module can contain only one stand-alone reference artifact



© Copyright IBM Corporation 2016

Figure 5-15. SCA module components: Stand-alone references

WB8601.2

Notes:

Stand-alone references are references to applications that are not defined as SCA components (for example, JavaServer Pages or servlets). With stand-alone references, these applications interact with SCA components. Stand-alone references can be used in a mediation module. Stand-alone references do not have either an interface or an implementation. A module can contain only one stand-alone reference artifact. With stand-alone references, you can use your existing applications together with Service Component Architecture components that are created in IBM Integration Designer.

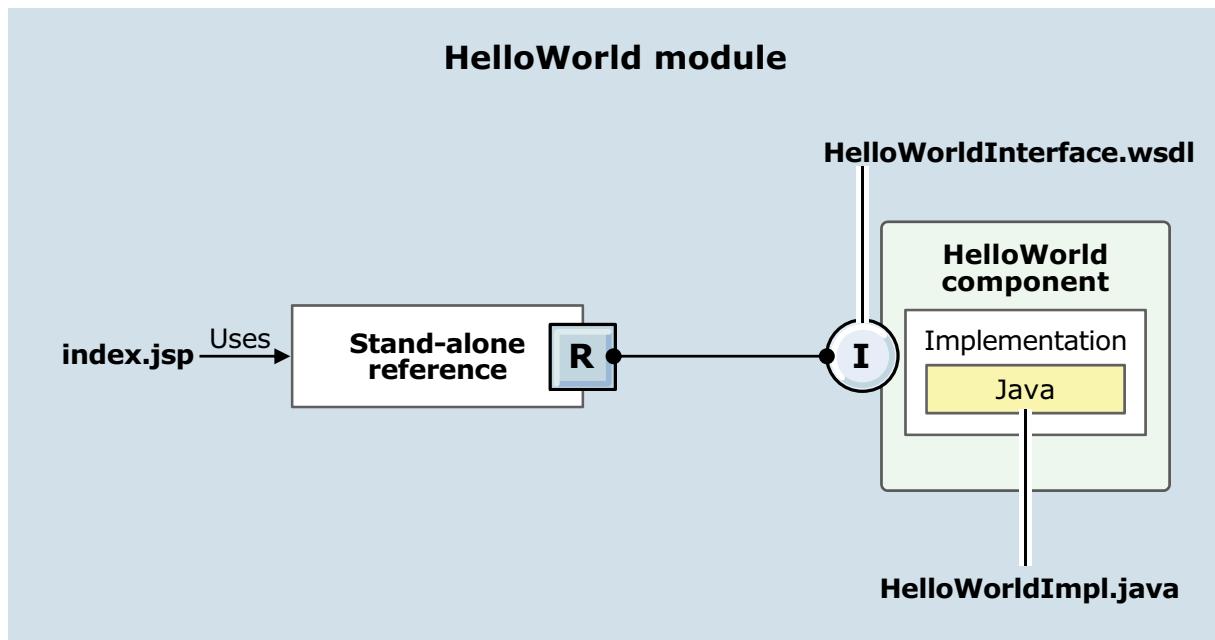
JavaServer Pages (JSP) files invoke SCA components by using stand-alone references in the module assembly. The JSP file is deployed in the same EAR file as the module assembly. Do not copy the `sca.references` file to another EAR file. The file does not work outside the module that it belongs to. If the web project is in another EAR file, then the JSP file is in another module, and imports and exports are required for it to work with services in another module.

The steps for adding the JSP file to the module are:

1. Create a web project.

2. Build the JSP file in the web project. If you are importing the code, import it into this web project. See the example of code that is provided.
3. Open the module (which has the stand-alone references) with the dependency editor. Add a dependency on the web project as a Java EE project. Make sure that the **Deploy with Module** check box option is selected so that the web project is added to the EAR file.

Example: HelloWorld



© Copyright IBM Corporation 2016

Figure 5-16. Example: HelloWorld

WB8601.2

Notes:

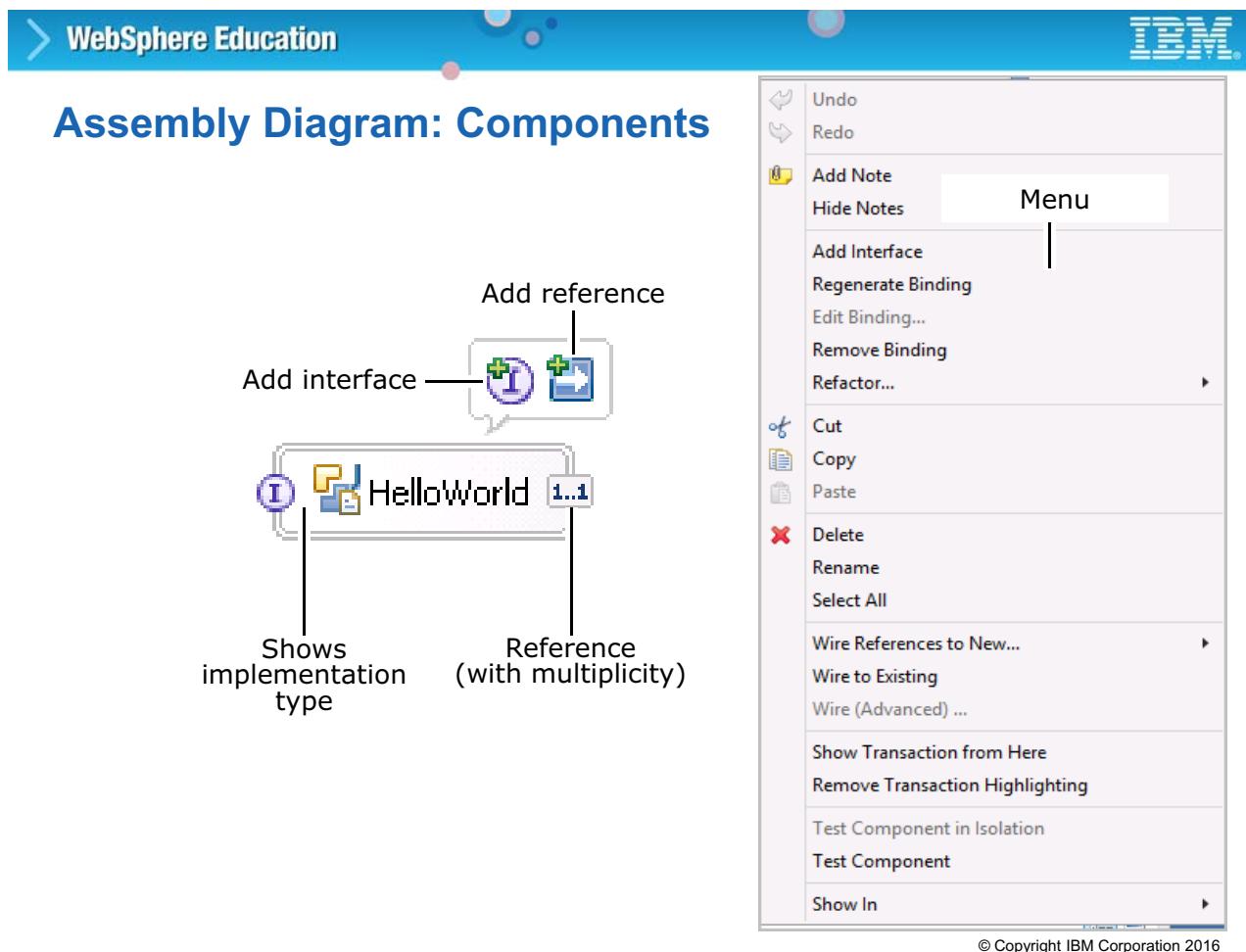


Figure 5-17. Assembly Diagram: Components

WB8601.2

Notes:

Composing SCA modules: Assembly Diagram editor

- You assemble SCA modules by first adding SCA components to the assembly diagram
- You then create visual relationships between the components in the module, by connecting them with wires

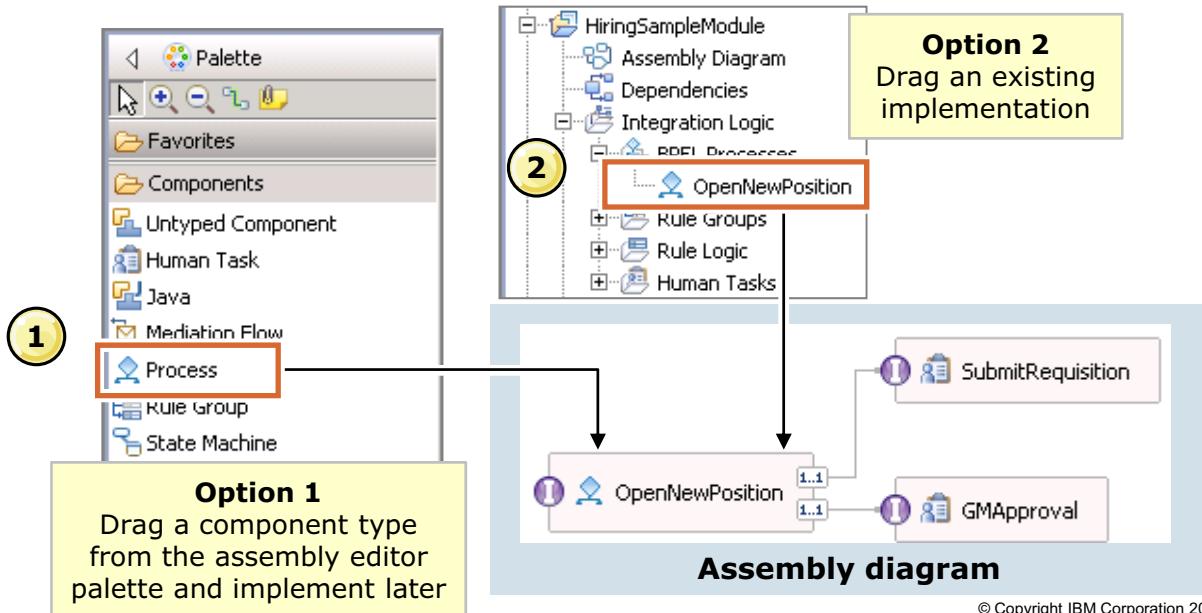


Figure 5-18. Composing SCA modules: Assembly Diagram editor

WB8601.2

Notes:

SCA components can be added to the assembly diagram in one of two ways:

1. Drag a component type from the assembly editor palette onto the diagram and generate the implementation later (top-down)
2. Drag an existing implementation (such as a BPEL process) onto the assembly diagram of the module (bottom-up)

The assembly editor palette is divided into several sections, each containing SCA components that can be added to the assembly diagram.

A “meet-in-the-middle” approach would be to drag an empty component onto the assembly diagram, and assign an existing implementation to it.

The **Components** section contains SCA components with implementations that provide business logic. The component with no implementation type can be used when you want to create a component but do not know which type of implementation to use. Later, you can specify the implementation type and generate its implementation. The Java component in the palette is implemented as a Java class. It supports both WSDL type interfaces and Java type interfaces. It is the only component type that supports Java type interfaces. Other components include process,

human task, state machine, rule group, import, export, stand-alone reference, interface map, and selector. Interface maps and selectors are special components that do not have business logic implementations. Interface maps contain mapping logic for interface operations. Selectors provide routing logic to invoke services.

By using the **Favorites** section, you gather frequently used icons into a favorites folder. To put an icon in this folder, right-click the icon and click **Add to favorites**. (If the favorites folder does not exist, it is created with the first icon added to it.) To hide the text labels when you are familiar with the icons, right-click the palette and toggle the “show tool names” setting. To increase the size of the icons, right-click the palette and toggle the “use large icons” setting.

The **Outbound adapters** and **Inbound adapters** sections are used to create an import or export component that uses an adapter. When you drag an adapter item from the assembly editor palette, you are asked to complete a wizard to generate the import or export.

An export is a published interface from a component or import that offers its service to the outside world, for example, as a web service. Exports have interfaces that are the same as, or a subset of, the interfaces of the component or import that they are associated with so that the published service can be called. An export that is dragged from another module into an assembly diagram automatically creates an import. Exports that are shown under the assembly diagram in the Business Integration view can also be used to create imports in other modules.

5.2. Testing SCA applications

Testing SCA applications



© Copyright IBM Corporation 2016

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

10.1

Figure 5-19. Testing SCA applications

WB8601.2

Notes:

Testing SCA components in the integration test client

- Using the integration test client in IBM Integration Designer, you can test:
 - An individual module
 - A set of interacting modules
 - An individual component
 - A set of interacting components
 - A test suite
- When testing components, tests are done on interface operations
 - You can determine whether the components are correctly implemented and the references are correctly wired
- In component tests, unimplemented components or unwired references can be emulated
 - Modules do not need to be complete before testing
 - Emulation can be programmatic

© Copyright IBM Corporation 2016

Figure 5-20. Testing SCA components in the integration test client

WB8601.2

Notes:

The integration test client is offered for unit testing modules and components, and provides visual tools that allow developers to start, view, and interact with resources that are being tested. The test client takes care of deploying the necessary modules at run time, and uses emulators to allow testing before all components in the application are fully implemented.

After developing a potential IT solution in the IBM Integration Designer environment, you want to test it. IBM Integration Designer comes with instances of IBM Process Server and IBM Process Center that can be controlled and managed from within the IBM Integration Designer environment. This feature allows the IBM Integration Designer user to test construction, without maintaining a full production environment copy of IBM Process Server or IBM Process Center, with all its associated setup and management. Tight integration provides IBM Integration Designer the opportunity to hook directly into the debugging capabilities, thus allowing the developer to start debugging tools that can set breakpoints to step through the solution. Integration also provides IBM Integration Designer the ability to connect to a repository on IBM Process Center to publish or deploy repository assets.

Like other IBM integrated development environments (such as IBM Rational Application Developer), IBM Integration Designer provides an integrated unit test environment for deploying

and testing your business applications during the development process. As part of the installation process of IBM Integration Designer, you can silently install the IBM Process Server unit test environment. However, if IBM Integration Designer must use a stand-alone IBM Process Server test server that you already configured, you can configure the tools to use that test environment during your development process.



Test configurations

- Test configurations control your tests
- A test configuration specifies one or more modules to test, each of which might include:
 - Zero or more emulators for components or references in the module
 - Zero or more monitors for the wires in the module
- When you open the integration test client, a default test configuration is automatically created that you can immediately use for testing
- The default test configuration is often all that you need for testing your modules and components
 - You can choose to edit and customize the default test configuration, or you can create and edit one or more new test configurations
 - Customized test configurations can be saved and reused

© Copyright IBM Corporation 2016

Figure 5-21. Test configurations

WB8601.2

Notes:

Emulators

- By using the integration test client, you can use emulators to emulate components and references in your modules
- During a test, when control flows to an emulated component or reference, the integration test client intercepts the invocation and routes it to the associated emulator
- Two types of emulators:
 - **Manual:** Test pauses for you to manually specify output parameter values
 - **Programmatic:** Uses a Java snippet to provide response values automatically

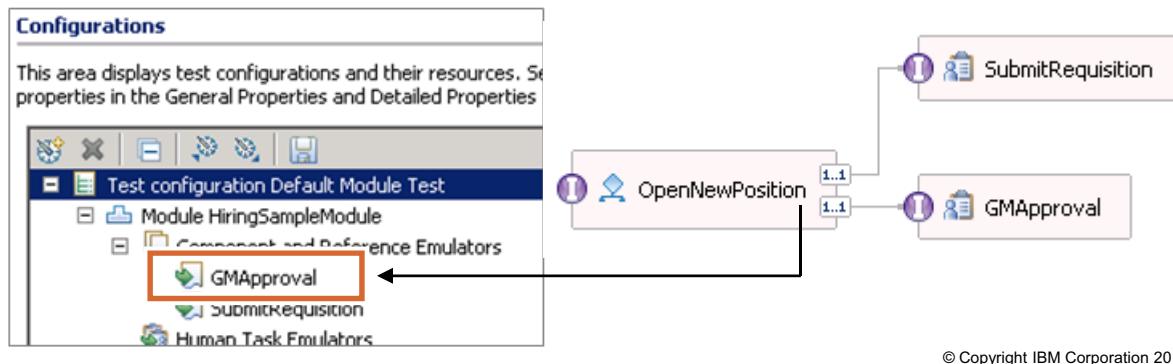


Figure 5-22. Emulators

WB8601.2

Notes:

A manual emulator is an emulator for which you must specify response values for an emulated component or reference at run time. When a manual emulator is encountered during a test, a manual “emulate” event is generated. By using test pauses, you manually specify some output parameter values or throw an exception for the emulated components or references.

A programmatic emulator provides response values for an emulated component automatically by using a Java program that is contained in a visual snippet or Java snippet. When a programmatic emulator is encountered during a test, a programmatic “emulate” event is generated, and the exception or the output parameter values are automatically provided.

Monitors (1 of 2)

- When you use the default test configuration or when you add a test configuration, monitors are automatically added for any component wires and exports in the module
- When you invoke an operation and run a test, monitors listen for requests and responses that flow over the wires and exports
 - If a request is detected, a **request** event is generated
 - If a response is detected, a **response** event is generated

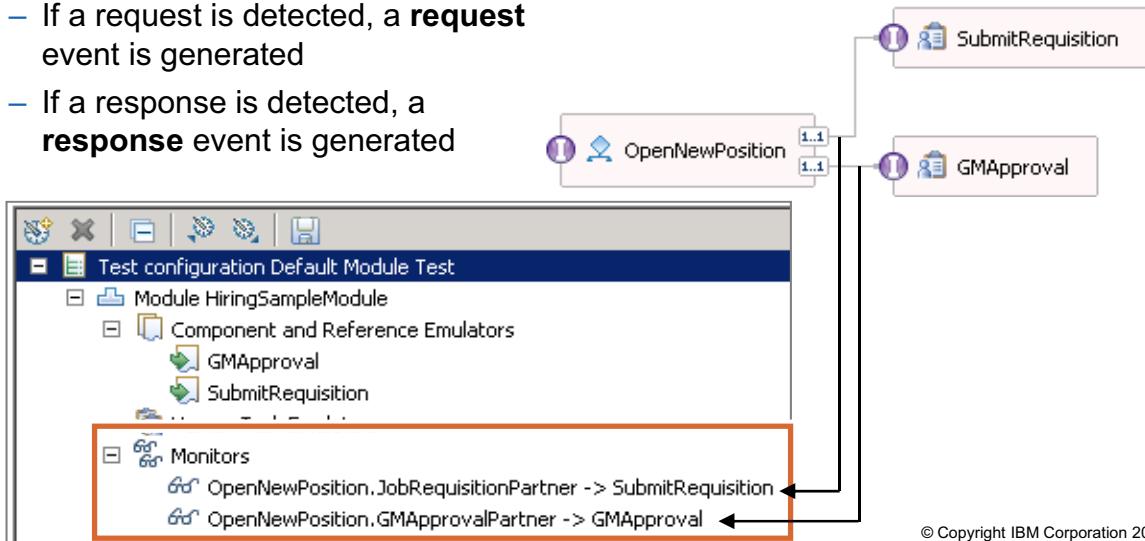


Figure 5-23. Monitors (1 of 2)

WB8601.2

Notes:

When you run a test by invoking an operation or emitting an event in the integration test client, several different types of events are generated over the course of the test. These events are either interactive or informational. Interactive events require you to manually specify values before the test can continue. By comparison, informational events are purely informative and they do not require any action.

By default, the integration test client always generates certain types of events, such as “return” events. However, you can customize test configurations to control whether other types of events are generated, such as monitor “request” and “response” events.

The event types that generated by the integration test include:

- Invoke event (manual):** This event is an interactive event. The event is generated when you start the integration test client or when you click the “invoke” icon in the integration test client. By using this event, you can select a test configuration, module, interface, and operation. You specify input parameter values to pass to the operation and invoke the operation.
- Invoke started event:** This event is an informational event. The event indicates that a test started.

- **Invoke event:** This event is an informational event. The event confirms when an operation is successfully invoked.
- **Request event:** This event is an informational event. The event indicates that a request is passed across a monitored wire. The event contains return values or exceptions.
- **Response event:** This event is an informational event. The event indicates that a response is passed across a monitored wire. The event contains return values.
- **Response event (exception):** This event is an informational event. The event indicates that a response is passed across a monitored wire. The event contains an exception. If the exception contains user-specified WSDL fault data that does not include a string type object, the test client displays the fault data. It also displays the module, component, interface, and operation that contributed to the exception. However, if the fault data includes a string type object, the test client displays the associated exception class, exception message, and trace. Similarly, if the exception is a Java exception or it contains a Java exception, the test client displays the exception class, exception message, and trace.
- **Emulate event (manual):** This event is an interactive event. The event indicates that a component or reference was encountered that a manual emulator is emulating. When a manual emulator is encountered during a test, the invocation of the operation pauses so that you can specify some output parameter values or select an exception to throw. The list of exceptions is populated based on either the throws clause of the J-type interface or the faults section of the W-type interface.
- **Emulate event (programmatic):** This event is an informational event. The event indicates that a component or reference was encountered that is being emulated with a programmatic emulator. Programmatic emulators use scripts to automatically pass output parameter values or throw exceptions, which means that you are not required to specify values or throw exceptions yourself.
- **Return event:** This event is an informational event. The event indicates that a response is received from the invocation of an operation.
- **Exception event:** This event is an informational event. The event indicates that an exception occurred. The exception can be an unmonitored exception that occurred during the invocation of an operation, or it can be an exception that you deliberately selected and threw. If the exception contains user-specified WSDL fault data that does not include a string type object, the test client displays the fault data. It also displays the module, component, interface, and operation that contributed to the exception. However, if the fault data includes a string type object, the test client displays the associated exception class, exception message, and trace. Similarly, if the exception is a Java exception or it contains a Java exception, the test client displays the exception class, exception message, and trace.
- **Invoke returned event:** This event is an informational event. The event indicates that an operation automatically completed or was manually stopped. However, when an invoke returns an event, it does not necessarily mean that all activities are complete as the result of an invocation. It means that the original invocation was returned. If the original invocation was one-way or if some resultant invocations were asynchronous, events can still be displayed within the **Invoke started** or **Invoke returned** events.
- **Attach event:** This event is an interactive event. The event is generated by right-clicking a module in the Business Integration view and clicking **Test > Attach** or by clicking the “attach”

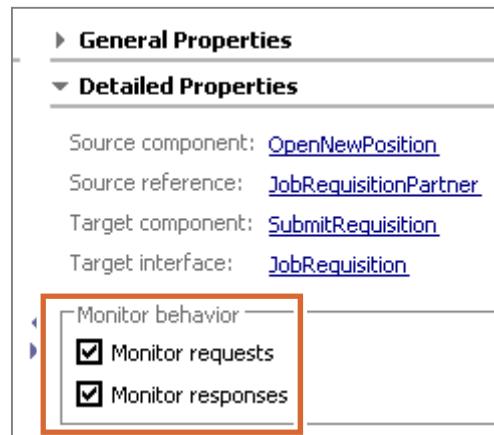
icon in the integration test client. With this event, you can attach the integration test client directly to a test configuration module. You can then use a JMS message, web service, JSP, or some other mechanism to invoke an operation rather than use the invocation mechanism that the test client provides.

- **Emit event:** This event is an interactive event. The event is generated when you start the integration test client by clicking the **Test > Test event definition** menu item or when you click the “emit” icon in the integration test client. With this event, you can select an event definition for testing. You can also specify values for the event definition and then emit an event that the event definition defines.
- **Emitted event:** This event is an informational event. The event indicates when an event is emitted for a selected event definition.
- **Succeeded event:** This event is an informational event. The event indicates that an emitted event is correctly emitted for the selected event definition.



Monitors (2 of 2)

- Events show parameter data that flows across the wires
 - Added to the events that are displayed in the Events area of the integration test client
- You can edit the monitors and change whether they monitor requests, responses, or both
- You can also remove the monitors or add more monitors as required



© Copyright IBM Corporation 2016

Figure 5-24. Monitors (2 of 2)

WB8601.2

Notes:



Entering data for testing

- In production systems, large amounts of data move across the components and server
- Testing involves not just making sure that the flow is correct but also that the right data is moving across
- Different ways of entering data for testing
 - Use test data table to specify expected input data and output response data for the components that you want to emulate
 - Using data pools, you can create, edit, and reuse the data pool

© Copyright IBM Corporation 2016

Figure 5-25. Entering data for testing

WB8601.2

Notes:

Using a test data table for testing

- Import Java extension classes or packages and specify, view, and edit values for operations.

Name	Type
--> Invoke CreditRiskAssessment:InputCriterion(Input)	
Input	CustomerApplication
accountNumber	string
applicationDate	string
applicationDecision	boolean
comments	string
companyName	string
contactFirstName	string
contactLastName	string
contactPhoneNumber	string
creditRating	string
creditReportNeeded	boolean

© Copyright IBM Corporation 2016

Figure 5-26. Using a test data table for testing

WB8601.2

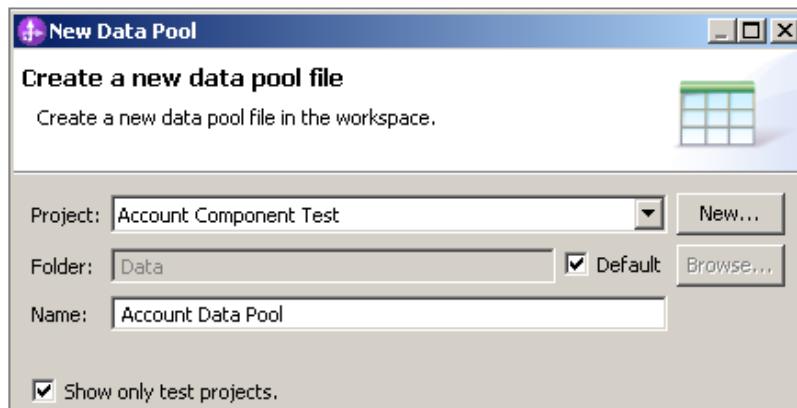
Notes:

The test data table view has two panes. These panes are described in the following sections:

- Imports: The Imports pane is used to import Java extension classes or packages. It provides code assist in the test data table.
- Defaults: The Default pane contains the test data table that is used to specify, view, and edit values for operations. The values in the test data table can be repeated in variations. A variation contains a unique set of input values and expected results, and you can run a test case against them. When a test case is run, it is run against all of the variations of the test data table.

Using a stand-alone data pool for testing

- The integration test client has a stand-alone data pool wizard
- The data pool:
 - Is saved in your workspace
 - Can be exported with a project
 - Can be checked into source control
- The data pool editor:
 - Can open multiple data pools
 - Copy values between pools



© Copyright IBM Corporation 2016

Figure 5-27. Using a stand-alone data pool for testing

WB8601.2

Notes:

On the events page of the integration test client, a value editor is provided to specify, view, edit, and pass values for operations, manual emulations, and event definitions. By using the value editor, you can save values to a data pool, where you can use the data pool editor to view and edit the values, and later reuse them in the value editor.

In the integration test client, you can use one or more data pools. You can save values to the data pools, edit values in the data pool editor, and reuse values from the data pools. You can manage the input values for your operations and output values for your manual emulations. When you create a data pool, it is added to a test project.



Overview: Testing SCA components (1 of 2)

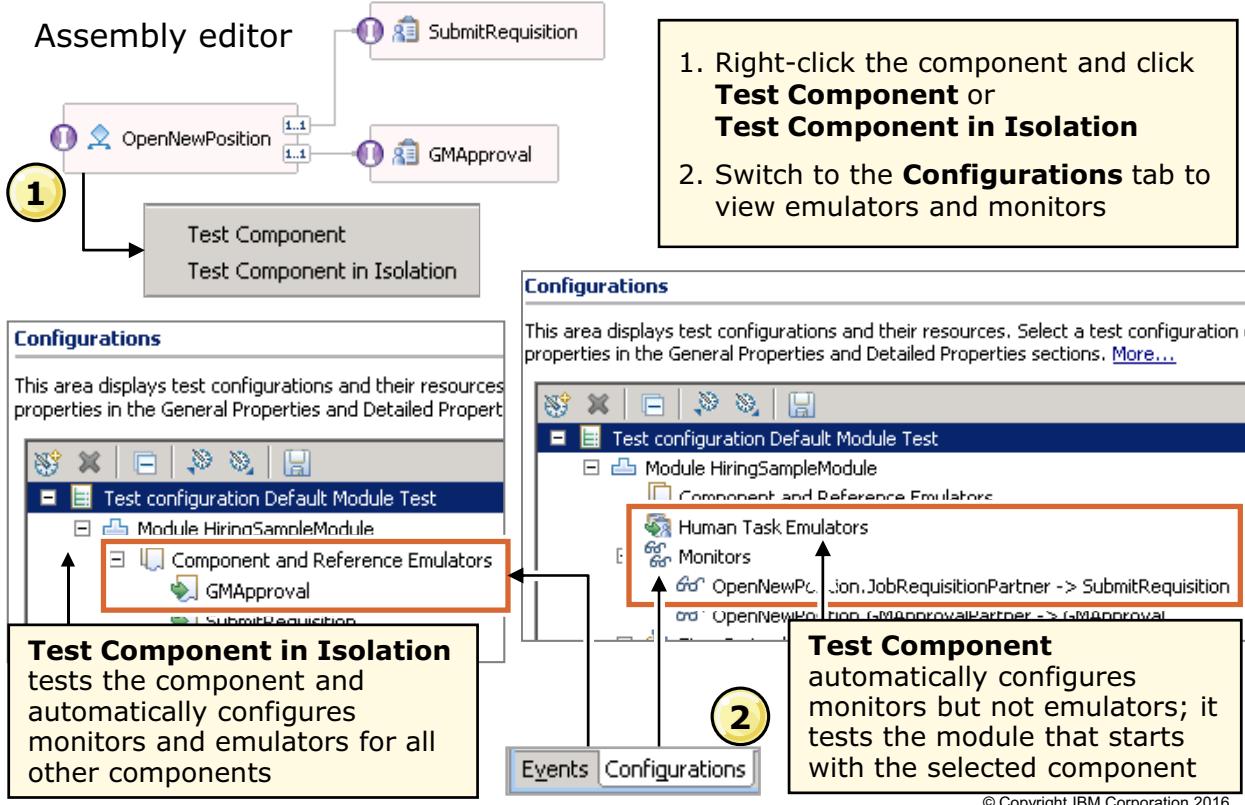


Figure 5-28. Overview: Testing SCA components (1 of 2)

WB8601.2

Notes:

The test client interface has two main features:

- Events page
- Configurations page

Almost all of the tasks that you can do in the integration test client are either initiated or done in one of the two pages.

In the configurations page of the integration test client, you can edit the default test configuration, or you can create and edit new test configurations. In this way, you can add modules to your test configurations, or add emulators and monitors to your test configuration modules, to more precisely control your tests.

The configurations page of the integration test client is divided into four sections.

- The **Configurations area** provides a navigation tree of your default test configuration and displays the following information:
 - The name of the test configuration
 - The names of the modules in the test configuration

- The names of any emulators in each test configuration module
- The names of any monitors in each test configuration module

In the configurations area, emulators for references have the following naming convention:

ComponentName.referenceName

By comparison, emulators for components have the following naming convention:

ComponentName

Monitors have the following naming convention: SourceComponentName.referenceName - ComponentName

2. The **Control area** provides the following icons to help you manage your default test configuration:
 - **Add:** Opens the configuration wizard so that you can add one of the following test configuration elements to the configurations page:
 - Module
 - Emulator
 - Monitor
 - **Remove:** Removes selected test configuration elements from the configurations page
3. The **General Properties** area provides the name and description (if any) of the default test configuration, module, emulator, or monitor that is selected in the configurations area.
4. The **Detailed Properties** area displays the specific properties of any test configuration element that is selected in the configurations area. For example, if the default test configuration is selected in the configurations area, the detailed properties area displays the name of any modules that are contained in the test configuration.

WebSphere Education

Overview: Testing SCA components (2 of 2)

Displays detailed properties for events such as an initial request and return parameters

General Properties

Detailed Properties

Module: [HiringSampleModule](#)
 Source component: [OpenNewPosition](#)
 Source reference: [JobRequisitionPartner](#)
 Target component: [SubmitRequisition](#)
 Target interface: [JobRequisition](#)
 Target operation: [createRequisition](#)

Request parameters:

[Value Editor](#) [XML Source](#)

Events

This area displays the events in a test trace. Select an event to display its properties in the Detailed Properties sections. [More...](#)

Execution trace (events)

© Copyright IBM Corporation 2016

Figure 5-29. Overview: Testing SCA components (2 of 2)

WB8601.2

Notes:



Testing SCA modules in the integration test client

- Entire modules and groups of interacting modules are tested by using a configured server runtime
 - Installation of the runtime environment is optional
 - The workspace preferences display installed runtimes
 - Runtimes can be local or remote
- By default, testing a module or group of modules automatically builds and packages the projects, publishes them to the server, starts the server, and starts the applications
 - You can disable automatic publishing in the preferences

© Copyright IBM Corporation 2016

Figure 5-30. Testing SCA modules in the integration test client

WB8601.2

Notes:

To disable automatic publishing:

1. Click **Window > Preferences**.
2. Expand **Business integration > Test** and select **Integration test client**.
3. Clear the **Publish automatically before starting test client** check box.

Overview: Testing SCA modules (1 of 2)

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter area to run the test. [More...](#)

Configuration:	Default Module Test
Module:	HiringSampleModule
Component:	OpenNewPosition
Interface:	HiringPosition
Operation:	createPosition

Initial request parameters:

Value editor XML editor

Name	Type
inputReqn	Requisition
reqNum	string
requestor	string
status	string
approvalNeeded	boolean

- Right-click the module and click **Test > Test module**

- The module is automatically published, and the default module test is loaded in the integration test client

© Copyright IBM Corporation 2016

Figure 5-31. Overview: Testing SCA modules (1 of 2)

WB8601.2

Notes:

IBM Integration Designer includes a number of options for data entry in the input editor. Note the following options:

- You can use the “maximize” icon to view the input editor in full screen.
- You can use the arrow keys to browse through the input fields.
- The input editor supports multi-line data entry, and the input dialog box is resizable.
- Icons indicate the input structure type and whether the input value is unset (the unset icon resembles a black “X”).
- Incorrect values are flagged in the errors column, and icons are to browse between errors. Input errors do not stop the test from proceeding; they are informational only.
- The menu (revealed by right-clicking an item in the input editor) contains several options:
 - Use previous value:** You can choose from a list of recently used values for a parameter.
 - Import from or Export to XML file:** You can import values from an XML file or export input data to an XML file.

- The **Copy value** and **Paste value** options are integrated with the system clipboard.

You can also create customized key bindings for the options in the test client. For example, you can create a key sequence to run a test. This feature is accessible by choosing **Window > Preferences > General > Keys**.



Overview: Testing SCA modules (2 of 2)

- Right-click the server, click **Add and Remove**, and add the modules to test
 - The projects are published whether the server is started or stopped
- Start the server and test the modules by using the integration test client, or test the modules “end-to-end” as they would display in production
 - SOAP is the default server communication protocol

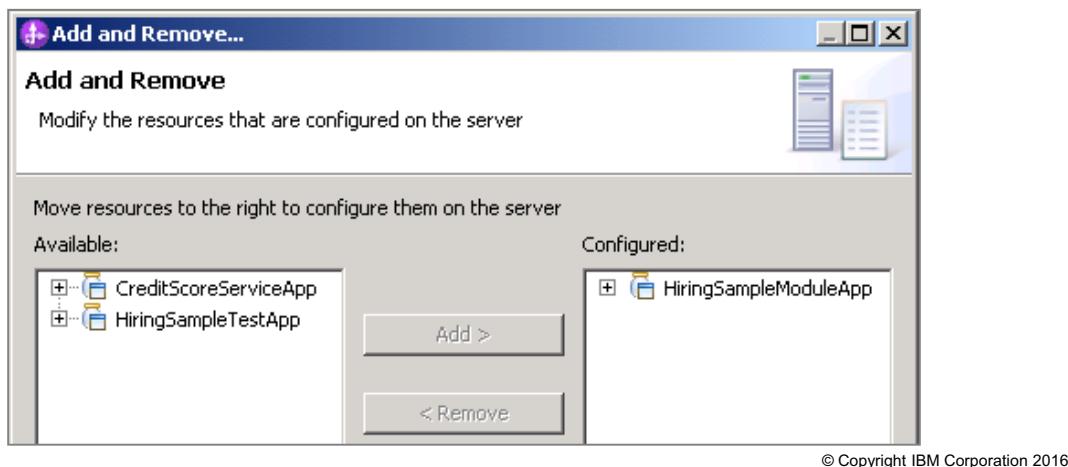


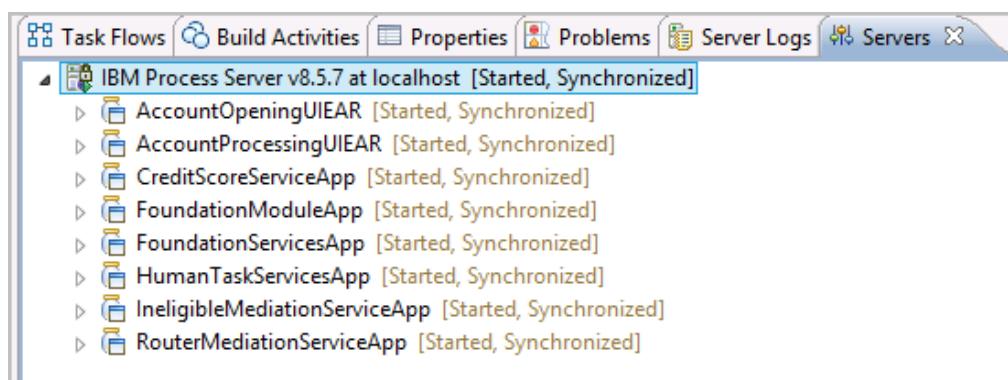
Figure 5-32. Overview: Testing SCA modules (2 of 2)

WB8601.2

Notes:

Using the Servers view when testing modules

- After deployment, the Servers view displays information about the state of the server and the projects that are published to it
- Projects and their state are listed under the server in a navigation tree
 - Projects are synchronized or must be republished
 - Right-click modules, and click **Remove** to remove them individually
- Server state is also visible:
 - Server is stopped, started, or in debug mode



© Copyright IBM Corporation 2016

Figure 5-33. Using the Servers view when testing modules

WB8601.2

Notes:

WebSphere Education

Server Logs view

- Server Logs view is used to display the server console and SystemOut.log messages
 - Server Console view also available
 - Filter messages by type and load invocation records
 - Refresh every 5 seconds (can be changed)

Server Logs

The Server Logs view is used to display the contents of the server console and server log files. It automatically displays console output for each server that is started, but you can also manually load and display the contents of the console and log files for any server. The Server Logs view provides several advantages over the traditional Console view, such as the ability to filter records and display invocation records in hierarchical format. If you want to open the Console view, select **Window > Show View > Console**.

Getting started with the Server Logs view

To load server console or log records into the Server Logs view, click the **Load Server Console or Log** icon

To filter records in the Server Logs view, click the **Select Records to Display** icon

To enable or disable cross-component tracing, click the **View Menu** icon

To load invocation records into the integration test client, click the **Load into Test Client** icon

Load SystemOut.log from a directory or another computer

- Able to load multiple logs
- Message color:
 - Green: Message
 - Purple: Warning
 - Red: Error

© Copyright IBM Corporation 2016

Figure 5-34. Server Logs view

WB8601.2

Notes:

The Server Logs view replaces the Console view and is used to view the server console output and messages that are written to the JVM log (SystemOut.log) of the server. This view is easier to read, filter, and search than the Console view, though you can still access the Console view by clicking **Window > Show view > Console**.

You can do several things in the Server Logs view that you cannot do in the console view. First, you can load multiple logs and traces into the Server Logs view to get a bigger picture of an error. In addition, you can open the assembly diagram from log content. The most important difference between the Console view and the Server Logs view is the ability to filter messages by type. For more information about the filtering options available for the Server Logs view, see the product documentation.

Cross-component tracing

- Cross-component tracing (XCT) allows developers to examine flows in a business process step-by-step
- Maps SystemOut.log records and trace.log records back to SCA programming model
- Enabling XCT with data snapshot captures input and output data in the log files

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

Type	Thread ID	Contents
Invocation sequence (OpenNewPosition:createPosition)	000000F5	
Start invoke (OpenNewPosition:createPosition)	000000F5	Start of the one-way invocation of operation OpenNewP
Start component (OpenNewPosition:createPosition)	00000128	Start of the component processing of operation OpenNe
In BPEL process	00000128	07e8a3dd-ddab-4640-be64-54345fc78d94 STATE b740:
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL process OpenNewPosition:
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL process OpenNewPosition:_F
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL process OpenNewPosition:_
Log message	00000126	>>> Find Job Candidates --- No Approval Needed
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL process OpenNewPosition:_
End component (OpenNewPosition:createPosition)	00000128	End of the component processing of operation OpenNe
End invoke (OpenNewPosition:createPosition)	000000F5	End of the one-way invocation of operation OpenNewP

© Copyright IBM Corporation 2016

Figure 5-35. Cross-component tracing

WB8601.2

Notes:

Cross-component trace (XCT) maps SystemOut.log and trace.log records back to the SCA programming model. It supports all SCA call patterns, including asynchronous one-way, asynchronous with callback, asynchronous with deferred response, or synchronous calls. It works with business objects, or simple data types, and supports the network deployment (ND) environment. In a network deployment environment, cross-component tracing can load files from each server, or can load only some of the files from all the servers. Using cross-component tracing, developers can create test client traces from log record data.

Cross-component tracing supports Business Process Execution Language (BPEL) microflows and long-running business processes. It follows the process flows and correlates log records to long-running processes, with the capability to support multiple “pick” and “receive” activities. Moreover, it also supports HTTP, JMS, WebSphere MQ, and MQ/JMS bindings.

The three settings available for cross-component tracing are:

- **Disabled:** No records are generated in the server console and logs.

- **Enabled:** Invocation records are generated into both the server console and the SystemOut.log and trace.log files. While cross-component tracing is enabled, recorded properties do not include any invocation input or output data.
- **Enabled with data snapshot:** Invocation records are generated into both the server console and the SystemOut.log and trace.log files. With this option, recorded properties include invocation input and output data, which is captured in input and output files under the logs\xct directory.



Unit summary

- Identify the purpose and business value of using SCA
- Identify and explain the function of the individual parts of an SCA component
- Define the concept of a service module and explain each of the service module components
- Describe how to test SCA applications in IBM Integration Designer
- Describe the features that are available in the IBM Integration Designer test environment

© Copyright IBM Corporation 2016

Figure 5-36. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. What is the purpose of an export?
2. What is the purpose of an import?
3. True or False: An SCA component can specify more than one interface.
4. What is the function of a reference in an SCA component?
5. How many references can be specified for an SCA component?
6. Define the purpose of emulators and monitors in the IBM Integration Designer integration test client.

© Copyright IBM Corporation 2016

Figure 5-37. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Checkpoint answers

1. The purpose of an export is to expose the function of your service to callers outside the module.
2. The purpose of an import is to call services that exist outside your module.
3. True.
4. A reference is used to specify how to invoke another SCA component or another service by using an import.
5. 0..N.
6. By using the integration test client, you can use emulators to emulate components and references in your modules. During a test, when control flows to an emulated component or reference, the integration test client intercepts the invocation and routes it to the associated emulator. When you invoke an operation and run a test, monitors listen for any requests and responses that flow over the wires or exports and produce events. These events show parameter data that flows across the wires.

© Copyright IBM Corporation 2016

Figure 5-38. Checkpoint answers

WB8601.2

Notes:

Exercise 3



Exploring IBM Integration Designer, part II

© Copyright IBM Corporation 2016

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

10.1

Figure 5-39. Exercise 3

WB8601.2

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Assemble an SCA application
- Examine project components
- Explore staging projects
- Use the IBM Integration Designer test environment
- Enable cross-component tracing to examine a business process in testing
- Deploy a library globally

© Copyright IBM Corporation 2016

Figure 5-40. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 3

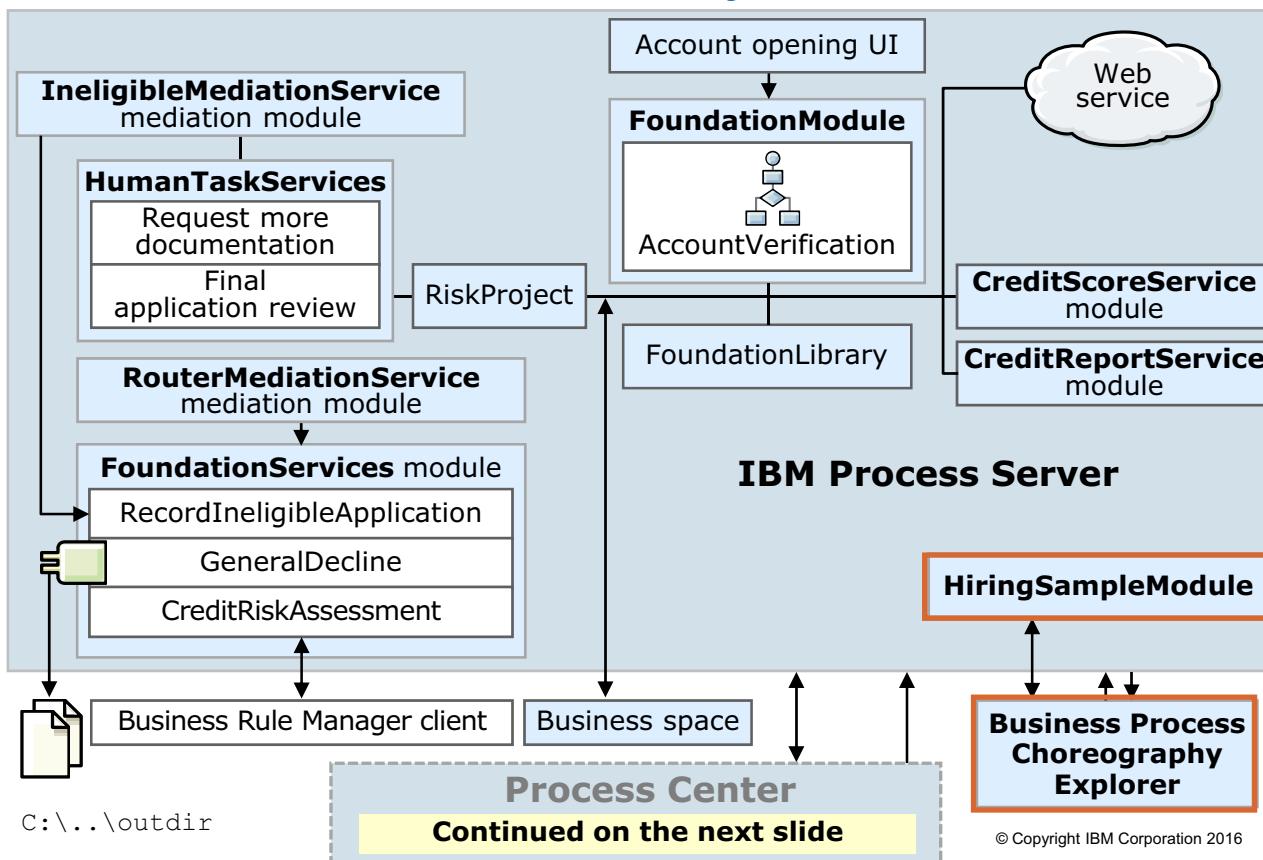


Figure 5-41. Course business scenario: What you work with in Exercise 3

WB8601.2

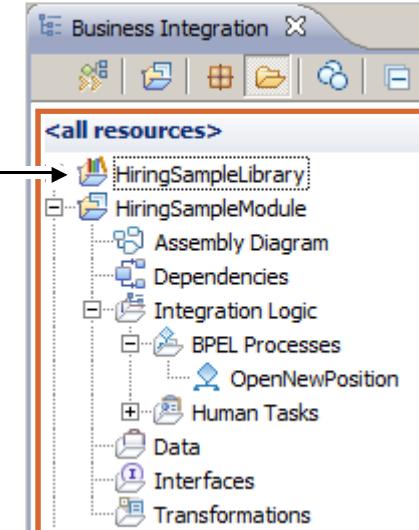
Notes:

Components that are required for Exercise 3

Prebuilt components that are imported in the lab:

- 1. HiringSampleLibrary**
- 2. HiringSampleModule**

Previously imported in Exercise 2



The screenshot shows the 'Business Integration' application window. In the center, there's a tree view under the heading '<all resources>'. The tree structure is as follows:

- HiringSampleLibrary
 - HiringSampleModule
 - Assembly Diagram
 - Dependencies
 - Integration Logic
 - BPEL Processes
 - OpenNewPosition
 - Human Tasks
 - Human Tasks
 - Data
 - Interfaces
 - Transformations

New components that you create in the lab:

None

© Copyright IBM Corporation 2016

Figure 5-42. Components that are required for Exercise 3

WB8601.2

Notes:



Use cross-component tracing in Exercise 3

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

Type	Thread ID	Contents
Invocation sequence (OpenNewPosition:createPosition)	000000f5	
Start invoke (OpenNewPosition:createPosition)	000000f5	Start of the one-way invocation of
Start component (OpenNewPosition:createPosition)	00000128	Start of the component processing
In BPEL process	00000128	07e8a3dd-ddab-4640-be64-5434!
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL proce
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL proces
Start BPEL process (OpenNewPosition)	00000126	Start of processing for BPEL proce
Log message	00000126	>>> Find Job Candidates --- No A
End BPEL process (OpenNewPosition)	00000126	End of processing for BPEL proces
End component (OpenNewPosition:createPosition)	00000128	End of the component processing
End invoke (OpenNewPosition:createPosition)	000000f5	End of the one-way invocation of

© Copyright IBM Corporation 2016

Figure 5-43. Use cross-component tracing in Exercise 3

WB8601.2

Notes:

Welcome to Business Process Choreographer Explorer.

The page you requested is only available to registered users. Enter your username and password and click Login.

User Name: _____

Password: _____

Login

My To-dos

Use this page to work on tasks that are assigned to you. [?]

Work on Release Transfer Start Change Bus

	Priority ◊	Task Name ◊	State ◊	Kind ◊	Owner
<input type="checkbox"/>	5	GMAApproval	Ready	To-do Task	
<input checked="" type="checkbox"/>	5	SubmitRequisition	Ready	To-do Task	

Items found: 1 Items selected: 1

Work on Release Transfer Start Change Business Category

	Priority ◊	Task Name ◊	State ◊	Kind ◊	Owner
<input type="checkbox"/>	5	GMAApproval	Ready	To-do Task	
<input type="checkbox"/>	5	SubmitRequisition	Ready	To-do Task	

Items found: 2 Items selected: 0

Page 1 of 1

© Copyright IBM Corporation 2016

Figure 5-44. Test by using the Business Process Choreographer Explorer

WB8601.2

Notes:

Business Process Model and Notation or BPMN is a standardized graphical notation for creating diagrams of business processes.

BPMN is used so that everyone who is involved can interpret and understand the model. Throughout development, many different parties are involved in modeling. Every stakeholder, from the least technical to the most technical, understands the model to provide valuable feedback and continuously improve the process.

BPMN also allows a way to compact your process definition. Many of the symbols represent ideas, so symbols allow for a more concise and smaller model than drawing a diagram without BPMN.

BPMN has many benefits, but most importantly, BPMN creates a standardized bridge for the gap between the business process design and process implementation. This single notation is agreed upon among multiple IBM BPM vendors for the benefit of the user community.

IBM implements and interprets these elements to have specific meanings and terminology in the IBM Process Designer product. For definitions of the BPMN specification, see the BPMN Specification document version 2.0 from the Object Management Group. You can learn more about BPMN at: <http://www.bpmn.org>

The element palette of IBM Process Designer is shown in the slide image. These items correspond to BPMN elements and are used in modeling a process.

A BPMN standards group that is called Object Management Group is an international, open membership, not-for-profit technology standards consortium. You can search the Internet for this group.



References

- OSOA: Service Component Architecture family of specifications:
 - <http://oasis-open.org/sca>
- developerWorks C Building SOA solutions with the Service Component Architecture:
 - Part 1:
http://www.ibm.com/developerworks/websphere/techjournal/0510_brent.html
 - Part 2:
http://www.ibm.com/developerworks/websphere/techjournal/0512_barclay.html
 - Part 3:
http://www.ibm.com/developerworks/websphere/techjournal/0602_barclay.html
- Team development and artifact management (for WebSphere Integration Developer V6.2):
 - <http://www.ibm.com/developerworks/webservices/library/ws-WIDdev/>
- Tuscany open source runtime implementation of SCA:
 - <http://tuscany.apache.org/>

© Copyright IBM Corporation 2016

Figure 5-45. References

WB8601.2

Notes:

Unit 6. SCA bindings

What this unit is about

This unit teaches you about SCA components and their bindings. You also learn how to use and implement web services in the SCA framework, specifically with IBM Integration Designer.

What you should be able to do

After completing this unit, you should be able to:

- List the various types of SCA import and export bindings
- Describe how SCA bindings facilitate integration with different types of applications
- Describe how web services are used in the Service Component Architecture framework

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

After completing this unit, you should be able to:

- List the various types of SCA import and export bindings
- Describe how SCA bindings facilitate integration with different types of applications
- Describe how web services are used in the Service Component Architecture framework

© Copyright IBM Corporation 2016

Figure 6-1. Unit objectives

WB8601.2

Notes:



Topics

- SCA bindings
- Web services

© Copyright IBM Corporation 2016

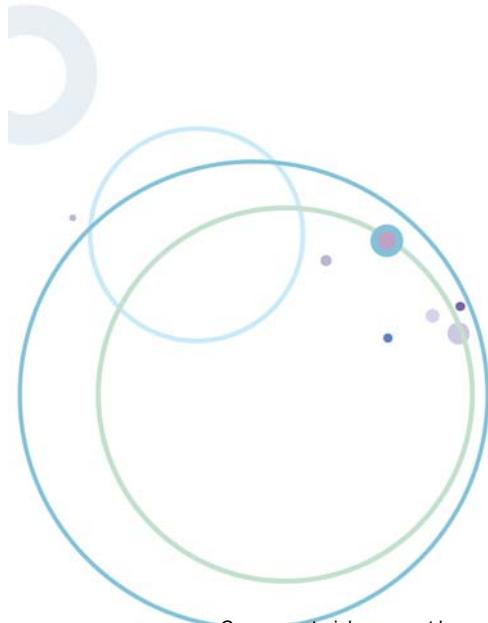
Figure 6-2. Topics

WB8601.2

Notes:

6.1. SCA bindings

SCA bindings



© Copyright IBM Corporation 2016

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

10.1

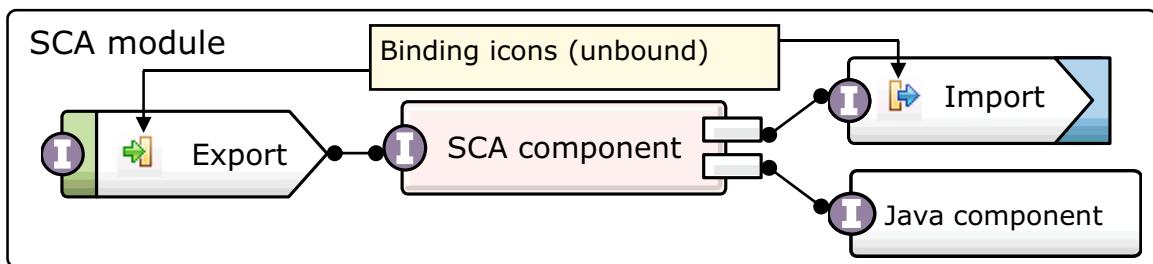
Figure 6-3. SCA bindings

WB8601.2

Notes:

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)



© Copyright IBM Corporation 2016

Figure 6-4. Incoming and outgoing interactions with external components

WB8601.2

Notes:

An export component is responsible for exposing a mechanism that a client uses 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 have something in common: they shield the heart of the module from having to know details of how to interact with external applications.

Every import has an associated interface type, which means that 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. This association means that 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.

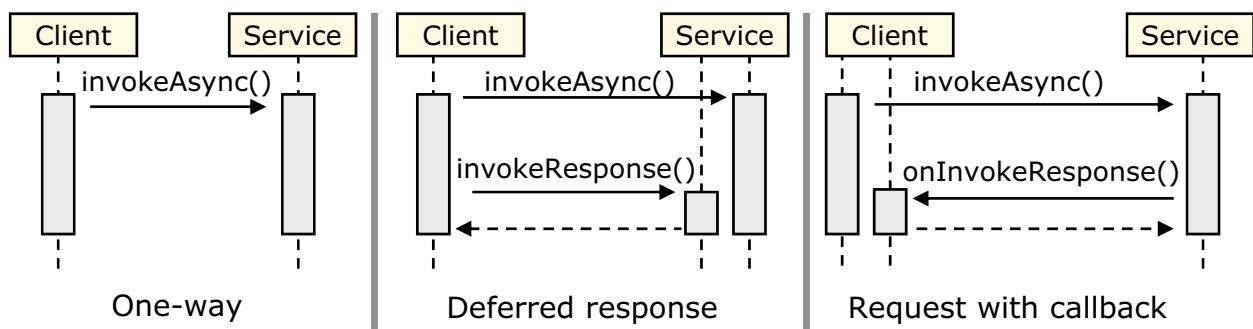
Imports and exports require binding information, which specifies the means of transporting data. Bindings for imports and exports have different purposes. An **import binding** describes the specific way that an external service is bound to an import component. An **export binding** describes how that 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.

The type of binding 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. By using an import with an SCA binding, you can access a service in another SCA module. By using a web service import binding, you can bind an external web service to an import. It is not necessary to define the import binding at 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 when the prototype evolves.

SCA invocation styles

- SCA components might call services synchronously or asynchronously, depending upon the preferred interaction style
 - The **synchronous** method is `invoke`
 - The **asynchronous** method is `invokeAsync`
- A synchronous call waits for the return value before proceeding
- Three kinds of asynchronous invocations:
 - **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



© Copyright IBM Corporation 2016

Figure 6-5. SCA invocation styles

WB8601.2

Notes:

The IBM Process Server runtime environment uses the service integration bus (SIBus) for asynchronous invocations, or Remote Method Invocation over Internet InterORB Protocol (RMI/IOP) for synchronous invocations. In asynchronous invocations, message consumers and producers do not communicate directly; they use a bus to send and receive messages. With all three types of asynchronous interaction styles, the client receives control back immediately from the SCA runtime environment upon an `invokeAsync()` call. However, three different ways can be used in which the client can capture the response later.

First, if the client chooses to discard the response entirely or if it is a call to a void method, the asynchronous invocation is said to be “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 do this request, the client must first implement the `ServiceCallback` interface. Then, after calling `invokeAsync()`, the SCA runtime environment provides a callback to the `ServiceCallback` handler to provide the response to the client.

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** are used from 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

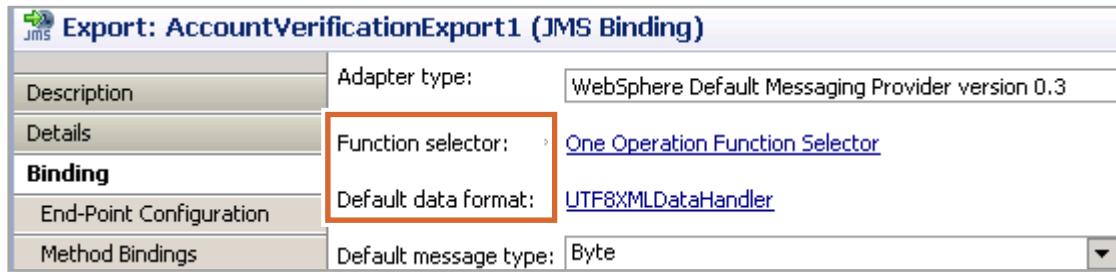


Figure 6-6. Import and export resources

WB8601.2

Notes:

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 message must be transformed into a business object. An instance of a `DataBinding` implementation is called, and the physical data 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, `DataBinding` again plays a role. `DataBinding` is called and supplied with an instance of a business object. The logic of the `DataBinding` instance must then serialize the content of the business object into a physical representation that is used as the body of the message. Although a `DataBinding` instance can easily be custom-written, IBM supplies starter or generic `DataBinding` implementations that are sufficient in most cases.

Quality of service qualifiers

- Qualifiers are specifications that are specified in the assembly editor, which define a set of communication characteristics of an application for:
 - Transmission priority
 - Level of route reliability
 - Transaction management
 - Security level
- Qualifiers define the level of management that a component requires at run time and can be set on:
 - Interfaces
 - References
 - Implementations
- Some implementations determine how qualifiers must be set
 - In these situations, the system generates the qualifiers for you

© Copyright IBM Corporation 2016

Figure 6-7. Quality of service qualifiers

WB8601.2

Notes:

The system generates qualifiers that promote the propagation of transactions and reliable asynchronous message delivery where possible.

For your reference, a list of supported quality of service qualifiers is provided in IBM Process Server in the appendixes. For more information about quality of service qualifiers, see the course WB858: *Developing Applications in IBM Business Process Manager Advanced V8.5.5 – II*, which is the second week of training for version 8.5.5 of the product.

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)
- Many callers use and expose several export components with different binding types

© Copyright IBM Corporation 2016

Figure 6-8. Export binding types

WB8601.2

Notes:

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. A component does not have more than one export with the same binding type.

Supported protocols for the web service binding include SOAP1.1/HTTP, SOAP1.2/HTTP, SOAP1.1/HTTP using JAX-RPC, and SOAP1.1/JMS. JMS can use various transport types, including TCP/IP and HTTP or HTTPS. The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported.

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 (same for export binding types):
 - 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 2016

Figure 6-9. Import binding types

WB8601.2

Notes:

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. A component does not call more than one import with the same binding type.

JCA adapter EIS bindings (1 of 2)

- Adapters:
 - Provide a service façade for existing or custom-built applications that are not exposed as services
 - Simplify interaction with potentially complex systems
- Business value of JCA EIS binding:
 - EIS bindings are standardized by using Java Connector Architecture
 - EIS export bindings allow enterprise information systems (for example: Oracle applications and SAP) to access SCA applications
 - EIS import bindings allow SCA applications to access EIS applications as if they were exposed as services
 - JCA adapters act as the intermediary between the EIS and IBM Process Server

Note: Adapters are covered in a later unit of this course.

© Copyright IBM Corporation 2016

Figure 6-10. JCA adapter EIS bindings (1 of 2)

WB8601.2

Notes:

Enterprise information system (EIS) bindings provide connectivity between SCA components and an external EIS. This communication is achieved by using EIS exports and EIS imports that support JCA 1.5 resource adapters and IBM WebSphere Adapters.

Your SCA components might require that data be transferred to or from an external EIS. When you create an SCA module that requires such connectivity, you include (in addition to your SCA component) an import or export with an EIS binding for communication with a specific external EIS.

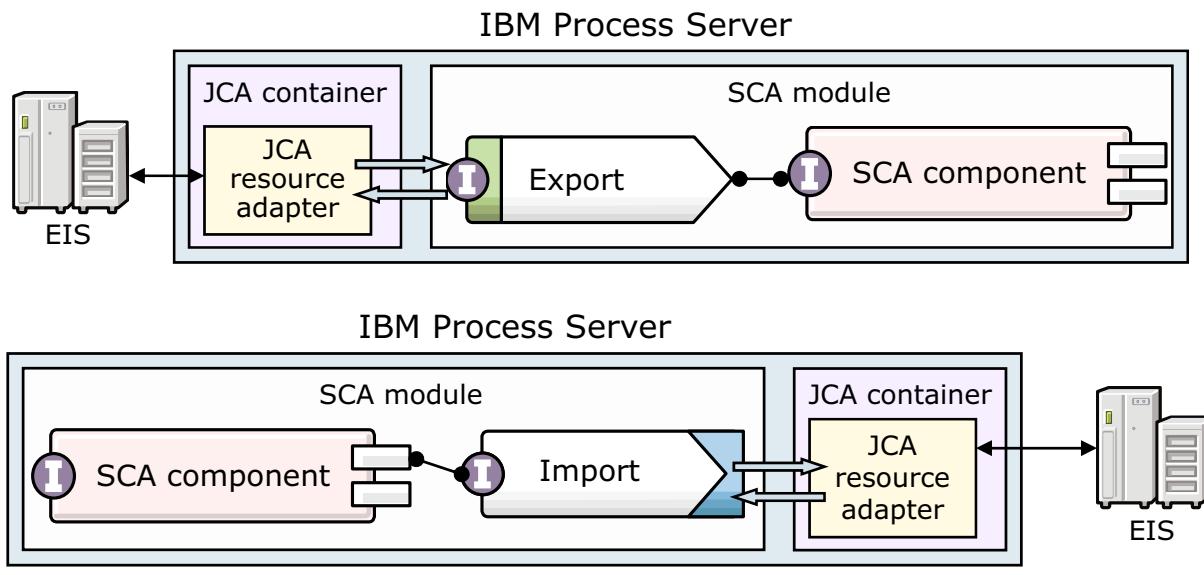
Resource adapters in IBM Integration Designer are used within the context of an import or an export. You develop an import or an export with the external service wizard, and in developing it, include the resource adapter. By using an EIS import, your application can invoke a service on an EIS system. By using an EIS export, your application on an EIS system can invoke a service that is developed in IBM Integration Designer. For example, you would create an import with the JD Edwards adapter to invoke a service on the JD Edwards system. When you use the external service wizard, the EIS binding information is created for you. After the SCA module that contains the EIS binding is deployed to the server, you can use the administrative console to view information about the binding or to configure the binding. The external service wizard for application

adapters creates imports and exports that represent a service on an EIS system. The bindings that are created are EIS bindings.

Adapters are covered later in this course.

JCA adapter EIS bindings (2 of 2)

- The WebSphere Application Server JCA container manages communication between the EIS, the adapter, and the export or import



© Copyright IBM Corporation 2016

Figure 6-11. JCA adapter EIS bindings (2 of 2)

WB8601.2

Notes:

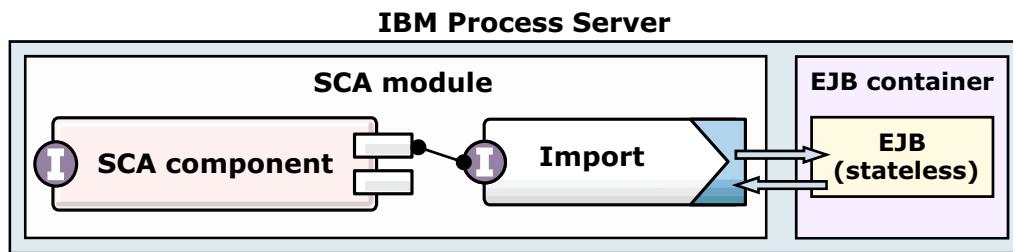
When an adapter is used, the module that contains that adapter has an export component that is bound to the adapter. The WebSphere Application Server JCA container that manages the instance of the adapter is cognizant of an external system update or request.

When an import component that is adapter-bound is used, the import component drives the JCA container and its associated JCA resource adapter. The resource adapter, in turn, contains logic to drive the back-end system. The import component has no specific knowledge of the back-end system. The JCA resource adapter handles that communication.

Adapters are covered in greater detail later in this course.

Enterprise JavaBeans (EJB) binding (1 of 2)

- EJB encapsulate business logic in Java EE applications, and they are divided into three types:
 - **Session:** Created by a client session (stateless session beans do not maintain state information, but stateful beans do)
 - **Entity:** Represents a persistent data object like a record in a relational database
 - **Message-driven:** Bean that responds to JMS messages
- Business value of stateless session EJB binding:
 - Access to existing Java EE applications and functions
 - Easy bridge from environments that are not Java (for example, .NET) to Java EE
 - Protection of investment in Java EE design and development



© Copyright IBM Corporation 2016

Figure 6-12. Enterprise JavaBeans (EJB) binding (1 of 2)

WB8601.2

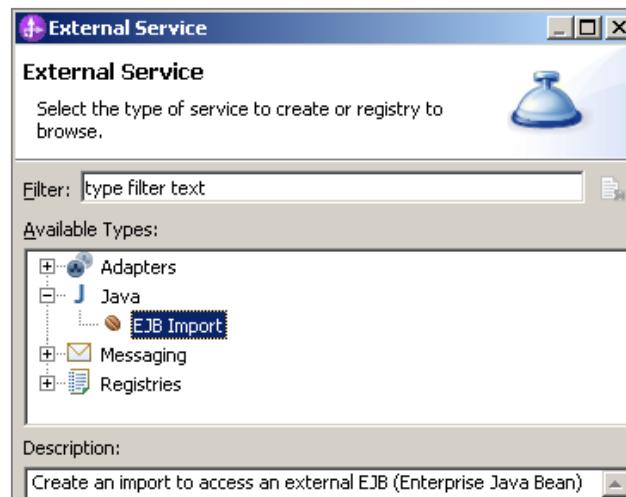
Notes:

Enterprise JavaBeans beans, or EJB beans, come in various types, including session, entity, and message-driven. Of these types, the stateless session bean is one of the most commonly used. Similar to a service, a stateless session bean exposes an interface and allows remote callers to invoke it in a stateless fashion. IBM Process Server provides first class support for this type of bean. Imports and exports can have a stateless session bean binding. EJB imports support working with EJB V3.0 remote and local interfaces and support EJB V2.1 local interfaces.



Enterprise JavaBeans (EJB) binding (2 of 2)

- Drag the session EJB onto the assembly diagram to create EJB import, WSDL, Java interface, or business objects
 - When invoked, an import calls the EJB
 - WebSphere Application Server EJB container handles the communication between an import and the EJB
 - Drag Java interface onto the assembly diagram to create an EJB export
 - Java EE applications use the EJB programming model to invoke
- Use the External Service wizard
 - Select an EJB import or export
 - Then, select the EJB



© Copyright IBM Corporation 2016

Figure 6-13. Enterprise JavaBeans (EJB) binding (2 of 2)

WB8601.2

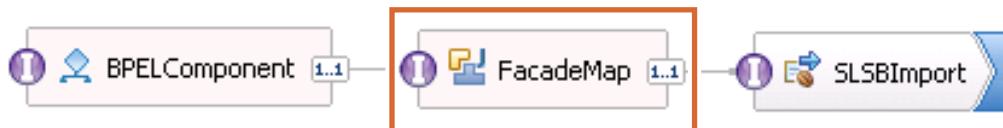
Notes:

In IBM Integration Designer, you can drag a stateless session bean descriptor onto the SCA assembly diagram. This action creates an import component that, when called, calls the EJB bean.

Another way to automatically populate the assembly diagram with an EJB import is to use the external service wizard. A new Java option is added to the wizard. You can select the appropriate Java file, and it prompts you for the interface type (WSDL or Java) of the EJB import that you want generated on the assembly diagram.

WSDL to Java interface mapping for EJB beans

- To wire an EJB Java interface to an SCA component with a WSDL interface, a map is required
 - For example: A business process has a WSDL interface that cannot be wired directly to an EJB import
- A façade mapping component can be created automatically
 - IBM Integration Designer prompts you when you drag the EJB descriptor onto the assembly diagram
 - The façade WSDL interface of the component is wired to the SCA component
 - The Java reference of the façade is wired to the import
 - For exports, the reference of the export is wired to the façade, and the reference of the façade is wired to the SCA component



© Copyright IBM Corporation 2016

Figure 6-14. WSDL to Java interface mapping for EJB beans

WB8601.2

Notes:

An SCA POJO mapping component can be created by using IBM Integration Designer or by using command line. The mapping component comprises a WSDL file that is generated by using the `java2WSDL` tool, a `.component` file, and a `.java` file. The `java2WSDL` tool has the implementation of the POJO mapping component. The `java2WSDL` tool is a utility that is included with WebSphere Application Server.

SCA bindings (1 of 2)

- SCA binding:
 - Default binding
 - Provides communication between SCA modules or applications
 - Cannot be used to communicate with applications outside the IBM Process Server environment
- Value in interaction:
 - Allows optimized and most efficient communication between components in the same JVM (Java virtual machine)
- The SCA runtime exposes an API that allows a caller to invoke an SCA described service
 - If the service is also exposed through SCA, it can map the caller to the called code
- After the module is deployed to the server, use the administrative console to view information about the binding
 - For import binding, you can change selected properties of the binding

© Copyright IBM Corporation 2016

Figure 6-15. SCA bindings (1 of 2)

WB8601.2

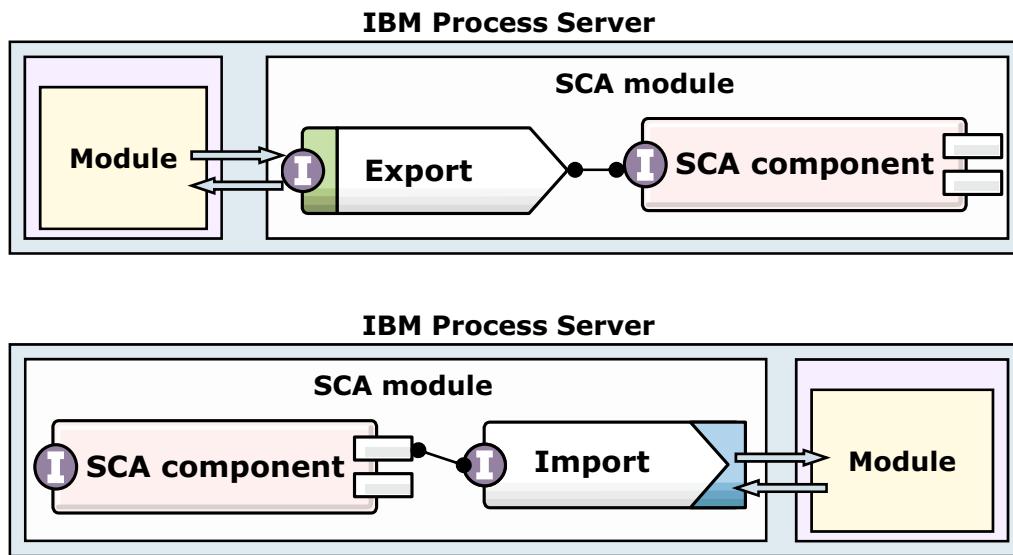
Notes:

Consider an SCA binding when these factors are applicable:

- All services are contained in IBM Integration Designer modules; that is, no external services exist.
- Performance is important.
- The modules are tightly coupled.

SCA bindings (2 of 2)

- If a service is exposed with an SCA bound export, another SCA module can invoke the service by using an SCA bound import component



© Copyright IBM Corporation 2016

Figure 6-16. SCA bindings (2 of 2)

WB8601.2

Notes:

One of the capabilities of SCA is to describe an interface that can be called. The SCA runtime environment can decide how the interface call is made. If the calling code and the SCA module exist in the same runtime environment, it does not make sense to marshal or encode the data only to unmarshal it again when it reaches the destination.

The SCA runtime environment exposes an API that allows a caller to invoke an SCA-described service and, if the service is also exposed through SCA, to map the caller to the called code. An export component can be given an SCA binding, which allows a calling application in the same Java virtual machine to call that module as efficiently as possible. Examples of calling applications include EJB beans, JavaServer Pages (JSP pages), servlets, message-driven beans, and other Java EE artifacts.

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 messaging
 - Supports configurable message correlation (inbound and outbound)
 - Supports event sequencing for exports
 - Supports configurable reply connection for imports
 - Supports both publish and subscribe
 - Supports queue-based messaging
- Three types of JMS bindings:
 - JMS/MQ
 - JMS (WebSphere Application Server)
 - Generic JMS (for independent vendor JMS providers)

© Copyright IBM Corporation 2016

Figure 6-17. JMS bindings (1 of 2)

WB8601.2

Notes:

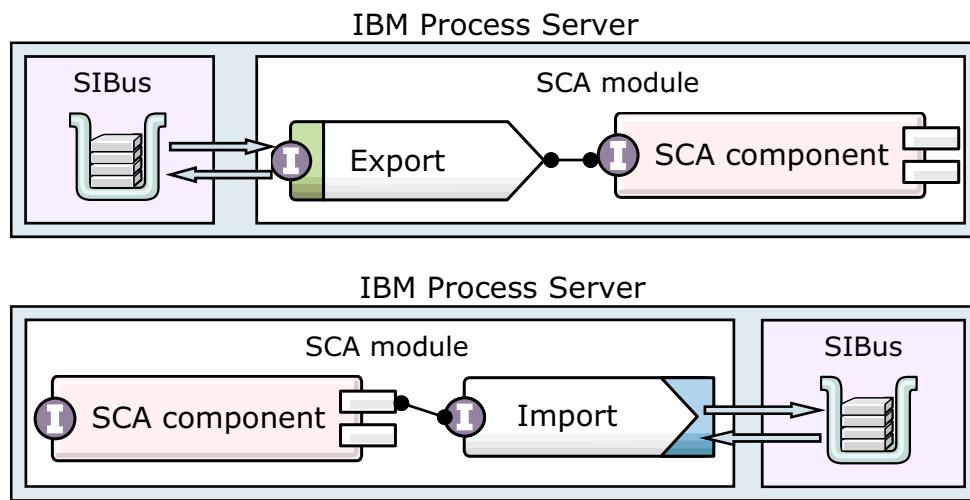
Java Message Service is a specification that describes how a Java application can send and receive messages. It supplies an API and semantics that are vendor independent. IBM implements JMS in the service integration bus (WebSphere Platform Messaging), IBM Process Server, and WebSphere MQ. If a Java 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

JMS bindings (2 of 2)

- Providing an export component with JMS binding allows 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 2016

Figure 6-18. JMS bindings (2 of 2)

WB8601.2

Notes:

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 a message that is produced when the service interface is called.

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
 - Supports correlation schemes and event sequencing
 - Security by using authentication aliases
 - Obeys 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 2016

Figure 6-19. Generic JMS bindings

WB8601.2

Notes:

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.

WebSphere MQ bindings (1 of 2)

- WebSphere MQ:
 - Is a message-oriented middleware product
 - Provides reliable, resilient application integration by passing messages between applications and web services
 - Uses queuing and transactional facilities that help preserve the integrity of messages across the network
- Business value of bindings:
 - Provides easy integration with WebSphere MQ, WebSphere Application Server, WebSphere MQ Workflow, and IBM Integration Bus
 - Exports with WebSphere MQ binding allow external applications to drive execution of an IBM Process Server application

© Copyright IBM Corporation 2016

Figure 6-20. WebSphere MQ bindings (1 of 2)

WB8601.2

Notes:

Consider a WebSphere MQ binding when these factors are applicable:

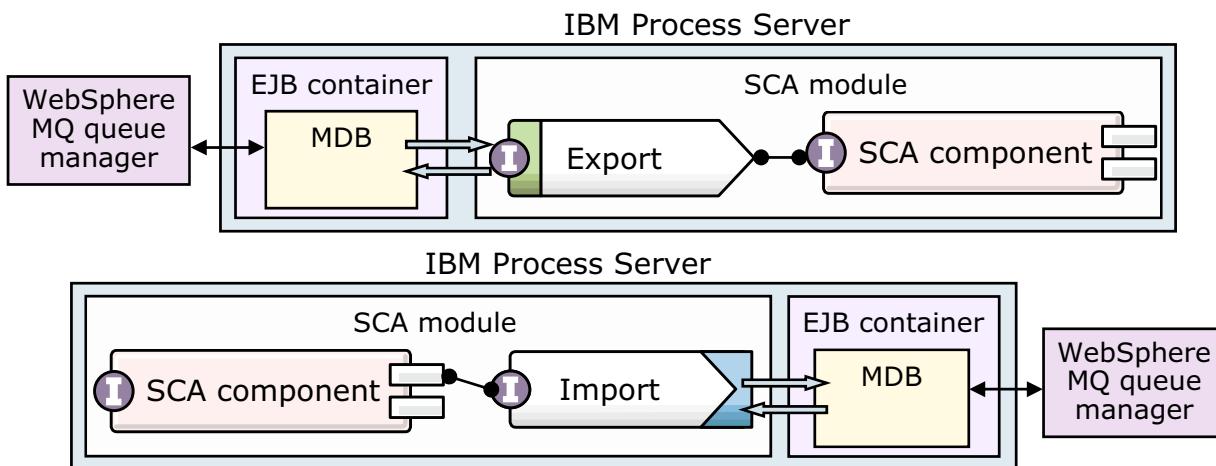
- You must access a WebSphere MQ messaging system, and you must use the WebSphere MQ basic functions.
- The services are loosely coupled.
- Reliability is more important than performance; that is, asynchronous data transmission is preferred over synchronous.

Consider an MQ/JMS binding when these factors are applicable:

- You must access a WebSphere MQ messaging system but can do so within a JMS context; that is, the JMS subset of functions is good enough for your application.
- The services are loosely coupled.
- Reliability is more important than performance; that is, asynchronous data transmission is preferred over synchronous.

WebSphere MQ bindings (2 of 2)

- When a message arrives on a queue, IBM Process Server monitors that queue (by using a message-driven bean)
- An import component with a WebSphere MQ binding provides an IBM Process Server module with the capability to drive the execution of an external application by sending a message to a queue



© Copyright IBM Corporation 2016

Figure 6-21. WebSphere MQ bindings (2 of 2)

WB8601.2

Notes:

WebSphere MQ export binding allows an external WebSphere MQ application to drive the execution of an IBM Process Server application. When a message arrives on a WebSphere MQ queue, IBM Process Server can be configured to monitor that queue. WebSphere MQ import binding allows an IBM Process Server application to drive the execution of a WebSphere MQ application by sending a message to a queue.

On the Propagation tab in the Bindings section of the Properties view, you can enable the **Propagate** operation name option for the binding. Use it to add an RFH2 header to the outbound message that identifies the operation that is associated with the message.

The WebSphere MQ JCA resource adapter supports the handling of basic WebSphere MQ headers, which is not possible with the JMS API and the old WebSphere MQ JMS provider in prior versions.

HTTP bindings

- HTTP bindings use HTTP or HTTPS transport protocols to provide connectivity to an SCA-based application
- Business value:
 - Clients can reuse and build upon existing HTTP infrastructures
 - Brings benefits of SOA to HTTP deployments and allows SOA applications to take advantage of widespread HTTP applications (including Web 2.0 applications)
- HTTP bindings for imports and exports support:
 - HTTP 1.0 – 1.1 and SSL over HTTP
 - Synchronous request/response invocation only
 - Binary, XML, and SOAP payloads (plus custom data bindings)
 - Specifiable content and transfer encodings
 - Custom HTTP methods in imports
 - Endpoint-based routing in exports
 - Ability to modify the HTTP binding attributes at run time

© Copyright IBM Corporation 2016

Figure 6-22. HTTP bindings

WB8601.2

Notes:

Enabling the HTTP binding in the business integration world helps you to easily compose HTTP-based services and to enhance the accessibility, performance, and scalability of your SCA applications. You can use the binding to allow an SCA application to use a wide variety of HTTP-based applications that exist on the World Wide Web. You can also create endpoints for HTTP clients, including the ones that can run in a simple web browser. Finally, you can more easily integrate your business integration applications with new emerging Web 2.0 technologies.

What is the HTTP binding?

Hypertext Transfer Protocol (HTTP) is a widely used protocol for transferring information about the web. Though originally designed to publish and retrieve HTML pages, it has now become a standard request/response protocol between clients and a server. It is defined according to the HTTP protocol that the World Wide Web Consortium (W3C) publishes. Today, many standard HTTP methods such as GET, POST, PUT, and DELETE are a part of this widely used protocol.

When working with an external application by using the HTTP protocol, an HTTP binding is necessary. The HTTP binding handles the transformation of data that is passed in as a message in a basic format to a business object in an SCA-based application. The HTTP binding also can

transform data that is passed out as a business object to the basic format that an external application expects for an incoming message.

Why has the HTTP binding grown in importance?

When the SOAP/HTTP (referred to as SOAP over HTTP) web services binding gained momentum, businesses began seeing the potential of the HTTP protocol for business needs and application integration. SOAP/HTTP web services became the most common means for business-to-business transactions over the Internet. However, adding more robustness to SOAP/HTTP to suit business needs led to interoperability problems among vendors and more complexity of the initial specification. These robust services included the Web Service Description Language (WSDL), supporting the XML schema specification, and adding quality of service (QoS). It also led to the need for more skills to use SOAP/HTTP.

Users realized that many situations did not require the extra functions added to SOAP/HTTP. They merely wanted to take advantage of the ubiquitous web infrastructure to send or receive relatively simple information by using the HTTP protocol. They started by using HTTP-based services to send and receive data in a set of different loosely defined formats.

The HTTP binding is suited for this type of user as it combines the ease of use and simplicity of the original HTTP protocol with integration to large, scalable, and secure SOA applications. Conversely, the HTTP binding allows SOA applications to take advantage of the many existing HTTP-based applications, bringing them into the SOA framework. The binding also provides access from an SOA application to services that conform to the Web 2.0 specification. In summary, the HTTP binding allows applications that are developed for IBM Process Server to communicate with and mediate between the many web services by using HTTP and other protocols. This communication capability will make this binding only more important over time.

Comparing the HTTP binding to the web services binding

Another binding, the web services binding, also can be used with applications by using the HTTP protocol. The difference between the web services binding and the HTTP binding is as follows:

- The web services binding supports the SOAP (and JMS) protocols only.
- The web services binding assumes that it is working with web services-based applications and so exposes the same model. In contrast, the HTTP binding assumes that it is working with basic HTTP applications and exposes a model more familiar to this audience.
- Therefore, the web services binding provides first class support within the SCA architecture for web services applications that communicate with the HTTP protocol and other protocols. In contrast, the HTTP binding allows IBM Process Server to mediate between, and communicate with, any HTTP application, thus bringing any HTTP application into the service-oriented architecture framework.

HTTP binding at run time

In IBM Integration Designer, the HTTP binding can be used on imports and exports. An import with an HTTP binding at run time sends a request with or without data in the body of the message from the SCA application to the external web service. In other words, the request is made from the SCA application to the external web service. Optionally, the import with the HTTP binding can receive data back from the web application in a response to the request.

With an export, a client application to a web service makes the request. The web service is a web application that runs on the server. The export is implemented in that web application as a servlet so the client sends its request to a URL address. The servlet passes the request to the SCA application in the runtime environment. Optionally, the export can send data to the client application in response to the request.

HTTP bindings use HTTP-centric features. Messages are presented to SCA or mediation components in a manner that preserves HTTP protocol and message header information, which provides a more familiar view to HTTP application programmers, users, and administrators.

HTTP imports and exports can be configured to support a range of common HTTP features. They support HTTP 1.0 and 1.1 protocols and HTTPS, which uses SSL for secure HTTP conversation. The supported request/response mechanism is synchronous only. The binding supports static and dynamic HTTP headers (dynamic header setting access is done through mediation modules).

An existing DataBinding framework is extended for HTTP conventions and provides mapping between SCA messages and HTTP message headers and bodies. The IBM supplied data bindings support several different data payloads. The supported payloads are XML, binary, and SOAP. Users can create their own custom data bindings.

You can use the HTTP import to select the HTTP method, version, and other configuration. You can use the HTTP export to specify the endpoint URL for the client to invoke the services.

When the application with HTTP binding is deployed on the server, the administrator can modify many of the HTTP binding attributes by using the administrative console or the `wsadmin` commands.

The HTTP binding handles the transformation of data that is passed in as a message in a basic format to a business object in an SCA-based application. The HTTP binding also can transform data that is passed out as a business object to the basic format that the external application expects.

6.2. Web services



Web services



© Copyright IBM Corporation 2016

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

10.1

Figure 6-23. Web services

WB8601.2

Notes:

Overview: Web services

- Web services connect businesses to each other and invoke services with appropriate security, reliability, and confidentiality
- If XML defines a platform-independent standard way to represent data, then web services define a platform-independent exchange for data
 - Application integration becomes easier
 - Web services use core technologies: XML, WSDL, and SOAP
- XML (Extensible Markup Language)
 - XML solves the problem of data independence
 - Use XML to describe data and to map that data into and out of any application
- WSDL (Web Services Description Language)
 - XML-based language to create a description of an underlying application
 - The description turns an application into a web service by acting as the interface between the underlying application and other web-enabled applications
- SOAP
 - SOAP is the core communications protocol for the web
 - Most web services use this protocol to communicate with each other
 - SOAP can be used over HTTP or JMS

© Copyright IBM Corporation 2016

Figure 6-24. Overview: Web services

WB8601.2

Notes:

Web services are the common denominator across the industry, and virtually all environments support them. It is highly likely that users of IBM Process Server have web service-based components that they want to integrate. Web services are the common denominator for interoperation because they are standardized, supported by all middleware vendors and environments, and consistent across platforms and architectures.

Web services deliver key business value:

- Platform and language neutral
- Transport and data independent
- Designed to work well with the web (HTTP)
- Foundation for service-oriented architecture (SOA)
- Allow reuse of existing applications and infrastructure
- Ease of integration with IBM Integration Designer and IBM Process Server
- Easily choreographed in business processes

WSDL is an XML document format for describing web services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented (RPC) messages. A WSDL document contains several parts:

- **Types** elements define custom data types for the service.
- **Message** elements declare message names and parameters.
- **PortType** elements list available operations.
- **Binding** elements describe supported protocols.
- **Service** elements map bindings to specific network addresses.



Discovering web services

- You typically discover service descriptions through a Universal Description, Discovery, and Integration (UDDI) registry or through WebSphere Service Registry and Repository
- By using WebSphere Service Registry and Repository, you can store, access, and manage information about services
 - You can use this information to select, invoke, and reuse services
- You can use WebSphere Service Registry and Repository to store information about services in your systems or in other systems that you already use, plan to use, or want to track
 - An application can check WebSphere Service Registry and Repository before it invokes 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
 - You can use the external service wizard with WebSphere Service Registry and Repository

© Copyright IBM Corporation 2016

Figure 6-25. Discovering web services

WB8601.2

Notes:

WebSphere Service Registry and Repository is the master metadata repository for service interaction endpoint descriptions. A broad definition of “service” applies here. This definition includes traditional web services that implement WSDL interfaces with SOAP/HTTP bindings and a broad range of SOA services. These services can be described by using WSDL, XSD, and policies, but might use a range of protocols and be implemented according to various programming models.

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 and other service metadata and endpoint registries and repositories. An example is Universal Description, Discovery, and Integration (UDDI), where service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service. Then, visibility is controlled and versions are managed. The proposed changes are analyzed and communicated. The usage is monitored, and other parts of the SOA foundation can access the service metadata. They can be confident 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.

In the other SOA lifecycle stages, WebSphere Service Registry and Repository is complementary with repositories that specialize in managing SOA artifacts during the development stages of the lifecycle. For example, a development artifact management system such as Rational ClearCase takes care of service and composite application building blocks. These building blocks can include 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 that are described assets according to the Reusable Asset Specification (RAS) standard. They implement governance processes that control the promotion of artifacts to assets and the approval process that is associated with them.

When the development team finishes its work and testing is complete, deployment teams further augment the service metadata, providing binding information for service endpoints that are used in composite applications. They manage deployment of the metadata from the development environment to the staging or production WebSphere 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 can 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.

Again, new service metadata, or more often a change in existing service metadata, can be discovered in other service endpoint registries and repositories. It is published in WebSphere Service Registry and Repository, and can be used as input for the application configuration and binding tasks that are the responsibility of deployment teams and solution administrators.

Discovered service metadata is incomplete and not yet suitable for broader visibility and consumption. Deployment teams work with asset managers to ensure that the metadata is augmented with the necessary semantics, permissions, and scoping constraints.

Importing a web service WSDL interface

- When you discover and import a web service WSDL interface and drag it onto an assembly diagram in IBM Integration Designer, a web service bound import is created to call the service
- The component then refers to a web service endpoint
 - This endpoint is the location where the web service is listening for incoming requests
- An import with a web service binding results in a web service call to an external partner that uses SOAP over HTTP or SOAP over JMS

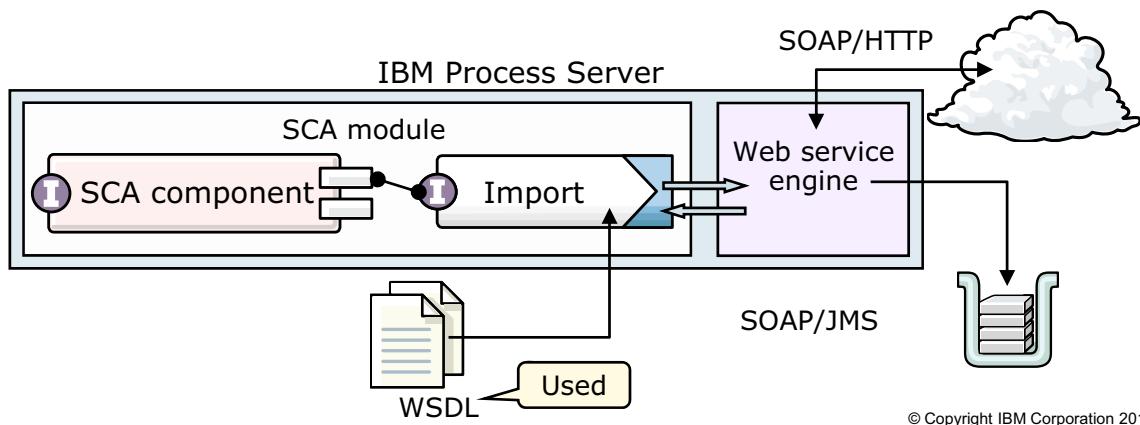


Figure 6-26. Importing a web service WSDL interface

WB8601.2

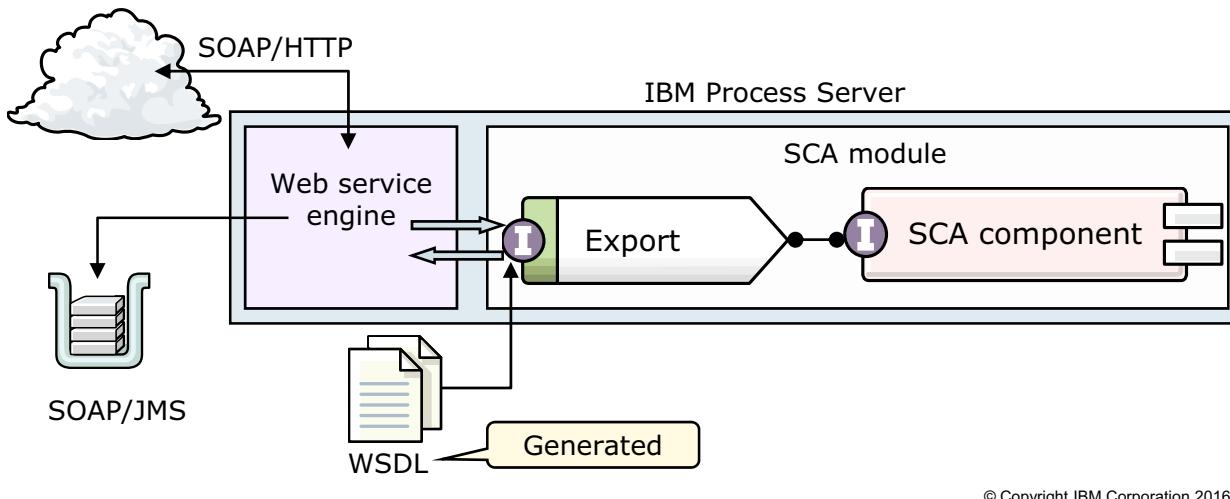
Notes:

An import component that is bound to a web service binding results in a web service call to an external partner when invoked. The request sent to the partner is either SOAP over HTTP or SOAP over JMS, depending on the binding request used. In this scenario, the external service that is called exists, so it is likely that a WSDL file describing that service is supplied.

When imported into IBM Integration Designer, the WSDL file can be directly inserted onto the SCA assembly diagram, causing the creation of “a ready-to-use web service bound import”. The WebSphere Application Server web service engine is used to build the request, send the request, and listen for the response.

Web service export binding (1 of 2)

- Exports with web service binding declare that the module exposes an interface that can be invoked remotely as a web service
- The export uses SOAP over HTTP or SOAP over JMS as the transport protocol
- When a web service export is declared, an associated WSDL file is generated



© Copyright IBM Corporation 2016

Figure 6-27. Web service export binding (1 of 2)

WB8601.2

Notes:

Binding to web services is probably the most common of the binding types. An export component with a web service binding declares that the module exposes an interface, which can be invoked remotely as a web service. The export component can use either the SOAP over HTTP or the SOAP over JMS transport protocol. When a web service export is declared, an associated WSDL file is generated that can be given as-is to partners who want to invoke the module.

When a module with a web service export is deployed, the web service engine starts listening on behalf of the service. When a request arrives, this listener causes the instantiation of a module, and control is then passed to that module. The SCA runtime environment is a consumer of the web services engine and exposes SCA modules as web services through SOAP/HTTP.

Through the business object service, in-memory objects can be externalized as XML, or XSD contained in WSDL. This externalization is a convenient way of dealing with objects and XML directly. It also means that you can take Java objects and treat them as XSD.

The following IBM Process Server components are involved in web service communication:

- Service Component Architecture runtime environment
 - Client to the web service engine

- Consumer of external web services
- Architecture that is not centered on Java
- Components that are exposed as SOAP and HTTP services
- Client to “external” SOAP/HTTP services
- Business object service (in-memory)
 - Used by clients to access XSD/WSDL artifacts
 - Direct serialization between XML and Java (SDO)
- Web service engine
 - Part of the base WebSphere Application Server
 - JAX-RPC handlers (used for mediating SOAP messages) invoked for traditional bindings
 - Service integration bus

WebSphere Education

Web service export binding (2 of 2)

- Web service bindings support JAX-WS based web services
- JAX-WS bindings support:
 - WebSphere policy sets
 - SOAP 1.2 and SOAP 1.1
 - Attachments (unreferenced, referenced, and swaRef)
- JAX-WS 2.2 with MTOM support
- The older JAX-RPC web service binding is supported but is deprecated

Select a Transport Protocol

Select the transport protocol that CreditScoreServiceExport1 will use for the Web service binding.

SOAP1.2/HTTP

Supports features such as JAX-WS handlers, WS-policy sets, and service gateways by using JAX-WS. [More...](#)

SOAP1.1/HTTP

Supports features such as JAX-WS handlers, WS-policy sets, and service gateways by using JAX-WS. [More...](#)

SOAP1.1/HTTP using JAX-RPC

Supports JAX-RPC handlers and RPC encoded messages. [More...](#)

SOAP1.1/JMS

Uses JMS synchronously as the underlying transport protocol. [More...](#)

© Copyright IBM Corporation 2016

Figure 6-28. Web service export binding (2 of 2)

WB8601.2

Notes:

IBM Integration Designer provides configuration options to enable JAX-WS bindings to send and receive web service messages, which include SOAP Message Transmission Optimization Mechanism (MTOM) attachments. This mechanism improves the transmission efficiency of large binary attachments in SOAP messages.

Before configuring a JAX-WS import or export binding to enable MTOM support, it is important to know that not all scenarios are supported:

- MTOM is not supported when the business object parsing mode is set to eager parsing (support is limited to a JAX-WS web service by using business object lazy parsing mode).
- MTOM is not supported when using a JAX-WS handler (support is limited to a JAX-WS web service, which does not use any JAX-WS handlers). The JAX-WS handlers that are specified on the web service should be removed.
- When using a service gateway mediation module, the Data Handler primitive cannot be used with MTOM messages. If direct access to the MTOM attachment data is required within the module, then a non-service gateway module must be used.

- When MTOM is enabled on a JAX-WS export binding, all responses are sent by using MTOM. If some clients do not support MTOM, use two JAX-WS exports, one with MTOM enabled and one with it disabled, and ensure that client applications use the correct endpoint address.
- MTOM is not supported when using the JAX-RPC binding for SOAP/HTTP or SOAP/JMS.

You can send and receive SOAP messages that include attachments that are represented in the service interface as `swaRef`-typed elements.

A `swaRef`-typed element is defined in the Web Services Interoperability Organization (WS-I) *Attachments Profile* Version 1.0 (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), which defines how message elements are related to MIME parts.

In the SOAP message, the SOAP body contains a `swaRef`-typed element that identifies the content ID of the attachment.

Policy sets reduce the complexity of configuring web services by providing reusable configurations.

A policy set is a collection of policy types, each of which provides a quality of service. These types are configured and can be associated with a web service provider or consumer. Policy sets work in pairs. You must have the same policy set on the service requester as on the service provider.

Therefore, on an export, you should have the same policy set on the export binding as on the client. On an import, you should have the same policy set on the import binding as on the service provider it is calling.

A policy set can be associated with imports and exports. Only the SOAP1.2/HTTP and the SOAP1.1/HTTP transport protocols support policy sets. A policy set cannot be associated with the SOAP1.1/HTTP by using the JAX-RPC transport protocol.

Testing web services

- When you are testing web service exports, you can import data from XML or HTTP files
 - SOAP messages are stripped from the file
 - Input SOAP messages are editable
 - An HTTP file is captured from the TCP/IP Monitor or attached
- The integration test client has a full XML editor (value editor and XML editor) with syntax validation

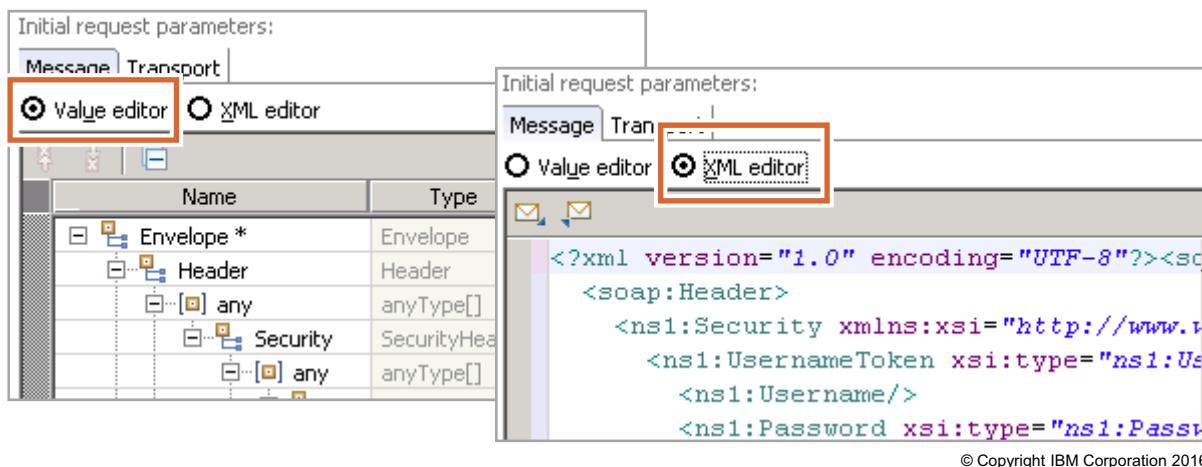


Figure 6-29. Testing web services

WB8601.2

Notes:

In the integration test client, you can interactively send sample SOAP messages without attachments to a web service export and view the SOAP response.

Before testing a web service export with a SOAP message, you must have an existing SOAP message that you can use for testing. If you have HTTP messages that contain SOAP messages, they must be in *.http files. If you have pure SOAP messages with no attachments, the SOAP messages must be in *.xml files.



Unit summary

Having completed this unit, you should be able to:

- List the various types of SCA import and export bindings
- Describe how SCA bindings facilitate integration with different types of applications
- Describe how web services are used in the Service Component Architecture framework

© Copyright IBM Corporation 2016

Figure 6-30. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. Define bindings in the context of SCA imports and exports.
2. What is the value of providing WebSphere MQ bindings for imports and exports?
3. What types of asynchronous invocation styles do SCA components support?
4. What is the difference between the JMS binding and the generic JMS binding?

© Copyright IBM Corporation 2016

Figure 6-31. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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

Checkpoint answers

1. SCA binding information provides a consistent means for communicating across module boundaries in IBM Process Server. Bindings determine how your import and export components interact with clients outside a module. Bindings specify the protocol, message format, and invocation style.
2. It provides use for the huge installation base. It provides easy integration with WebSphere MQ, WebSphere Application Server, WebSphere MQ Workflow, and WebSphere Message Broker.
3. One-way, deferred response, and request with callback.
4. The JMS binding provides preconfigured resources that are optimized for WebSphere Application Server and IBM Process Server. Resources are automatically configured during deployment. Generic JMS does not contain preconfigured resources and does not create resources automatically during deployment.

© Copyright IBM Corporation 2016

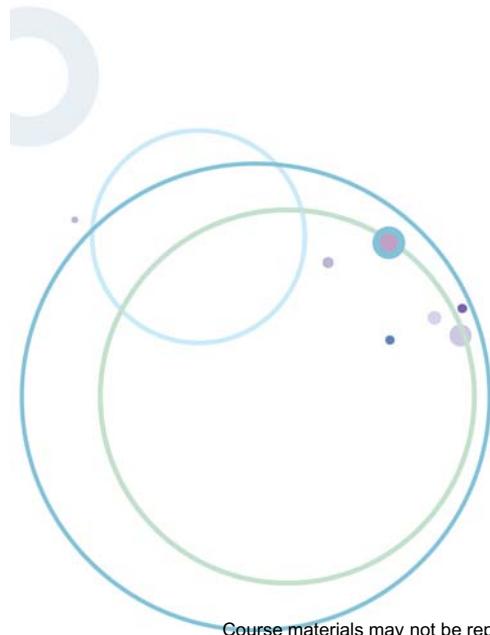
Figure 6-32. Checkpoint answers

WB8601.2

Notes:



Exercise 4



Working with web services

© Copyright IBM Corporation 2016

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

10.1

Figure 6-33. Exercise 4

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Import an external Web Services Description Language (WSDL) file into IBM Integration Designer
- Create an SCA component from a web service interface file
- Use the integration test client to test a web service
- Use a web service export to expose an existing IBM Process Server application

© Copyright IBM Corporation 2016

Figure 6-34. Exercise objectives

WB8601.2

Notes:

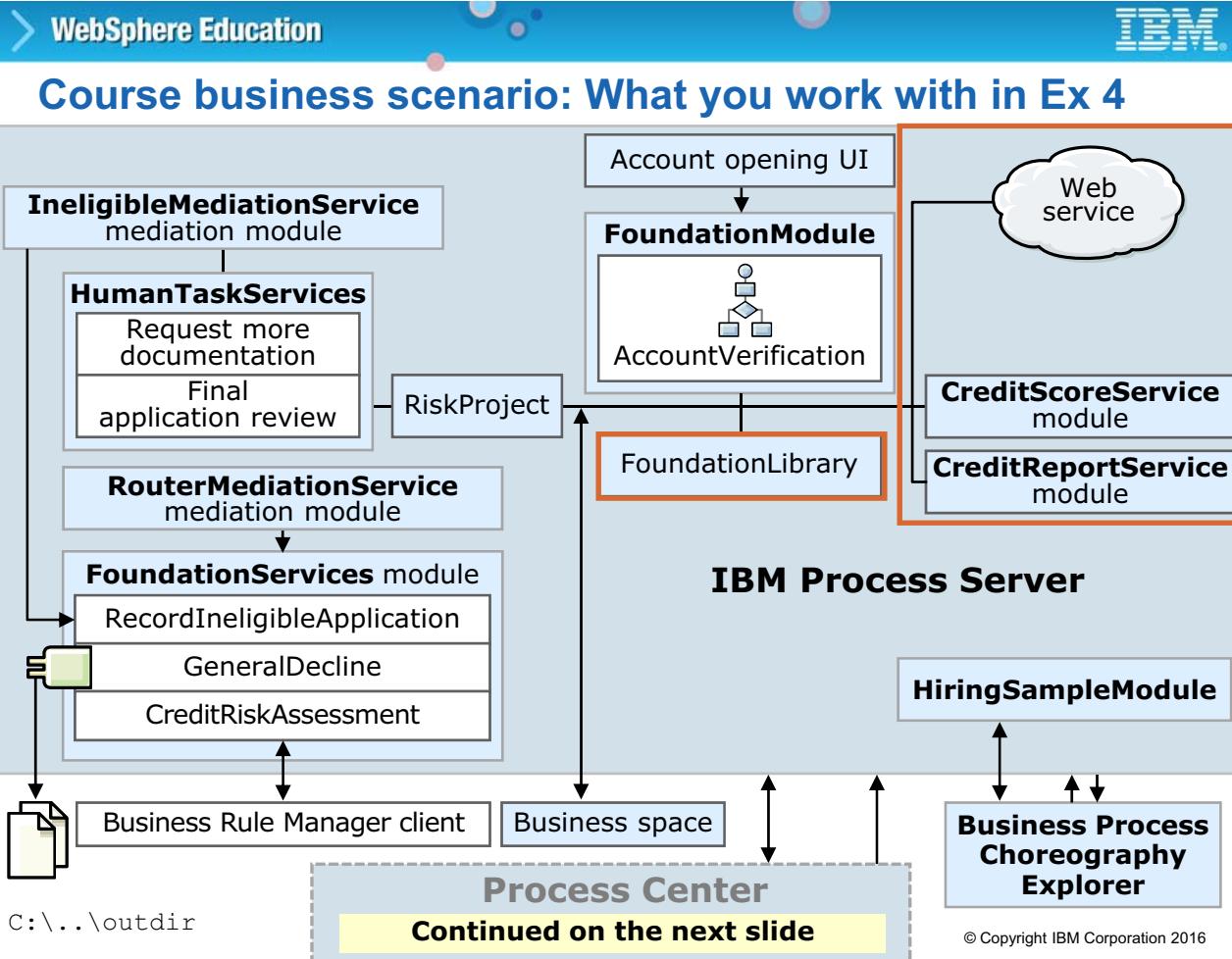


Figure 6-35. Course business scenario: What you work with in Exercise 4

WB8601.2

Notes:

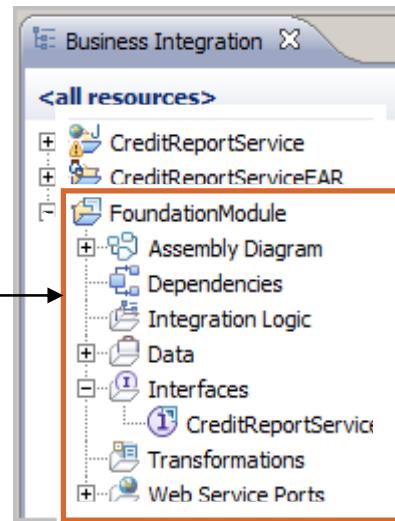
Components that are required for Exercise 4 (Section 1)

Prebuilt components that are imported in the lab:

1. **CreditReportService.war**
 - Contains the **CreditReportServiceEAR** application
 - You deploy the application to IBM Process Server
2. **CreditReportService.wsdl**
 - Interface that describes the **CreditReportService** web service
 - Defines an operation **CalculateCreditScore**
 - Defines the **CreditCheckDetail** business object

New components that you create in the lab:

1. **FoundationModule**
 - Encapsulates and groups artifacts by type
2. **CreditReportService**
 - SCA Import component that invokes the **CreditReportService** web service



© Copyright IBM Corporation 2016

Figure 6-36. Components that are required for Exercise 4 (Section 1)

WB8601.2

Notes:

Business Process Model and Notation, or BPMN, is a standardized graphical notation for creating diagrams of business processes.

BPMN is used so that everyone who is involved can interpret and understand the model.

Throughout development, many different parties are involved in modeling. Every stakeholder, from the least technical to the most technical, understands the model to provide valuable feedback and continuously improve the process.

BPMN also allows a way to compact your process definition. Many of the symbols represent ideas, so symbols allow for a more concise and smaller model than drawing a diagram without BPMN.

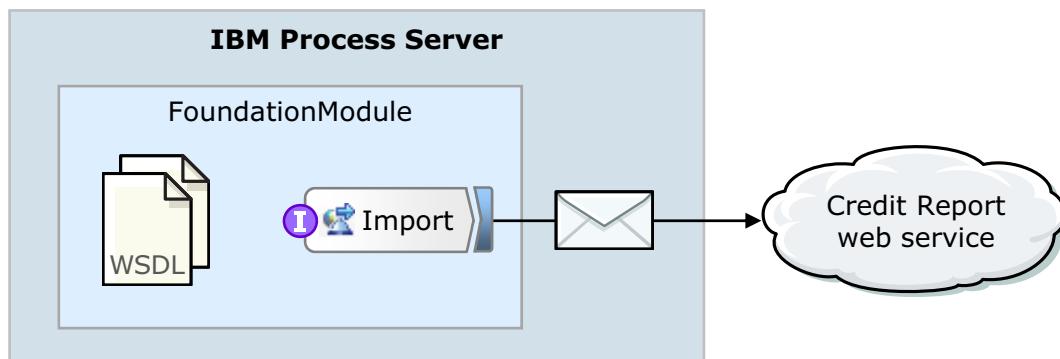
BPMN has many benefits, but most importantly, BPMN creates a standardized bridge for the gap between the business process design and process implementation. This single notation is agreed upon among multiple IBM BPM vendors for the benefit of the user community.

IBM implements and interprets these elements to have specific meanings and terminology in the IBM Process Designer product. For definitions of the BPMN specification, see the BPMN specification document version 2.0 from the Object Management Group. You can learn more about BPMN at: <http://www.bpmn.org>

The element palette of IBM Process Designer is shown in the slide image. These items correspond to BPMN elements and are used in modeling a process.

A BPMN standards group that is called Object Management Group is an international, open membership, not-for-profit technology standards consortium. You can search the Internet for this group.

Invoke an external web service in Exercise 4 (Section 1)



© Copyright IBM Corporation 2016

Figure 6-37. Invoke an external web service in Exercise 4 (Section 1)

WB8601.2

Notes:

Components that are required for Exercise 4 (Section 2)

Prebuilt components that are imported in the lab:

1. CreditScoreService

- Module containing the **CreditScoreRG** rule group that calculates the credit score

2. CreditScoreService.wsdl

- Interface that describes the **CreditScoreService** web service
- Defines the **CreditCheckRequest** business object

New components that you create in the lab:

1. FoundationLibrary

- Container for shared business objects and interfaces:
CreditScoreService.wsdl interface and
CreditCheckRequest business object

2. CreditScoreService

- Export SCA component that exposes the **CreditScoreService** web service

3. CreditScoreService

- Import SCA component that calls the service through the SCA export

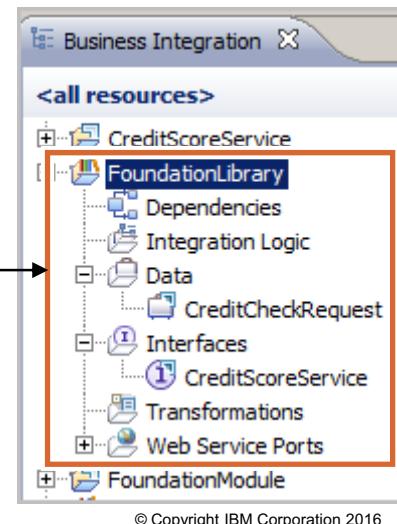
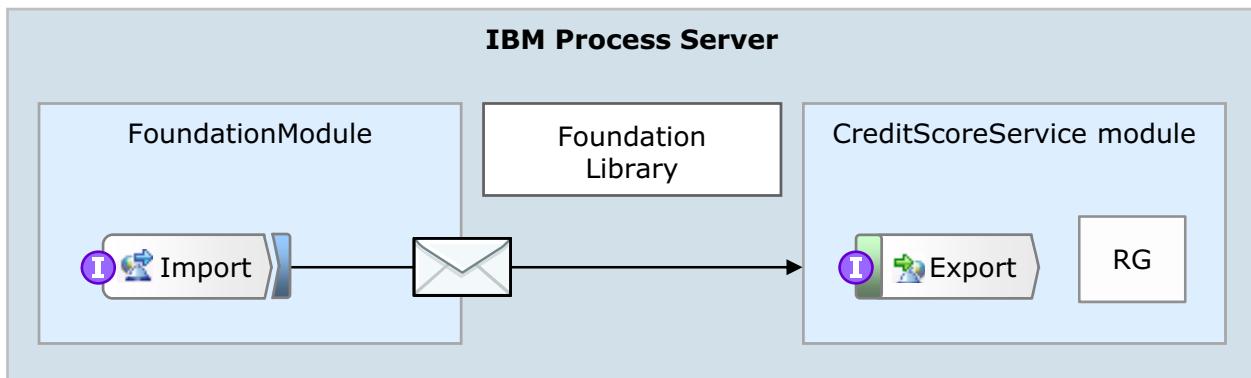


Figure 6-38. Components that are required for Exercise 4 (Section 2)

WB8601.2

Notes:

Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)



© Copyright IBM Corporation 2016

Figure 6-39. Expose an application as a web service that is running in IBM Process Server in Exercise 4 (Section 2)

WB8601.2

Notes:



References

- OSOA: Service Component Architecture family of specifications
 - <http://oasis-open.org/sca>
- developerWorks: Building SOA solutions with the Service Component Architecture
 - Part 1
http://www.ibm.com/developerworks/websphere/techjournal/0510_brent.html
 - Part 2
http://www.ibm.com/developerworks/websphere/techjournal/0512_barclia.html
 - Part 3
http://www.ibm.com/developerworks/websphere/techjournal/0602_barclia.html
- IBM Redbooks: “Business Process Management Deployment Guide Using IBM Business Process Manager V8.5”
 - <http://www.redbooks.ibm.com/abstracts/sg248175.html?Open>

© Copyright IBM Corporation 2016

Figure 6-40. References

WB8601.2

Notes:

Unit 7. Business objects

What this unit is about

This unit explains the Service Data Objects (SDO) architecture and framework, and how SDO is used in a solution.

What you should be able to do

After completing this unit, you should be able to:

- Identify the purpose and advantages of using the SDO framework
- Define the data object and data graph components of the SDO framework
- Identify the purpose and advantages of using the business object framework
- Define the business object and business graph components of the business object framework

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Identify the purpose and advantages of using the SDO framework
- Define the data object and data graph components of the SDO framework
- Identify the purpose and advantages of using the business object framework
- Define the business object and business graph components of the business object framework

© Copyright IBM Corporation 2016

Figure 7-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to Service Data Objects
- Introduction to business objects

© Copyright IBM Corporation 2016

Figure 7-2. Topics

WB8601.2

Notes:

7.1. Introduction to Service Data Objects

Introduction to Service Data Objects



© Copyright IBM Corporation 2016

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

10.1

Figure 7-3. Introduction to Service Data Objects

WB8601.2

Notes:

Introduction to Service Data Objects (SDO)

- SDO provides a framework for data manipulation
 - SDO API unifies representation of data from multiple sources
 - Not necessary to know multiple technology-specific APIs
- SDO framework supports a disconnected programming model (manipulate data without a connection to the source)
- SDO is integrated with XML
- The key components of the SDO framework are data objects, data graphs, and data object metadata

Component	Description
Data object	Fundamental data structure for representing business data
Data graph	A container for a hierarchical set (a tree) of data objects
Data object metadata	Metadata is the schema definition of the object; it contains information about the data in the data object (property types, relationships, and constraints)

© Copyright IBM Corporation 2016

Figure 7-4. Introduction to Service Data Objects (SDO)

WB8601.2

Notes:

It is not uncommon for an enterprise application developer to know several data access technologies, such as JDBC, XML, JMS, web services, and enterprise information systems. Unfortunately, developers are then required to become experts in many different data access technologies.

The goal of SDO is to provide a programming model that unifies data representation across heterogeneous data sources, and to simplify application development for developers and tools providers.

SDO provides a common API that can be used regardless of the data store that is being accessed. This common way of representing data also makes SDO an ideal choice for data abstraction in a service-oriented architecture.

Additionally, built into the SDO architecture is support for some common programming patterns. SDO supports a disconnected programming model. Typically a client might be disconnected from a particular data access service (DAS) while working with a set of business data. However, when the client completes processing, and must apply changes to a data store by way of a DAS, a change summary provides the appropriate level of data concurrency control. This change summary information is built into the SDO programming model.

Another important design point to note is that SDO integrates well with XML. As a result, SDO naturally fits in with distributed service-oriented applications.

Finally, SDO is designed to support both dynamic and static data access APIs. The dynamic APIs are provided with the SDO object model and provide an interface that allows developers to access data even when the schema of the data is not known until run time. In contrast, the static data APIs are used when the data schema is known at development time, and the developer prefers to work with “strongly typed” data access APIs.

SDO is a programming model that IBM and BEA proposed jointly as JSR 235. Since that time, SDO architects have published and revised specifications, which are available at the following websites:

- <http://oasis-open.org/sca>
- <http://www.jcp.org>

Data objects

- Data objects are the fundamental structures for representing business data
- A data object holds data as a set of properties
- Each data object provides read and write access to properties through:
 - Getters and setters
 - XPath (XML Path Language)
- Properties can be:
 - Primitive data types (such as strings)
 - Commonly used data types (such as dates)
 - Multivalued fields (such as arrays)
 - Other data objects
- In memory, data objects are represented as instances of `commonj.sdo.DataObject`
 - Objects are serialized to XML

© Copyright IBM Corporation 2016

Figure 7-5. Data objects

WB8601.2

Notes:

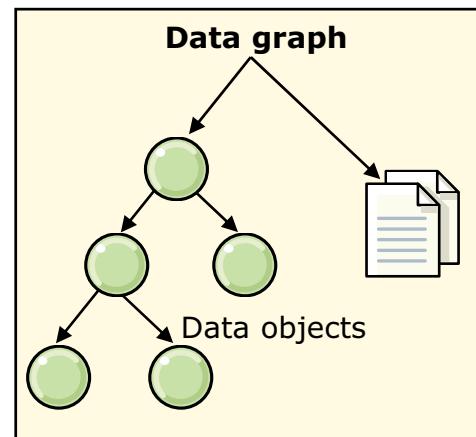
The fundamental concept in the SDO architecture is the data object. It is not uncommon to see the term “SDO” used interchangeably with “data object.” A *data object* is a data structure that holds primitive data, multivalued fields (other data objects), or both. The data object also holds references to metadata that provides information about the data included in the data object. In the SDO programming model, the `commonj.sdo.DataObject` Java interface definition represents data objects. This interface includes method definitions that allow clients to get and set the properties that are associated with the **DataObject**. As an example, consider modeling customer data with an SDO data object. The properties that are associated with the customer might be:

firstName(String), **lastName(String)**, and **customerID(long)**. This sample shows how you would use the **DataObject** API to get and set properties for the customer data object:

```
DataObject customer = ...
customer.setString("firstName", "John");
customer.setString("lastName", "Doe");
customer.setInt("customerID", 123);
int id = customer.getInt("customerID");
```

Data graphs

- Data graphs are an optional wrapper around a root data object and associated data objects (in a tree structure)
 - Can include data objects from different data sources
- Data graphs include a change summary that records modifications to the data tree
 - Object changes: Reference to the object whose properties are changed, the properties that changed, the new value, and the old value
 - Object creations: Data objects that are added to the data graph
 - Object deletions: Data objects that are removed from the data graph



© Copyright IBM Corporation 2016

Figure 7-6. Data graphs

WB8601.2

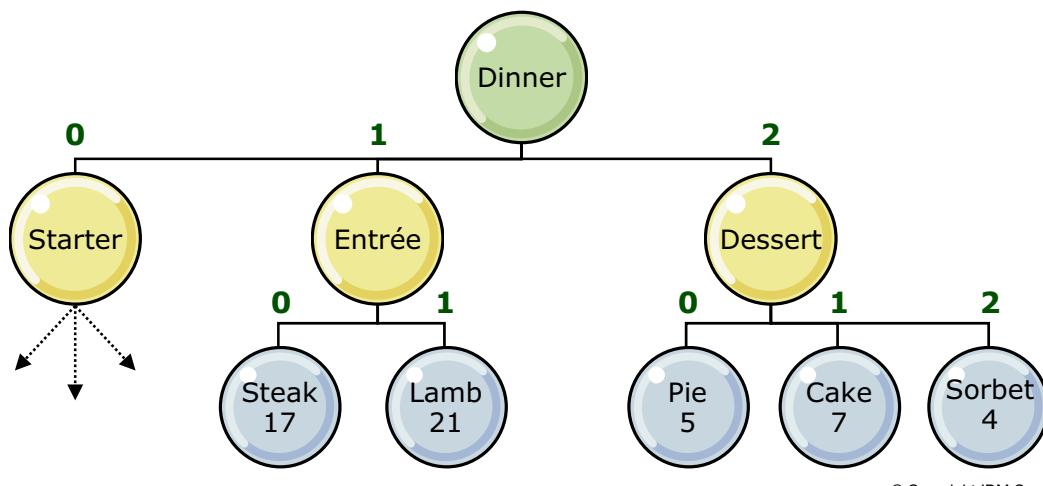
Notes:

Another important concept in the SDO architecture is the *data graph*. A data graph is a structure that encapsulates a set of data objects. From the top-level data object contained in the graph, all other data objects are reachable by traversing the references from the root data object.

Another important feature included in the data graph is a change summary that is used to log information about the data objects in the graph that changed during processing. In the SDO programming model, the `commonj.sdo.DataGraph` Java interface definition represents data graphs. In addition, the `commonj.sdo.ChangeSummary` interface defines the change summary information. A complete object model for SDO V1.0 is included in the specification document.

Example: SDO data tree

- This data tree represents a dinner menu at a restaurant
 - The root object is a “menu type” with a value of Dinner
 - The related child objects are “course type” (Starter, Entrée, and Dessert) and “dish type” (Steak: \$17, and Lamb: \$21)
- Clients traverse the tree beginning at the root, and access the child objects by their index number: 0, 1, 2...n



© Copyright IBM Corporation 2016

Figure 7-7. Example: SDO data tree

WB8601.2

Notes:

7.2. Introduction to business objects

Introduction to business objects



© Copyright IBM Corporation 2016

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

10.1

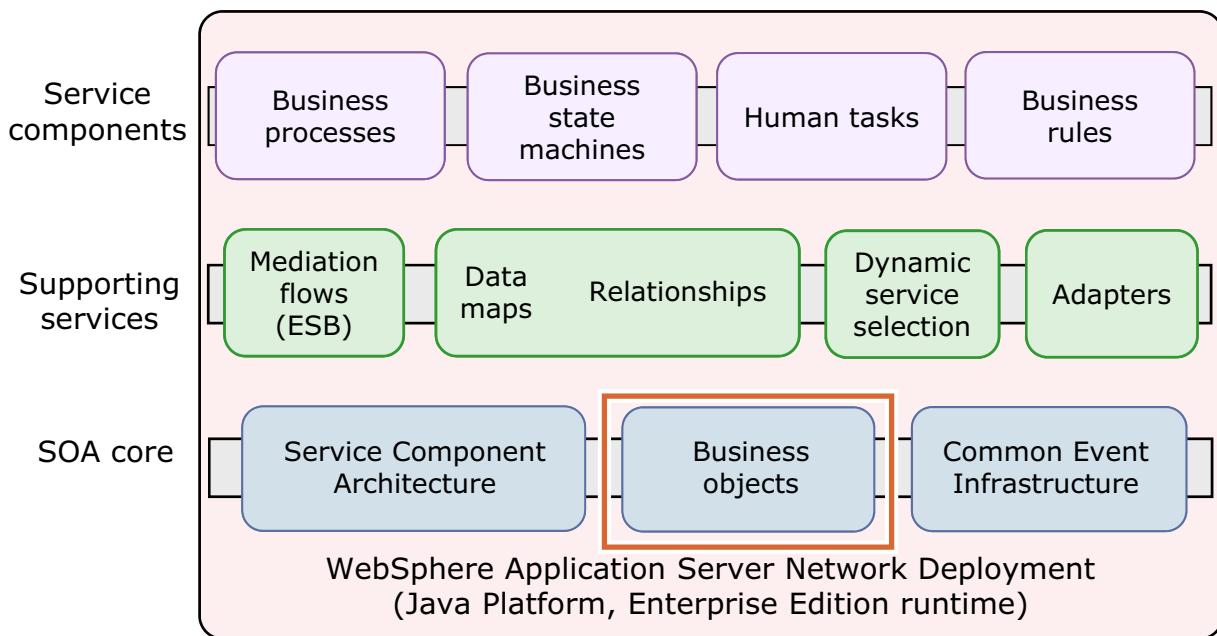
Figure 7-8. Introduction to business objects

WB8601.2

Notes:

Business objects are an SOA core component

- Business objects are an SOA core component
- Business objects provide an abstraction layer for data objects



© Copyright IBM Corporation 2016

Figure 7-9. Business objects are an SOA core component

WB8601.2

Notes:

Business objects, which are extensions of Service Data Objects (SDO), provide an abstraction layer for data objects and are part of the SOA core components. Each of the SOA core components is based on an open standard: SCA, SDO, and Common Event Infrastructure. Business objects are a composite of the SDO specification and IBM extensions.

Introduction to the business object framework

- The business object framework is intended to provide a data abstraction for the Service Component Architecture (SCA)
 - The business object framework and common API are designed to mitigate complexities of working with disparate business data in SOA
 - The business object framework provides some additional business integration functions that are not found in SDO
- Business objects represent the data that flows between SCA components in a service-oriented architecture
- Component interfaces use business objects as inputs and outputs

© Copyright IBM Corporation 2016

Figure 7-10. Introduction to the business object framework

WB8601.2

Notes:

Server products such as IBM Business Process Manager implement the SDO specification by using business objects. SCA components exchange data by passing business objects to each other. Business objects (SDOs) are thus the primary data abstraction for the Service Component Architecture (SCA).

The business object framework in IBM Process Server is based on SDO technology. However, the SDO specification does not include business integration capabilities such as “verbs” or “event summaries.” WebSphere Adapters and earlier versions of IBM Process Server require these capabilities. The event summary and verb are now deprecated. The added features that are provided are not part of the core SDO specification.

Business objects that are represented as SDOs

- All business objects are implemented as SDOs

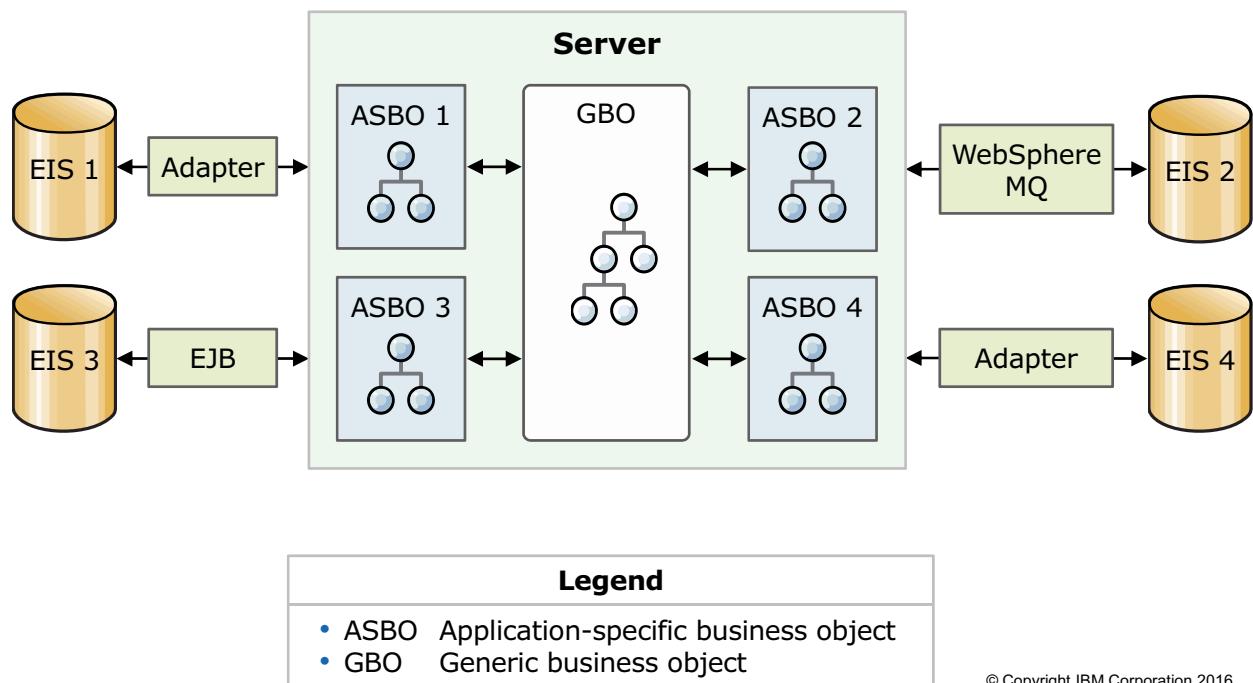


Figure 7-11. Business objects that are represented as SDOs

WB8601.2

Notes:

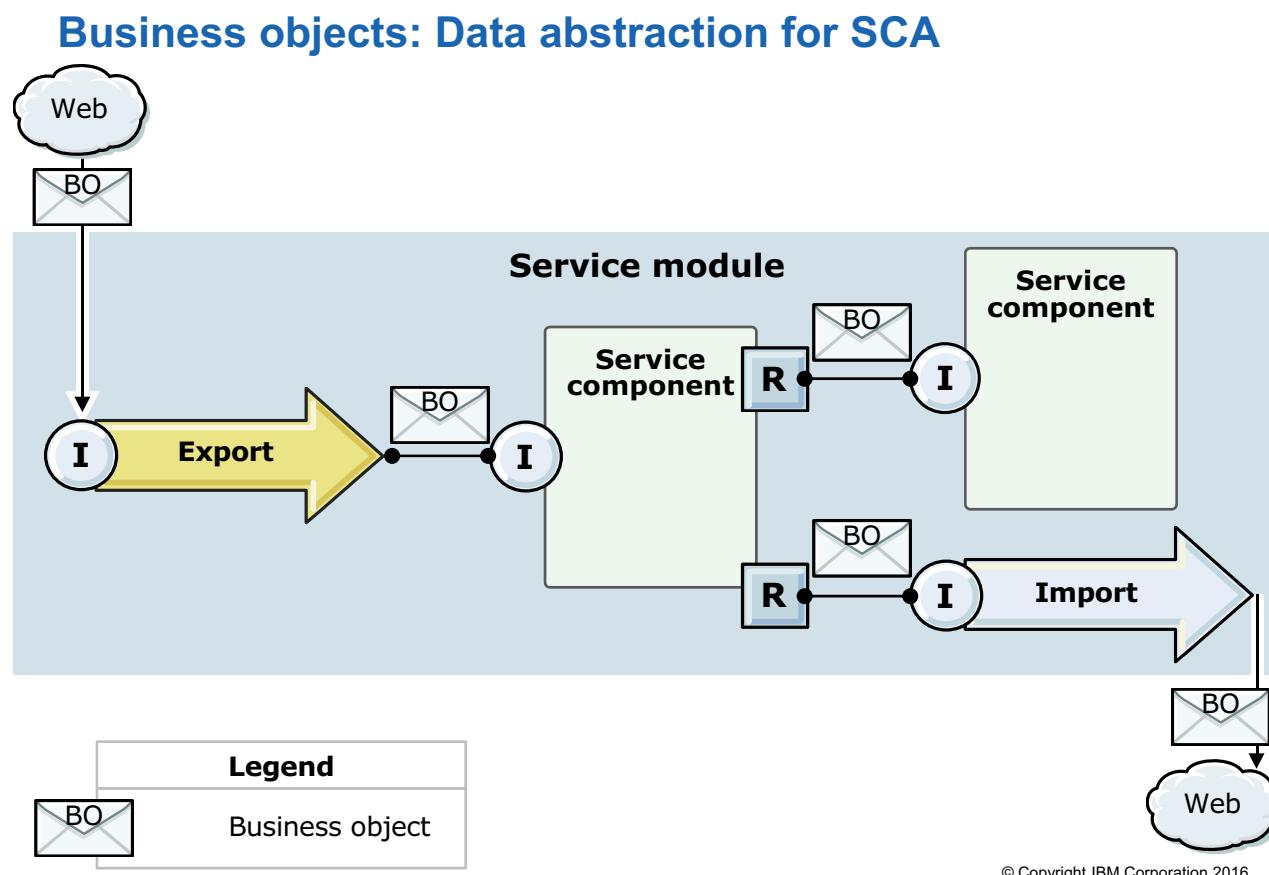


Figure 7-12. Business objects: Data abstraction for SCA

WB8601.2

Notes:

Business objects (SDOs) are the primary data abstraction for the Service Component Architecture (SCA).

This diagram illustrates how SCA provides the framework to define service components and compose these services into integrated applications. It further shows that business objects represent the data that flows between services. Whether the interface that is associated with a particular service component is defined as a Java interface or a WSDL portType, the input and output parameters are represented by using business objects. The data abstraction for SCA is achieved through business objects (SDO), and the SCA programming model is used to manage invocation.

Composition of business object framework

- The business object framework is composed of the following components:

Component	Description
Business object (BO)	Fundamental data structure for representing business data
Business graph (BG)	Optional wrapper for a business object or hierarchy of business objects to provide enhanced information such as: <ul style="list-style-type: none"> Change summary Event summary (IBM extension to SDO): Deprecated Verb (IBM extension to SDO): Deprecated
Business object type metadata	<ul style="list-style-type: none"> Annotate business objects with application-specific information
Business object services	<ul style="list-style-type: none"> A set of services that facilitate working with business objects These services are in addition to the capabilities provided in SDO

© Copyright IBM Corporation 2016

Figure 7-13. Composition of business object framework

WB8601.2

Notes:

The table on this page introduces several key concepts that make up the business object framework. The intent is to provide a high-level picture of the major pieces that compose the entire framework before considering the details of each concept.

The business object is the fundamental data structure that is used to represent business data. It is not uncommon to use the term *business object* to refer to the entire framework. However, in this presentation when the term *business object* is used, it refers to the fundamental data structure for representing business data and not to the overall architecture.

The business graph is used in the business object framework to wrap a top-level business object and to provide more information that is used to enhance the data. In particular, the data graph includes the change summary for the data in the graph (similar to the SDO change summary information), event summary, and verb information (used for data synchronization between EIS systems). The business graph is similar to the concept of the data graph in the SDO architecture. However, notice that the event summary and the verb portion of the enhanced information are not included with the SDO data graph concept. The event summary and verb are described in more detail. The two extensions to the SDO framework are deprecated.

Verbs were previously used to communicate the type of after-image (enterprise information system) event that was being moved through the business integration runtime environment. A verb provided information to adapters about what type of action to take on a business object in the EIS to which it was connected. If the verb was “create”, for example, then the change summary would be empty (no change is needed). The most common verbs that adapters supported were:

- Create (create a record in the EIS)
- Retrieve (retrieve a record from the EIS)
- Update (update a record in the EIS)
- Delete (delete a record from the EIS)
- Update with delete (update one record in the EIS while deleting another related record)

However, the list of supported verbs can be restricted, or users can specify more verbs. Verbs can be extended beyond the standard set, or they can be constrained to a particular set of values.

Typically, adapter developers are the ones who used the event summary in previous releases. The event summary had two purposes:

- To carry the ObjectEventID
- To provide an extensible markup metadata mechanism for DataObjects at run time

ObjectEventID is the mechanism that is used to uniquely identify an instance of an object that is displayed in the runtime environment. In previous versions, this information was appended to each business object that was designed top-down. In the business object specification, this information must be separated from the original business object to remain non-intrusive. Thus, in the business object framework, this information was carried in the event summary, where the unique identifier is associated with a DataObject in the business object hierarchy of the business graph.

Event information can also be carried in the event summary. This summary is a string that can be used to add metadata that is associated with each object in the business object hierarchy of the business graph. One potentially useful model for event information in the event summary is to mark up contained business objects with a verb other than the standard create, update, and delete verbs. The event summary is also used to add metadata to business objects at run time. For example, you can use the event summary to carry a verb other than the standard verbs.

The business object type metadata is available to annotate business objects with application-specific information.

Business object services are a set of services that facilitate working with business objects. These services are available in addition to the capabilities that are already provided by SDO and are needed to provide function where the initial version of SDO did not specify a solution. In the revised SDO specification, these types of services are included where applicable. The business object services are a set of business object service APIs provided to enhance existing SDO capabilities, including business object create, copy, compare, and serialize. They also include services to access enhanced business graph capabilities: event summary and change summary.

Business objects (1 of 2)

- Business objects are the primary data structure for business data and data types that are defined in WSDL (interface) definitions
- Business objects are modeled by using XML schemas (XSD)
 - Can import business object schema definitions from, or export to, other systems
 - Support for the full XML schema data type system
- At run time, `commonj.sdo.DataObject` is used to represent business objects in memory as an SDO instance
 - Created from XSD files by using the business object factory
 - Accessible by using the SDO API and XPath
- Support is provided for data object schemas from industry standards organizations
 - HL7, ACORD, and OAGIS
 - IBM Business Process Manager Industry Packs provide prebuilt data objects

© Copyright IBM Corporation 2016

Figure 7-14. Business objects (1 of 2)

WB8601.2

Notes:

A business object is the primary structure for representing business data in the IBM Process Server runtime environment. This structure also includes document literal message definitions as would be found in a WSDL definition when defining included data types. The business object relates to the `DataObject` construct in SDO. In fact, business objects are represented in memory as an SDO `commonj.sdo.DataObject`. Therefore, if you are doing development work that involves programmatically working with business objects, it is important to become familiar with the SDO APIs. Also, you might occasionally see a business object that is referred to as an SDO because the SDO `DataObject` is used to represent a business object in the client programming model.

Currently, the only model for modeling or defining business objects is XML schema (XSD). The business object framework supports the full XML schema type system and facet capabilities. For this reason, business object definitions that third-party systems create can be successfully imported and used in an IBM Process Server application module. Development teams that might use a model other than XML schema to define business objects must convert these business objects to XML schema for use in IBM Process Server.

IBM Integration Designer support for industry standard organizations extends beyond the ones that are mentioned here (HL7: Health Level Seven, ACORD: Association for Cooperative Operations

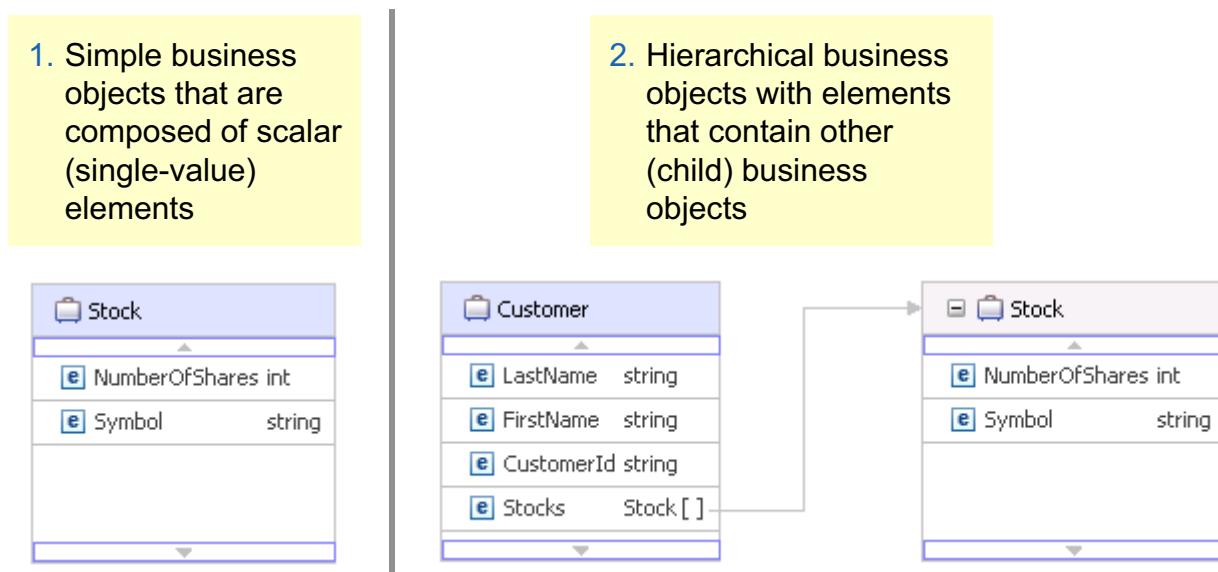
Research and Development, OAGIS: Open Applications Group Integration Specification). For a complete list of supported industry schemas, consult the IBM Integration Designer product documentation.

Corresponding to the addition of support for industry standard schemas, IBM Integration Designer also gives you the ability to exclude libraries when cleaning a project. This feature is especially useful for artifacts that are stable, such as industry schemas. When you use the **Project > Clean** menu item to invoke a full build of projects in the workspace, the resources in all of the libraries are automatically revalidated. If you have libraries that contain large XML schemas or WSDL files, the process of revalidating the library resources can considerably add to the time required for the build to complete. However, if you have one or more libraries that contain large files and you do not expect the libraries to change, you can exclude them from the builds and then reduce the overall build time.

Working with industry standard schemas often means working with many business objects. In IBM Integration Designer, you can enter a filter in the parent category so that you can quickly search and find an artifact that you want. For example, if you have many business objects in the data types category, you can use inline filtering to quickly locate a specific business object. You can also mark artifacts as favorites, which do not get filtered, so you can tailor the Business Integration view to show only particular artifacts. These features are especially useful when working with industry schemas.

Business objects (2 of 2)

- Business objects are collections of elements with names and data types
- Two types of business objects:



© Copyright IBM Corporation 2016

Figure 7-15. Business objects (2 of 2)

WB8601.2

Notes:

Business objects have two types. In the simplest case, a business object can be constructed of only scalar properties as is shown in the Stock business object. In addition, business objects can be defined to be hierarchical (composed of one or more properties that reference a nested business object definition).

The Customer business object that is shown on this slide is included to illustrate an example of a hierarchical business object. In this example, each customer is associated with a collection of stocks. The data type for the Stocks property is the Stock business object, represented as an array.

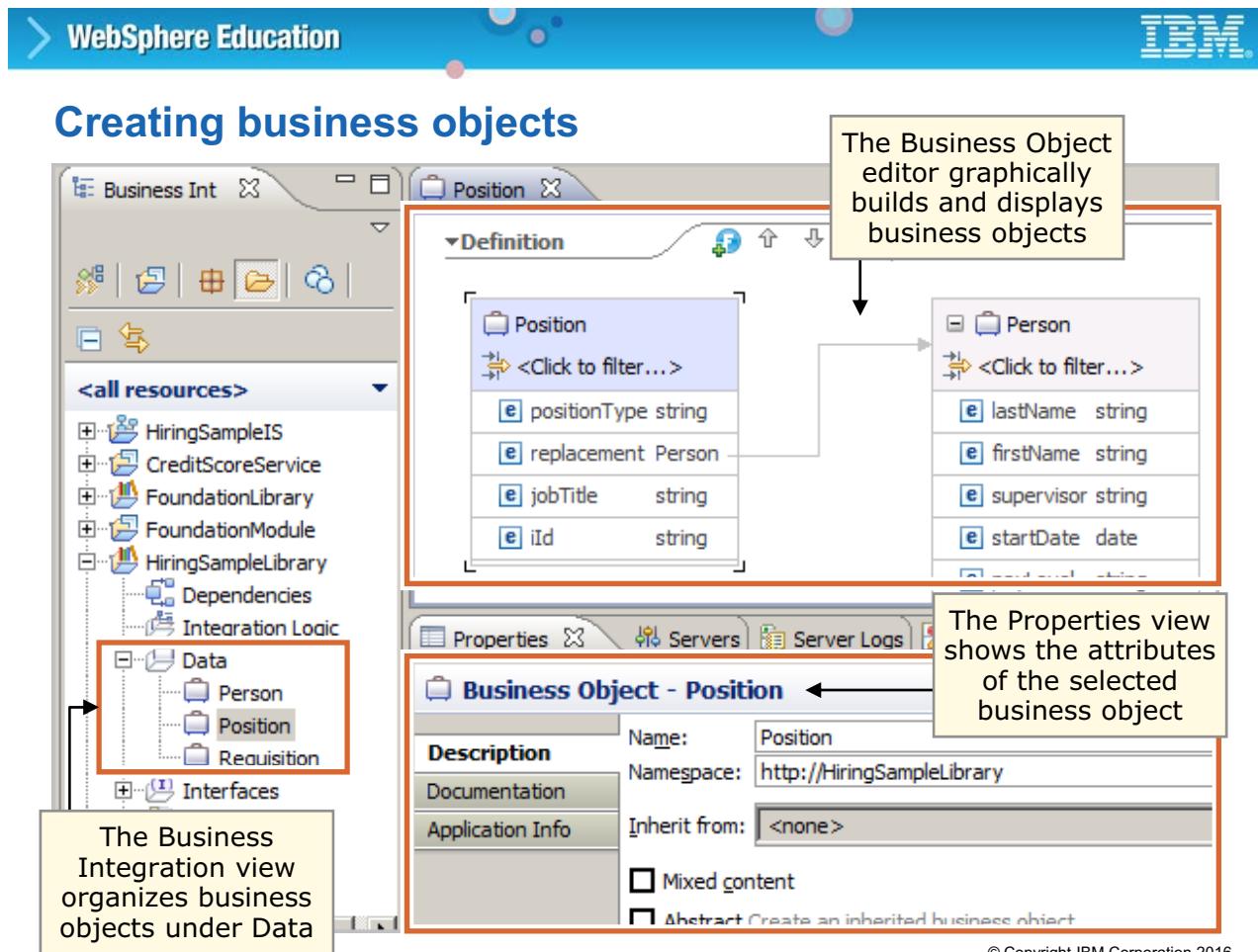


Figure 7-16. Creating business objects

WB8601.2

Notes:

The Business Integration perspective has a Business Integration view, which provides a logical view of the key resources in each module and library. Within each module or library, the resources are categorized by type. Logical resources that are shown in the navigation tree do not necessarily have a one-to-one mapping to files. Artifacts that are not necessary for the development of integration applications are not shown in this navigation tree. Business objects are listed under the library data types section. When you open a resource from the Business Integration view with an editor, the resource is displayed in the editor area. The diagrams for the business object editor are composed on the canvas of the editor. When you use the Properties view with editors, you can modify properties of selected elements in the editor.

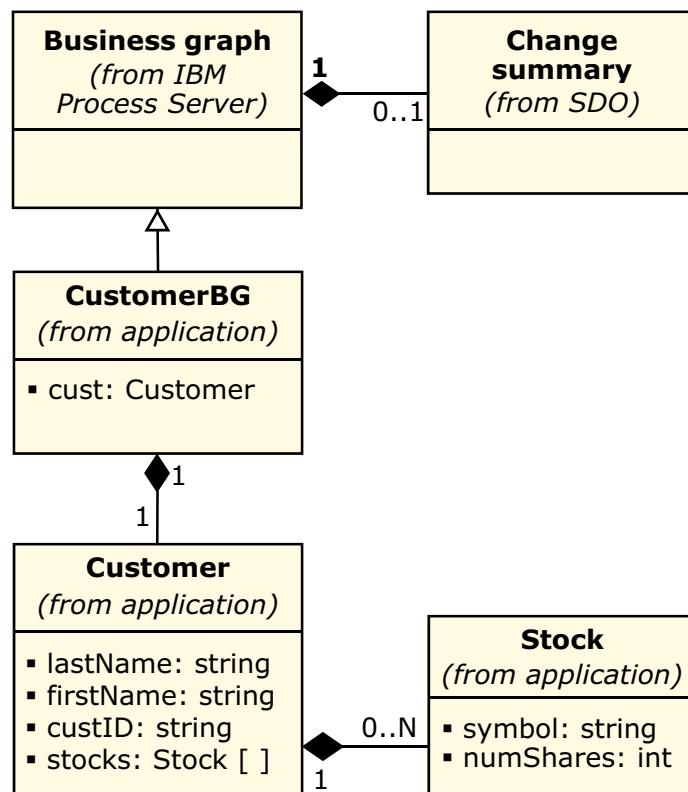
To define new business objects by using the business object editor, complete the following steps:

1. Expand the new module that is located inside the Business Integration section of the Integration Designer window.
2. Right-click the Data Types folder and select **New > Business Object**.
3. Type a new Name in the Business Object window. For example, type `Customer` to create a customer business object.

4. Click **Finish**. The new business object is added to the Data Types folder.
5. Click the “Add a field to a business object” icon and add the necessary fields to the business object.
6. Click the Save icon.
7. Repeat the previous steps for each business object that you want to create.

Business graphs

- A business graph is an optional container around a business object or a hierarchy of business objects
- Business graphs are SDO data graphs with extensions
- Business graphs include:
 - Root and associated business objects (in a tree)
 - A change summary header
- Graphs are seldom used outside the adapters (some adapters require graphs for the change summary)



© Copyright IBM Corporation 2016

Figure 7-17. Business graphs

WB8601.2

Notes:

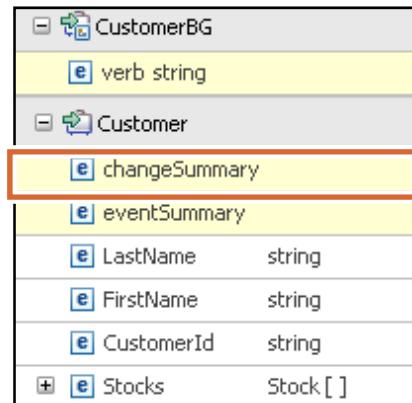
This slide shows a view of the programmatic design of business objects. The class diagram shows the relationship between SDOs and BOs, and the relationship between business graphs and business objects.

A business graph is a concept that is related to the SDO data graph concept. In the IBM Process Server runtime environment, a business graph is used as a container to wrap a top-level business object and provides several important capabilities that are related to the contained business object hierarchy.

Business graphs are available in the runtime to provide supporting metadata to adapters that are connected to back-end systems. Depending on the adapter, a business graph might not be required to indicate to the back-end system what this data represents, or what action should be taken regarding it. Business graphs are optional and are required only when you are adding business objects to a module created with an older version. If business graphs exist, they are processed, but the verb is ignored.

Business graph: Change summary

- The change summary is used to record changes to contained business objects
- Change summaries support disconnected data patterns and command-event models
 - Disconnected data pattern:** Records changes to business objects while disconnected from the data source
 - Command event model:** Allows an EIS to capture and publish data changes for use by other systems that must be aware of these events
- Two types of change summary usage:
 - Implicit:** Change logging is turned on, and any changes to contained business objects are logged
 - Explicit:** Modify the change summary explicitly (IBM extension)



© Copyright IBM Corporation 2016

Figure 7-18. Business graph: Change summary

WB8601.2

Notes:

The Customer business object in the figure needs the enhancements that a business graph provides. To produce these enhancements, you would create a business graph called CustomerBG. The **CustomerBG** is defined by using XML schema and can be thought of as a specialized type of business object that includes the verb for the business graph and a single business object of the **Customer**.

Change summaries support two use cases. The first use case is a disconnected use model where it is necessary to be able to capture changes that are made to the contained business object between processes. For example, consider a scenario where a service (**service1**) constructs a business graph that is based on a query to a data store and then passes that business graph to another service (**service2**) for processing. The second service can change the business objects that are contained in the business graph and then later call the first service to apply those changes to the data store. In this case, change summary information is necessary to assure the appropriate level of data concurrency.

Another important use case is the situation where the information that is contained with the business graph is used to capture and publish data changes to the business integration runtime environment from an EIS system. In this case, the change summary, event summary, and verb that

is associated with the business graph are all used to communicate the appropriate information that is needed for data synchronization among different EIS systems.

Some more notes on business graphs are as follows:

- Business graphs are used for enrichment of only top-level business objects. However, a contained business object in one scenario can still be a top-level business object in another scenario and a candidate for enrichment with a business graph.
- A business graph is represented as a `commonj.sdo.DataObject` in memory, but the event header and the change summary header cannot be accessed through the normal SDO APIs that are used to access contained DataObject properties.
- The change summary that is associated with a business graph can be used in two ways. The first way is to make implicit change summary updates. These updates are made by turning on change summary logging through the `commonj.sdo.ChangeSummary` interface. Doing so begins tracking changes that are made whenever any of the DataObject APIs are called for the business objects that are contained in the business graph.
- The other type of change summary tracking can be made explicitly by using the business object service that allows the change summary to be updated directly. This service must be offered as a business object service because this capability is not available with the base SDO change summary function, and it is needed to support the command-event model. An example of explicit change summary usage is in the course appendixes.

Data objects in IBM Business Process Manager

IBM Integration Designer

- Supports modeling and developing integrated data objects
- Business objects follow the SDO framework
- Business objects are modeled by using the XSD standard
- Runtime objects are instances of `commonj.sdo.DataObject`

IBM Process Designer

- Supports modeling data objects
- SDO objects are not supported
- Business graphs are not supported, so:
 - No change summaries exist
 - Disconnected data model is not supported
 - Command event model is not supported
- Data objects are stored as XML BLOBs in the IBM Process Center repository
- Data objects become instances of JavaScript objects at run time
- Data objects might be exported as XML for use in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 7-19. Data objects in IBM Business Process Manager

WB8601.2

Notes:

Process Designer and Integration Designer provide developers with one common business object model for representing different kinds of business entities from different domains.

In Process Designer, business objects are focused on a data type representation. Base business objects (variable types) are provided in system toolkits, and you can create custom variable types called custom business objects. In Integration Designer, which is available only with IBM Business Process Manager Advanced, business objects can represent more complex XSD constructs. In Integration Designer, business objects have a close affinity with XML schemas.

At development time in Integration Designer, the business object model enables developers to define business objects as XML schema definitions. At run time, the business data that is defined in the XML schema definitions is represented as Java business objects. In this model, business objects are loosely based on early drafts of the Service Data Object (SDO) specification and provide the complete set of programming model application interfaces required to manipulate business data.



Unit summary

- Identify the purpose and advantages of using the SDO framework
- Define the data object and data graph components of the SDO framework
- Identify the purpose and advantages of using the business object framework
- Define the business object and business graph components of the business object framework

© Copyright IBM Corporation 2016

Figure 7-20. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. What is the purpose of the business object framework?
2. Name two ways in which the business object framework differs from the Service Data Object framework.
3. What is the purpose of a change summary?
4. How are business objects represented in memory at run time?
5. What is the purpose of the “business object compare” utility?

© Copyright IBM Corporation 2016

Figure 7-21. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

Checkpoint answers

1. Service Data Objects provide a framework for application data and unify the representation of data from multiple sources by using a single API for data access: the SDO API.
2. The business object framework supports event summaries, verbs, and explicit manipulation of the change summary. The SDO framework does not.
3. The change summary is used to record changes to contained business objects.
4. At run time, `commonj.sdo.DataObject` is used to represent business objects in memory as an SDO instance.
5. It shows the structure of business objects in two different modules or libraries, and you can visually determine additions, removals, and changes. Also, with the “copy report to clipboard” option, you can paste changes into a spreadsheet, document, or other report.

© Copyright IBM Corporation 2016

Figure 7-22. Checkpoint answers

WB8601.2

Notes:

Supplemental note: Business object framework runtime environment

- The business object framework runtime is:
 - Built on a high-speed XML infrastructure that enables optimized XML processing and enhanced XML fidelity
 - An internal engine that is visible to process developers by using the business object, XPath, and XSLT programming models available in the IBM Process Server SCA components
- This table describes how data in the runtime is modeled and integrated

Abstraction	Implementation	Description
Instance data	Business object	Primary mechanism for representing business objects
Instance metadata	Business graph	Wrappers around simple business objects that carry a change summary
Type metadata	Enterprise or business object type metadata	Might be added to business object definitions to enhance their runtime value
Services	Business object services (APIs)	Set of capabilities that are provided on top of the basic SDO capabilities

© Copyright IBM Corporation 2016

Figure 7-23. Supplemental note: Business object framework runtime environment

WB8601.2

Notes:

Exercise 5



Creating business objects and shared interfaces

© Copyright IBM Corporation 2016

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

10.1

Figure 7-24. Exercise 5

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Compare business objects between IBM Process Designer and IBM Integration Designer
- Implement business objects and define their elements
- Implement interfaces that use business objects as inputs and outputs

© Copyright IBM Corporation 2016

Figure 7-25. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 5

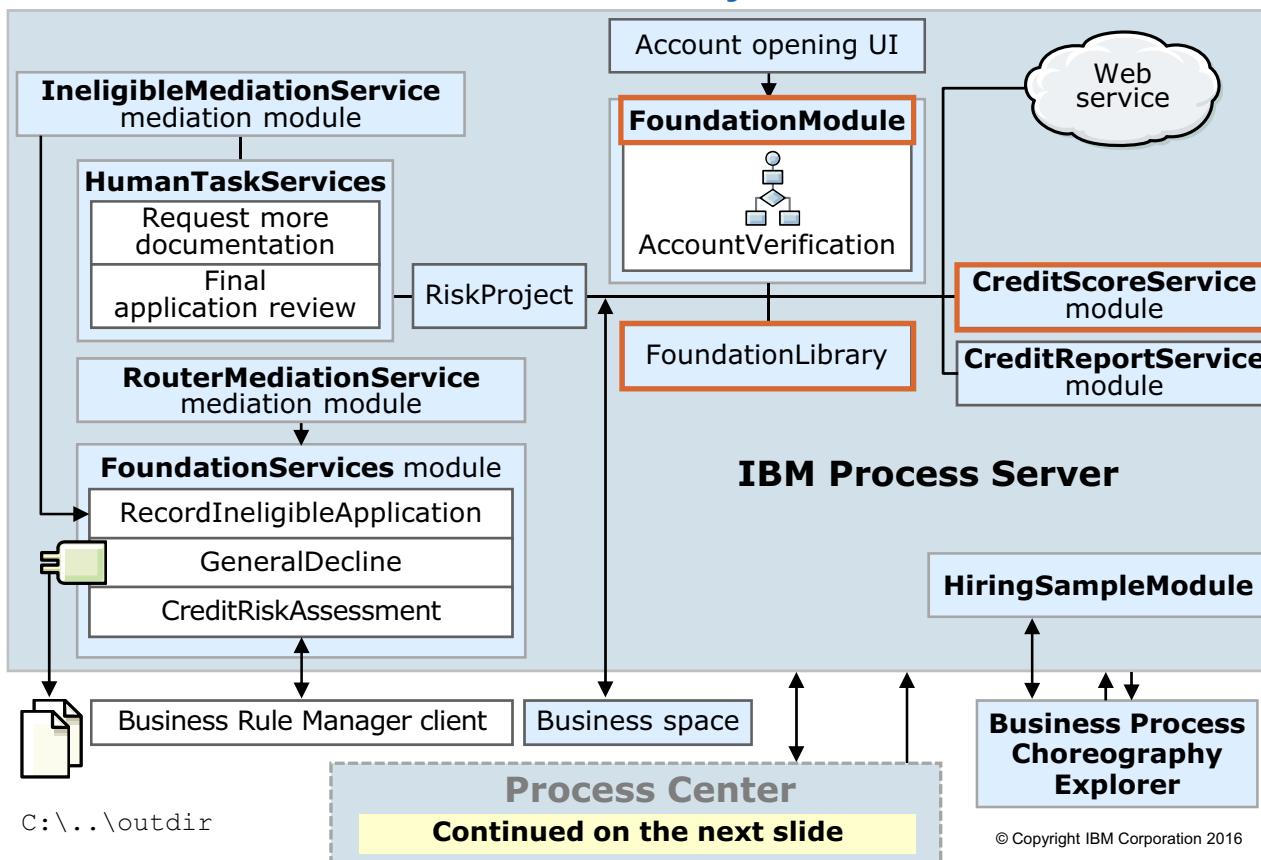


Figure 7-26. Course business scenario: What you work with in Exercise 5

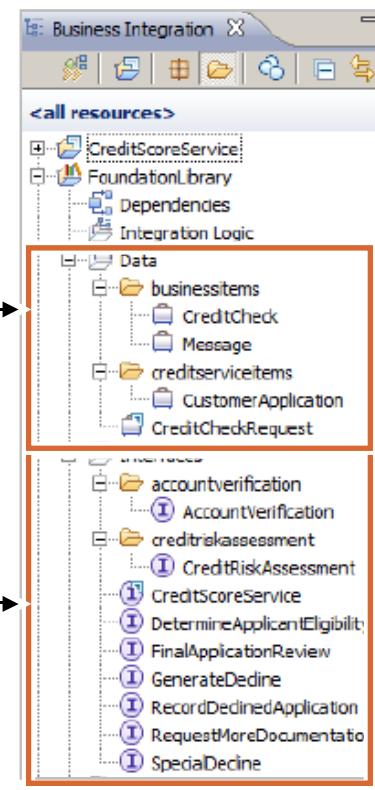
WB8601.2

Notes:

Components that are required for Exercise 5 (1 of 2)

Prebuilt components that are imported in the lab:

1. **FoundationModule** module
2. **CreditScoreService** module
3. **FoundationLibrary** container for **business objects**:
 - CreditCheck
 - Message
 - CustomerApplication
 - CreditCheckRequest
4. **FoundationLibrary** container for **interfaces**:
 - AccountVerification
 - CreditRiskAssessment
 - CreditScoreService
 - DetermineApplicantEligibility
 - FinalApplicationReview
 - GenerateDecline
 - RecordDeclineApplication
 - RequestMoreDocumentation
 - SpecialDecline



© Copyright IBM Corporation 2016

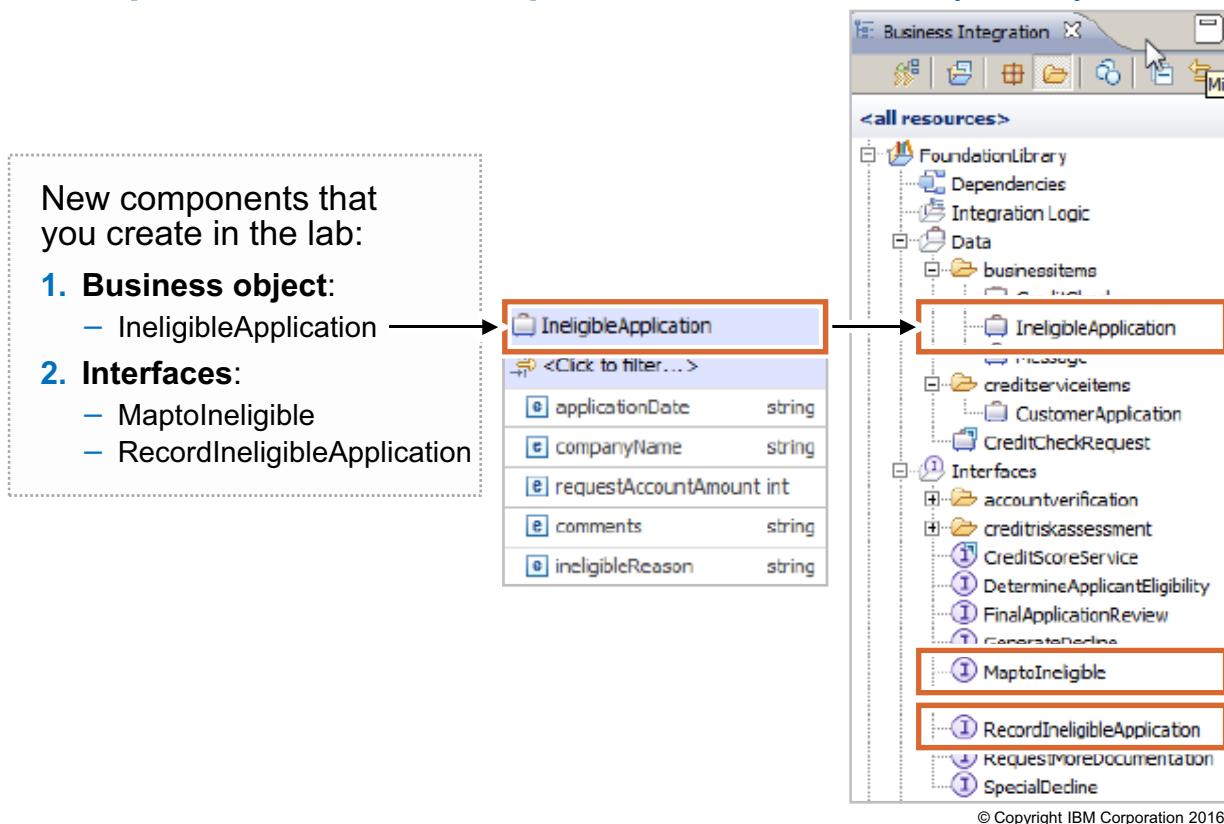
Figure 7-27. Components that are required for Exercise 5 (1 of 2)

WB8601.2

Notes:

WebSphere Education

Components that are required for Exercise 5 (2 of 2)



© Copyright IBM Corporation 2016

Figure 7-28. Components that are required for Exercise 5 (2 of 2)

WB8601.2

Notes:



References

- OSOA: Service Component Architecture family of specifications
 - <http://oasis-open.org/sca>
- developerWorks topics about Service Data Objects (SDO)
 - <http://www.ibm.com/developerworks/java/library/j-sdo/>
- Service Data Objects
 - <http://www.oasis-open.org/sdo>
- XML schema
 - <http://www.w3.org/XML/Schema>

© Copyright IBM Corporation 2016

Figure 7-29. References

WB8601.2

Notes:

Unit 8. Business process choreography overview

What this unit is about

This unit introduces you to Business Process Execution Language (BPEL) and business process development that uses Business Process Choreographer.

What you should be able to do

After completing this unit, you should be able to:

- Describe the purpose and business value of using the WS-BPEL standard
- Describe the function of the business process container
- Describe the difference between long-running and microflow (short-running) business processes
- List and describe the seven parts of a business process

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Describe the purpose and business value of using the WS-BPEL standard
- Describe the function of the business process container
- Describe the difference between long-running and microflow (short-running) business processes
- List and describe the seven parts of a business process

© Copyright IBM Corporation 2016

Figure 8-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to Web Services Business Process Execution Language (WS-BPEL)
- Elements of BPEL processes

© Copyright IBM Corporation 2016

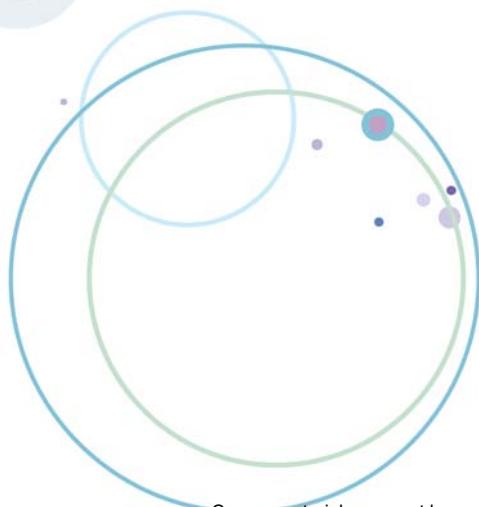
Figure 8-2. Topics

WB8601.2

Notes:

8.1. Introduction to Web Services Business Process Execution Language (WS-BPEL)

Introduction to Web Services Business Process Execution Language (WS-BPEL)



© Copyright IBM Corporation 2016

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

10.1

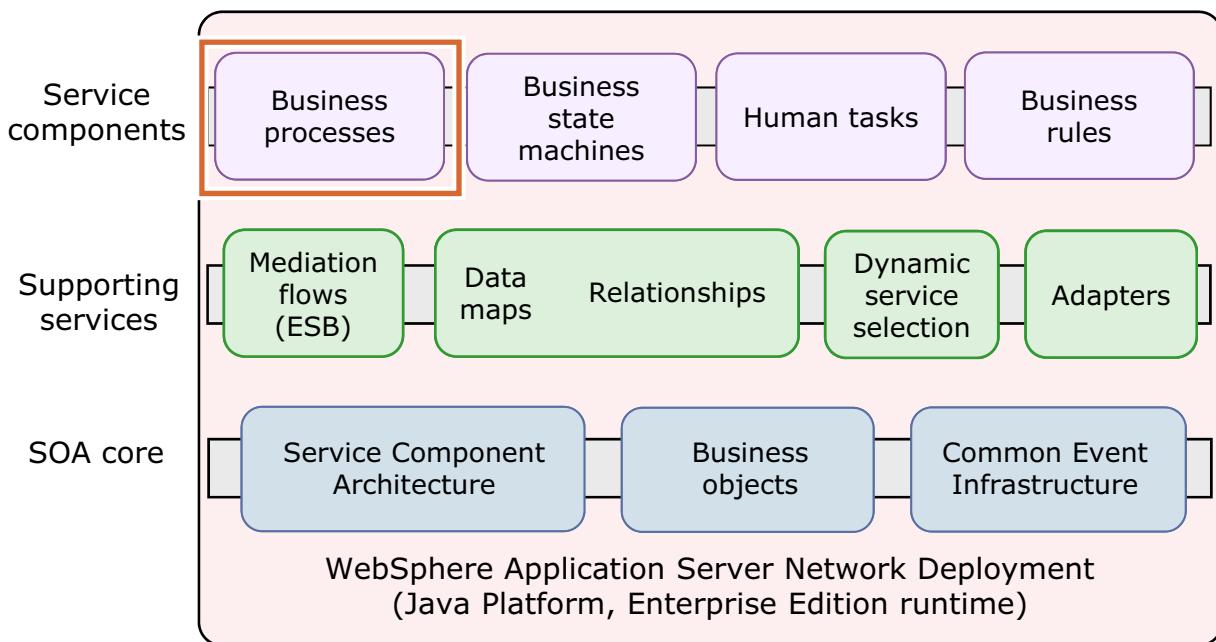
Figure 8-3. Introduction to Web Services Business Process Execution Language (WS-BPEL)

WB8601.2

Notes:

Business processes are service components

- Business processes are part of the service components layer



© Copyright IBM Corporation 2016

Figure 8-4. Business processes are service components

WB8601.2

Notes:

WS-BPEL (Web Services Business Process Execution Language) is the language for composing business processes. Processes that are created with the business state machine editor and with the business process editor are deployed as BPEL processes on IBM Process Server. Processes and state machines are part of the “service component” layer. BPEL (business process or state machine) is one of the implementation types for an SCA component.



Business processes in IBM Business Process Manager

- Business processes might be captured in IBM Process Designer or in IBM Integration Designer
- IBM Process Designer
 - Captured as business process
 - Represented as business process diagram (BPD)
 - Implementations that are captured in one process application or toolkit only
 - Limited implementation options
 - One human task client
- IBM Integration Designer
 - Captured as BPEL process
 - Represented with Business Process Execution Language (BPEL)
 - Implementation might span several modules
 - Might be shared in libraries
 - Several implementation options
 - Several human task clients

For differences between BPD and BPEL and when to use which, see Unit 2

© Copyright IBM Corporation 2016

Figure 8-5. Business processes in IBM Business Process Manager

WB8601.2

Notes:

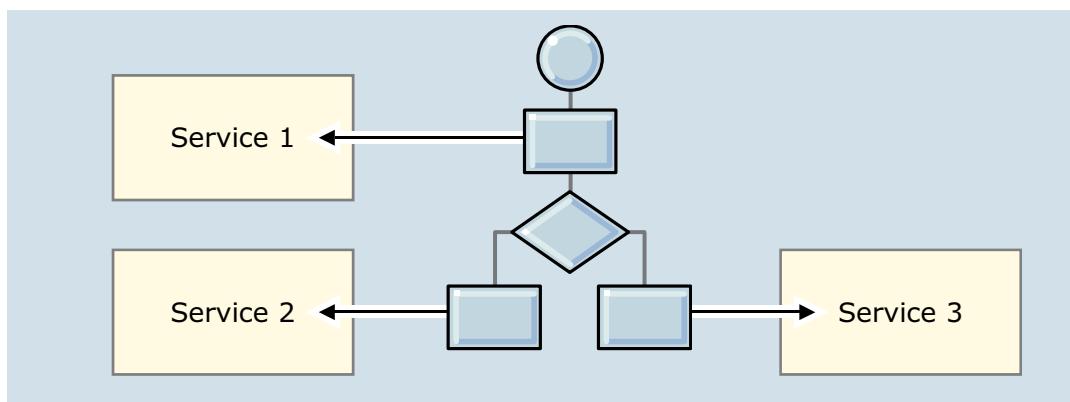
Business processes can be captured in either IBM Process Designer, IBM Integration Designer, or both. However, differences exist between the two tools:

- **IBM Process Designer**
 - Business processes are captured as business process artifacts based on Business Process Model and Notation (BPMN). They must be built in either a process application or a toolkit.
 - Business processes do not use a standard language, such as BPEL.
 - The only way to share a process is to place it in a toolkit.
 - Implementation options are limited (human tasks, JavaScript services, simple business rules, and others).
- **IBM Integration Designer**
 - Business processes are captured as BPEL artifacts. They can be built in either a module or a library.
 - They use the standard BPEL with IBM extensions (WS-BPEL).

- They are loosely based on BPMN.
- Business processes can rely on services in other modules through imports and exports, or through libraries.
- BPEL processes in modules can be exposed as other types of services through their bindings (such as web services and WebSphere MQ bindings).
- A BPEL process can have several types of implementations, for example: including the full power of Java, integration with JRules, and integration with external services.
- Several types of human task clients are available, including HTML-Dojo in Business Space, JavaServer Faces, and the BPEL Process Choreographer Explorer.

Overview: Business Process Execution Language (1 of 2)

- A business process is a flow of execution paths that are described in WS-BPEL, including:
 - Which services are invoked
 - In what order services are invoked
 - The movement of data between services
- BPEL facilitates the building of composite integration applications by allowing the reuse of existing IT assets that are exposed as services



© Copyright IBM Corporation 2016

Figure 8-6. Overview: Business Process Execution Language (1 of 2)

WB8601.2

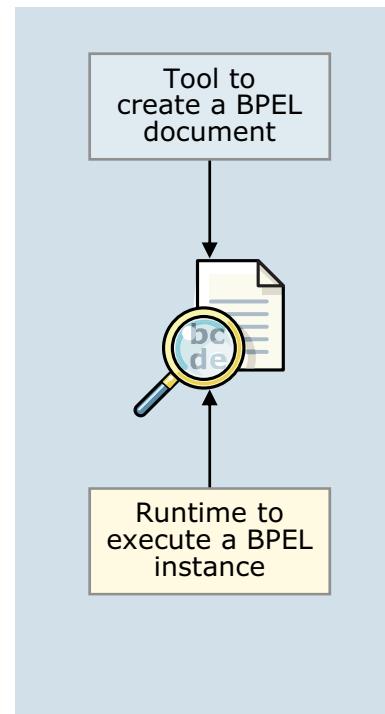
Notes:

The business process component of IBM Process Server is arguably the most important aspect of the solution. This component provides the engine and management to run the steps for each instance of a process. Process orchestration occurs within the engine, and like the conductor of an orchestra, the process engine guides all the other components.

To further illustrate, this diagram shows a set of services that provide their “black box” functions, in which only their externally visible behavior is considered and not their implementation or inner workings. To build a solution, you design a process that choreographs their execution. This process is described in the BPEL language. The core concepts that must be captured are: which services are used, in what order they are called, and what data is passed from one service to another.

Overview: Business Process Execution Language (2 of 2)

- Implementation in BPEL decouples the process from the runtime engine, making the processes vendor-neutral and technology-neutral
 - Other companies provide products with BPEL runtimes
- BPEL supports the required technology patterns for business
 - For example: Error handling, compensation, and asynchronous processing
- The BPEL-compliant business process container included with IBM Process Server manages and runs business processes
 - Runs complex business processes securely, consistently, and with transactional integrity
 - Provides high performance and quality of service
 - Provides fault tolerance and error-detection capability



© Copyright IBM Corporation 2016

Figure 8-7. Overview: Business Process Execution Language (2 of 2)

WB8601.2

Notes:

When you create a business process, you create a BPEL document that describes the choreography of your process, and you use an appropriate runtime environment to run instances of the process. You need a tool to create the BPEL description. In IBM Integration Designer, a business process editor is provided. BPEL is detailed and is described in XML. As a result, the code is not easily readable to humans. It would be difficult to compose a BPEL process by using a simple XML editor.

When you create the BPEL document, it exists independently from the tool that created it. This independence is the portability aspect of BPEL. It is not coupled to any particular choice of implementation if the vendor complies with the specification.

An OASIS technical committee manages the WS-BPEL specification (currently at V1.1 with V2.0 now available): <http://www.oasis-open.org/home/index.php>

The specification is based on WSDL and other XML standards:

- Interfaces are described by using WSDL.
- XML schema and XPath expressions define data context handling.

BPEL4WS was version 1.1 of the BPEL specification. When version 2.0 of the specification was introduced, the open source committee renamed the specification to WS-BPEL. In general, people refer to the specification as BPEL. It is an XML-based language that is used to define the flow of a business process.

The WS-BPEL standard allows business process models to be defined independent of the implementation, keeping the processes separate from the underlying infrastructure or technology. This independence fits nicely with the concept of a service-oriented architecture (SOA) where interfaces are kept separate from implementations. WS-BPEL uses services and service interfaces as a means of defining the connections between the different steps.

For example, in a business process with five steps, interfaces on the steps indicate the type of data that is passed and potentially received and the type of operation to do. The WS-BPEL standard uses other industry standards such as Web Services Description Language (WSDL) to define steps and interfaces. XML Schema Definition (XSD) is used to define data structures. The BPEL process is an XML file that is interpreted at run time to indicate the sequence of steps that make up a business process. XPath support is also provided as a primary means of working with data objects that are passed between steps.

The BPEL specification and its concepts have existed for several years. Originally started by IBM, BEA, and Microsoft, the specification is now more refined; SAP and Siebel (and others) support it as well.

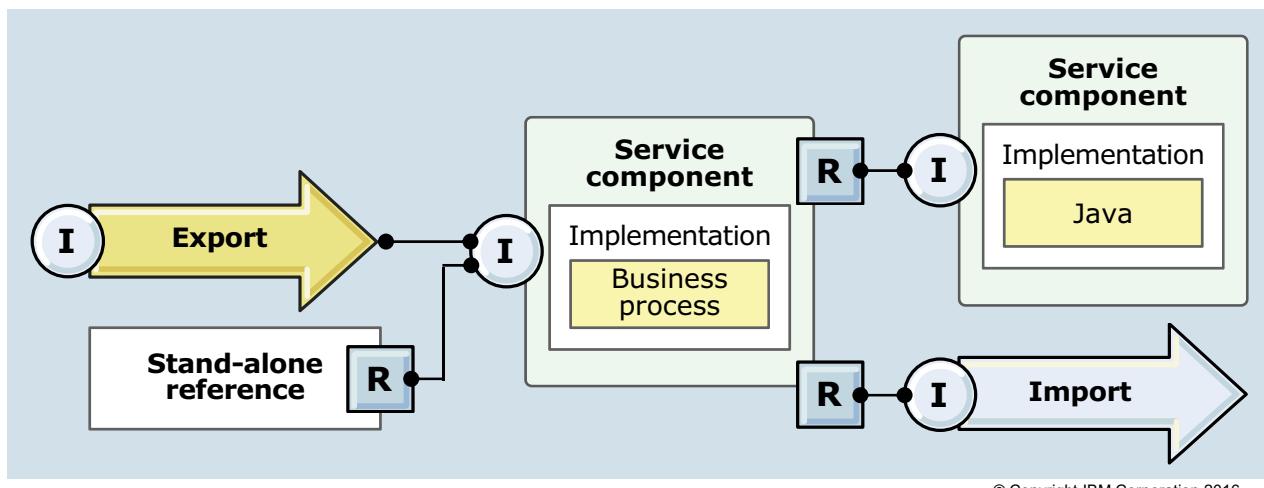
BPEL defines three separate parts or phases of a business process architecture:

- The development of a business process model
- The implementation of the activities that are involved in the business process
- The runtime engine that runs the business process model by determining the correct path through the business process and calling the necessary activities

External partners are invoked as web services. The BPEL business process is also available as a web service, so it can potentially become an activity or subprocess of a larger process.

BPEL interoperates with SCA and SDO

- BPEL processes can invoke other SCA components
- BPEL processes can be invoked as SCA components
- Partner Links are resolved to SCA components or external services
- SDO provides a standard data format for messages
 - SDO messages are sent and received
 - Business process variables use SDO



© Copyright IBM Corporation 2016

Figure 8-8. BPEL interoperates with SCA and SDO

WB8601.2

Notes:

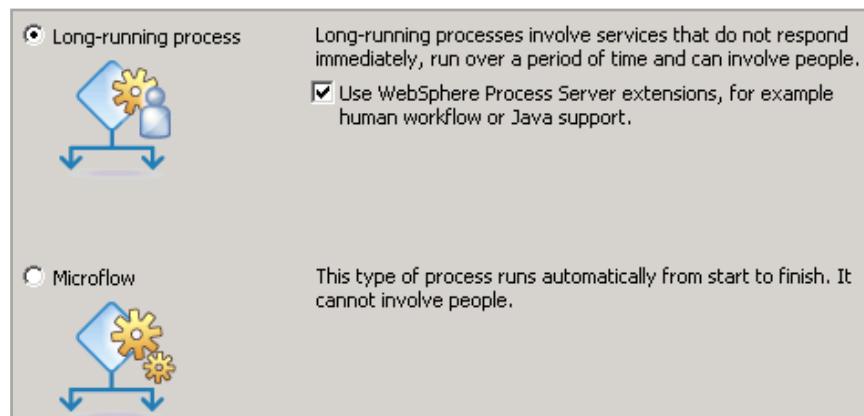
BPEL is an SCA implementation type. Therefore, business processes are exposed as service components, and the caller is unaware that the service is rendered in BPEL. The interface (interface partner) provides the information of how to invoke the BPEL (operations that are provided and the types of inputs, outputs, and fault messages). Processes can be invoked as SCA components.

BPEL implements reference partners and interface partners (called partner links) that are converted into SCA component interfaces and references.

SCA and SDO are used for internally doing the steps of the business process; they represent the data and how clients can work with business processes. Clients can easily invoke a business process as an SCA component and pass data to the process in the form of business objects (SDO). The way that business processes work with other services is also defined in SCA. You can create a service component with a BPEL implementation and wire it to a partner link (if it exists in the same module). Or you can wire the component to an import (if the partner exists in another module). Business processes can also be exposed to the outside world by creating an export component or by using a stand-alone reference (if a non-SCA component in the same module calls the business process).

Microflows versus long-running processes

- Business processes run on IBM Process Server as either microflows (short-running) or long-running processes
 - Known as non-interruptible and interruptible processes
- A microflow process is used for running short business processes or small units of work within a larger business process (subprocesses)
 - Microflows complete or fail and are not persisted
- A long-running process might run for hours, days, or weeks
 - Frequently involves components with lengthy response times such as human tasks
 - State of the process must be persisted



© Copyright IBM Corporation 2016

Figure 8-9. Microflows versus long-running processes

WB8601.2

Notes:

The first type of business process is a non-interruptible process that is used for running short business processes or a small unit of work within a larger business process (a subprocess). A microflow business process either completes or fails. No intermediate state can be maintained.

Consider a credit card verification process. This business process is short and simple: accepting personal information and returning a result, which indicates whether the credit card was valid. In this business process, it might start with a number of lookups that check information about the individual before returning the results. These lookups either succeed and return a result, or fail and the entire credit card validation fails. The client can monitor the failure, and can call the business process again. In any case, no state information is being maintained in the different lookup steps.

The second type of business process is long-running (interruptible). These business processes might take hours, days, or weeks to complete, and because of the possible random completion time, state must be maintained during execution. If state is not maintained and the current execution thread that is running the business process ends (for example, if the server fails), the work that is accomplished would be lost.

An example of a long-running business process is a loan approval process. A loan approval process gathers personal financial information and evaluates it. If the individual is financially sound,

a loan can be offered and the funds can be reserved. Because the evaluation of the financial information can take a number of days, the state of this business process must be maintained. After each activity in the process is completed, the results must be recorded to follow which information was processed and how much work was completed.

When you create a business process in IBM Integration Designer (either long-running or microflow), you must elect whether to use IBM extensions to BPEL in the process. To disable the extensions, clear the **Use IBM Process Server extensions** check box in the new business process wizard. If you decide to turn off the extensions, you lose the following functions:

- **Process type**
 - Microflow
- **Expression language**
 - Java
- **Actions**
 - Human task
 - Snippet
 - Generalized flow
 - Collaboration scope
- **Properties for all activities (including the process)**
 - Description
 - Documentation
 - Display name
 - Custom properties (not applicable for structures)
 - Enable persistence and queries of business-relevant data
- **Process properties**
 - Auto-delete
 - Autonomy (whether a process runs as a peer or as a child of the invoking process)
 - Compensation sphere
 - Valid from
 - Ignore missing data (use this option to suppress runtime faults when data is missing during assignments)
- **Extensions for specific activities**
 - Scopes that can be flagged as non-compensable
 - Transactional behavior and the “continue on error” function of the invoke activity
 - Compensation of the invoke activity
 - Expiration setting on the invoke activity
 - Administrative tasks for processes, invokes, and snippets

- Authorization tasks for receive, OnMessage, and OnEvent
- Query properties on variables
- Use of data type variables on the details tab of messaging activities like invoke, receive, reply, OnEvent, OnMessage

The following lists some cases when you would **not** want the extension enabled:

- When you are designing the process to use or edit in another set of tools
- When you are planning to run the process in a runtime environment other than IBM Process Server
- When you want to exchange information with a business partner who is not using the IBM Integration Designer set of tools or the IBM Process Server runtime environment

8.2. Elements of BPEL processes

Elements of BPEL processes



© Copyright IBM Corporation 2016

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

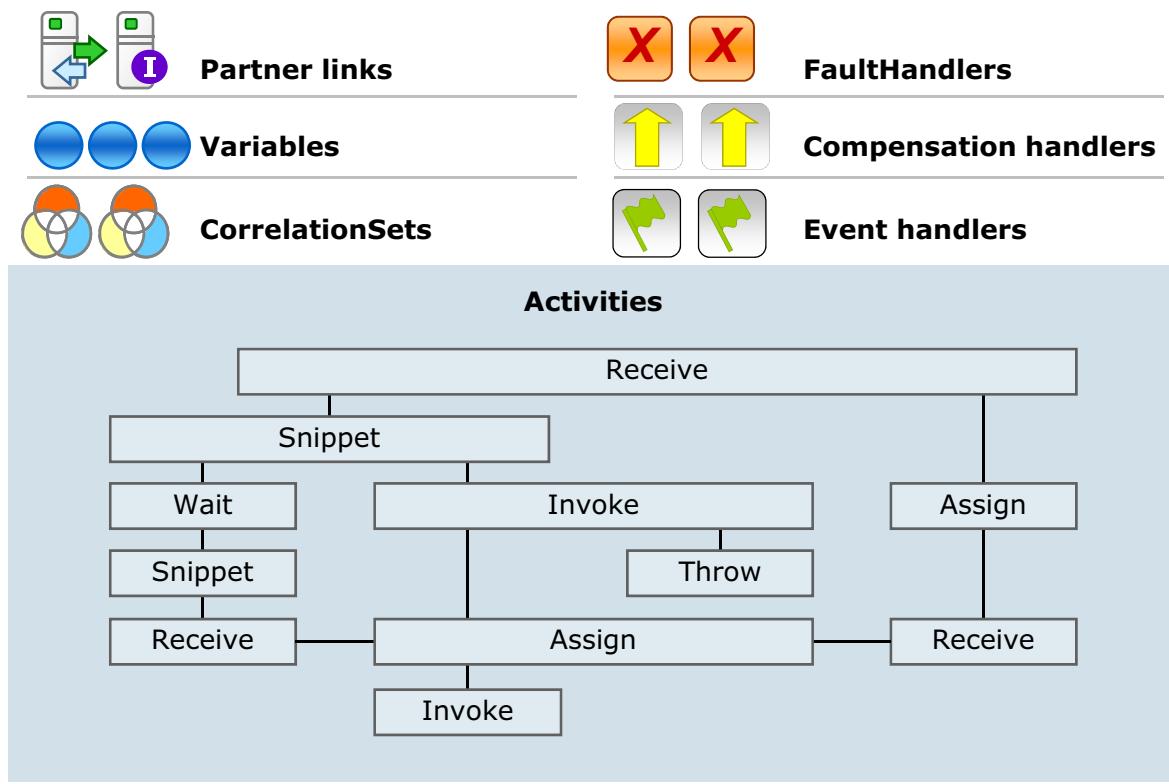
10.1

Figure 8-10. Elements of BPEL processes

WB8601.2

Notes:

Main elements and concepts of a BPEL process



© Copyright IBM Corporation 2016

Figure 8-11. Main elements and concepts of a BPEL process

WB8601.2

Notes:

Elements of a business process defined

Icon	Element description
	1. Partners define parties that interact with the process
	2. Variables specify information that is used while running the business process
	3. Correlation sets match messages to the correct process instance
	4. *Fault handlers recover from partial and unsuccessful work that is done in the current scope of the business process
	5. *Compensation handlers contain actions that do reverse operations for a particular scope or activity
	6. *Event handlers do work that is based on an event or an asynchronous message
Varied	7. *Activities are used to define the process logic

© Copyright IBM Corporation 2016

Figure 8-12. Elements of a business process defined

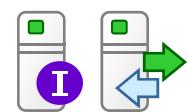
WB8601.2

Notes:

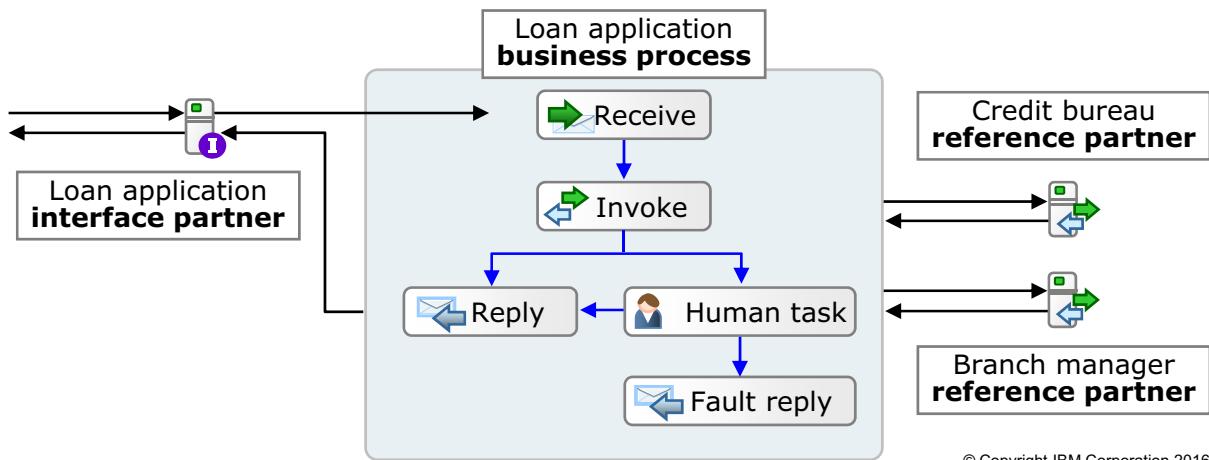
The BPEL specification defines the following seven primary elements or concepts:

- Partner links (partners)
- Variables
- Correlation sets
- Compensation handlers
- Event handlers
- Activities
- Fault handlers

Process elements: Partners



- **Partners** represent the external entities with which your business process interacts
 - An interface file describes the operations that the partner advertises
- The business process engine recognizes two categories of partners:
 - **Interface partners** define the services that the business process provides, which clients can invoke
 - **Reference partners** define the services that the business process can invoke



© Copyright IBM Corporation 2016

Figure 8-13. Process elements: Partners

WB8601.2

Notes:

A partner is an abstract entity that represents an external service with which the business process interacts. The WS-BPEL specification introduces the notion of a partner link to describe a reference to one of these entities. IBM Integration Designer automatically creates the partner links for you when you create a partner.

The WS-BPEL specification does not differentiate between interface and reference partners. Both types of partners exist in the same category. A partner link might have both an interface and a reference role. Certain operations represent entry points into the business process, while other operations represent outbound calls from the process to other services.

In SOA, each service becomes a building block of a larger service. To define the service interaction point, a partner link component is necessary. This page introduces the concept of an interface partner (where data is coming into this business process) and a reference partner (where data gets sent to an external service for processing).

Process elements: Variables



- **Variables** hold data that represents the state of a process
- Variables:
 - Can be internal, received from partners, or sent to partners
 - Can be inputs or outputs for Invoke, Receive, and Reply activities
 - Can be manipulated from Assign and Snippet activities
 - Can be global (process-level) or local
- Variables are typically associated with WSDL message types
 - XSD data types are also available
- Variables are tied to scopes (containers of one or more activities)
 - Initialized by the runtime per the scope before they can be used
 - You can control the initialization order for multiple variables
- Global variables:
 - Are available to the process and embedded scopes
 - Can be queried at run time
- The process tray shows both global variables and variables that are defined in the selected scope (local variables)
 - Can drag variables from the tray onto activities to assign inputs and outputs

© Copyright IBM Corporation 2016

Figure 8-14. Process elements: Variables

WB8601.2

Notes:

Variables represent the state of the business process. Variables store information about data that is coming into the business process, which is then reused and sent to the various steps that are run as part of the business process. Variables are also used to hold the data that is returned from a service back into the business process or to hold internal information such as counter values for iterative processes. The variable capability enhances in line with the BPEL specification to allow the use of XSD data types. WSDL message types can also be used to facilitate the service-oriented architecture that BPEL builds upon. The business process steps are defined as an interface that has messages, and those messages can be used as the basis for your variable types.

After the business logic, one of the most important aspects of building a business process is working with variables and establishing the state. It includes how the state is used and how information is passed from one activity to another. In IBM Process Server, variables must be initialized in the business process engine or explicitly initialized by the developer.

Variables can be initialized in the business process engine by specifying the variable as the destination for a receive activity, a receive choice activity, or an event. With the receive activity, information comes in the form of a message, which is stored in a variable. The business process engine initializes that variable, and you can use it later. You can also take parts of the variable and

move them to another variable. To use the second variable, you must first move some data into it or explicitly instantiate it.

In WS-BPEL V1.1, the XPath extension function `GetVariableData()` is used to access process variables. In WS-BPEL V2.0, XPath variable references (the `$variable` notation) are used to access elements of process variables or the status of a link. XPath variable notation applies to all XPath expressions and queries in assign activities or for all kinds of conditions. This notation represents a syntax change only. The WS-BPEL 1.1 notation is still supported.

You can set initialization values for BPEL variables and initialize complex business objects with the XML Literal option. To easily initialize variables that point to complex business objects, use the Value Composer, which is part of the assignments in an assign activity. You can initialize variables according to BPEL V2.0 specification. You can move variables up or down in the BPEL editor tray to control the order of initialization since ordering of variables when they are initialized can make a difference in how the application runs. You can drag variables onto invoke, receive, reply, and human task activities to assign inputs and outputs. In addition, you can drag variables onto collaboration scope activities to set the folder variable if the variable is of type caseFolder. You can drag a variable onto a forEach activity if the variable is of type array, and you can drop a variable onto a throw activity if the variable is a fault variable.

One of the primary ways to work with variables is the assign activity. An assign is used to move data from one variable to another and to manipulate data. For more complex operations, visual or text snippets can be used. Visual snippets offer many visual constructs that you can use to work with variables and operations from a predefined set of library functions.

Visual snippets generate Java code for you, simplifying the process of working with variables by reducing the need for integration specialists to understand low-level details. Text snippets, or “text mode,” uses Java programming and data object APIs to work with variables.

Using the variables tray, you can drag variables onto invoke, receive, reply, and human task activities to assign inputs and outputs. You can drag variables onto collaboration scope activities to set the folder variable if the variable is of type caseFolder. You can drag variables onto forEach activities if the variable is of type array, and you can drag variables onto throw activities if the variable is a fault variable.

Process elements: Correlation sets (1 of 2)



- **Correlation sets** route an incoming message to a long-running business process instance
 - Messages must contain a set of variables and values that uniquely identify the contents
- A correlation set has a name and is composed of correlation properties
 - Each property consists of a name and a data type
 - Message contents are mapped to correlation properties
- At run time, values in the message determine the business process instance to which that message is routed
 - The unique values need to be initialized the first time the correlation set is used
- Correlation sets can be specified for processes or for individual activities

© Copyright IBM Corporation 2016

Figure 8-15. Process elements: Correlation sets (1 of 2)

WB8601.2

Notes:

Correlations direct requests to a business process instance, and long-running processes require them. For example, if 1000 loan applications are “in flight” and the credit department is sending credit report messages for multiple credit applications, correlation sets ensure that the messages are sent to the correct instances. Within a correlation set, you define the parts of an incoming message that indicate uniqueness. These parts then map to some type of value within the business process state. The messages can then be matched with an instance that contains the same unique message parts.

It is important to note that correlation sets are provided from unique values in application data. An example would be a social security number, order number, customer number, or some value that is defined as part of the business process. Correlation sets can consist of a single value or multiple values, and multiple correlation sets can exist as the long-running business process runs over a time period. As the business process state changes, different unique identifiers might be used to identify an instance. The correlation set must be initialized, typically at the beginning of a long-running business process.

Process elements: Correlation sets (2 of 2)

- Business processes maintain state information when interacting with partners
- The business process engine has the following characteristics:
 - Accepts an incoming message
 - Correlates the message elements to the parts in the correlation set
 - Compares the message values to the values in the set
 - The message is directed to the appropriate business process instance that matches the values in the set

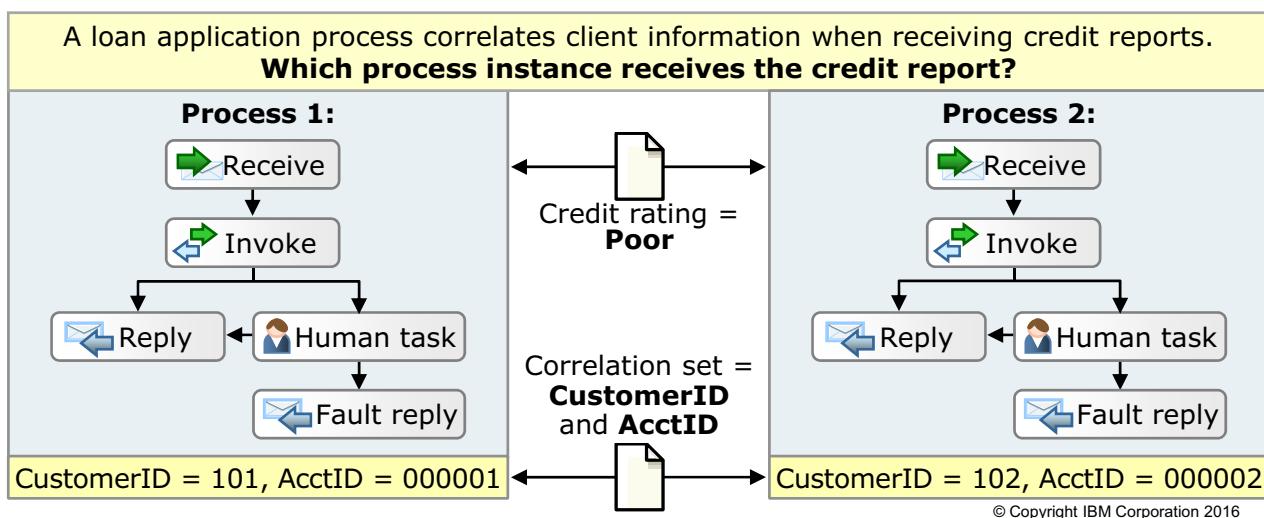


Figure 8-16. Process elements: Correlation sets (2 of 2)

WB8601.2

Notes:

Correlation sets are used in runtime environments when multiple instances of the same process run. The sets allow two partners to initialize a BPEL process transaction, temporarily suspend activity, and then recognize each other again when that transaction resumes.

In the slide, two running instances of the loan application process exist. While both of the customers have a credit rating of “poor”, they need to recognize their own instance of the “poor” credit report. For the two customers to recognize their “own” process instances, the correlation set composed of CustomerID and AcctID uniquely associates the correct process to the correct customer. This association ensures that the correct credit rating reaches the correct customer.

The process choreography runtime environment does not generate the correlation ID (although the business process container does map messages to unique instances). The correlation ID is required at the time that the business process instance is started. Typically, the correlation ID value is some unique value that is known to the starter of the business process or is generated and returned to the starter before the business process is started. When the business process is started, the correlation ID is passed on the input message.

Consider a stock trading process where many different customers might be looking to buy or sell stocks. Each customer would contact the customer’s broker with the intention to trade a particular

stock. The broker would start an instance of the stock trade process with the customer number as the unique identifier. The customer number can be specified on the request and matched to a particular running instance.

Process administrators can repair correlation sets. Modeling or runtime errors can cause the initialization of a correlation set to fail even though the workflow engine needs the values. Business Process Choreographer Explorer uses new correlation set APIs and provides a repair page to correct this problem.

Process elements: Activities

- **Activities** are the individual business tasks that implement the larger business goal that the process represents
- An activity can be basic, structured, or associated with error processing
 - Basic activities do not contain other activities: human task, snippet, or reply
 - Structured activities are activities that contain other activities: sequence, or while loop
 - Activities for fault handling and compensation are activities that process expected and unexpected error conditions
- BPEL also uses handlers with certain activities
 - Handlers contain activities that are run as a result of events or during error processing

© Copyright IBM Corporation 2016

Figure 8-17. Process elements: Activities

WB8601.2

Notes:



Unit summary

- Describe the purpose and business value of using the WS-BPEL standard
- Describe the function of the business process container
- Describe the difference between long-running and microflow (short-running) business processes
- List and describe the seven parts of a business process

© Copyright IBM Corporation 2016

Figure 8-18. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. List the seven main parts of a business process.
2. Define the concept of a partner link.
3. What is the purpose of a correlation set?
4. What are the two types of business processes?

© Copyright IBM Corporation 2016

Figure 8-19. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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

Checkpoint answers

1. Activities, fault handlers, event handlers, compensation handlers, variables, partners (partner links), and correlation sets.
2. Partners represent the external entities with which your business process interacts.
3. A correlation set routes an incoming message to the running business process instance that should handle it.
4. Microflows and long-running processes.

© Copyright IBM Corporation 2016

Figure 8-20. Checkpoint answers

WB8601.2

Notes:

Exercise 6



Creating a business process, part I

© Copyright IBM Corporation 2016

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

10.1

Figure 8-21. Exercise 6

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create a business process
- Implement WS-BPEL interface partners and reference partners
- Create process variables
- Compare business processes between IBM Integration Designer and IBM Process Designer

© Copyright IBM Corporation 2016

Figure 8-22. Exercise objectives

WB8601.2

Notes:



Course business scenario: What you work with in Ex 6

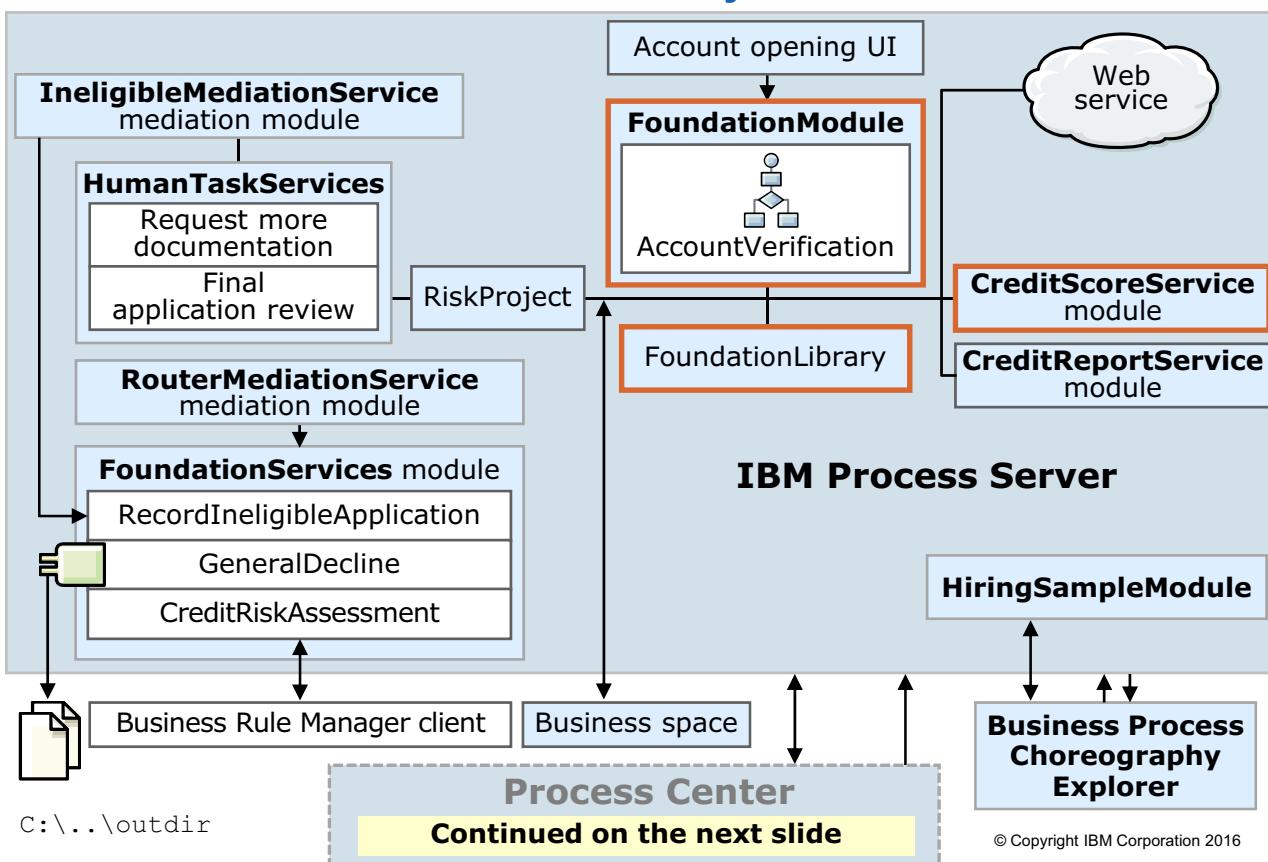


Figure 8-23. Course business scenario: What you work with in Exercise 6

WB8601.2

Notes:

Account verification process: Overview

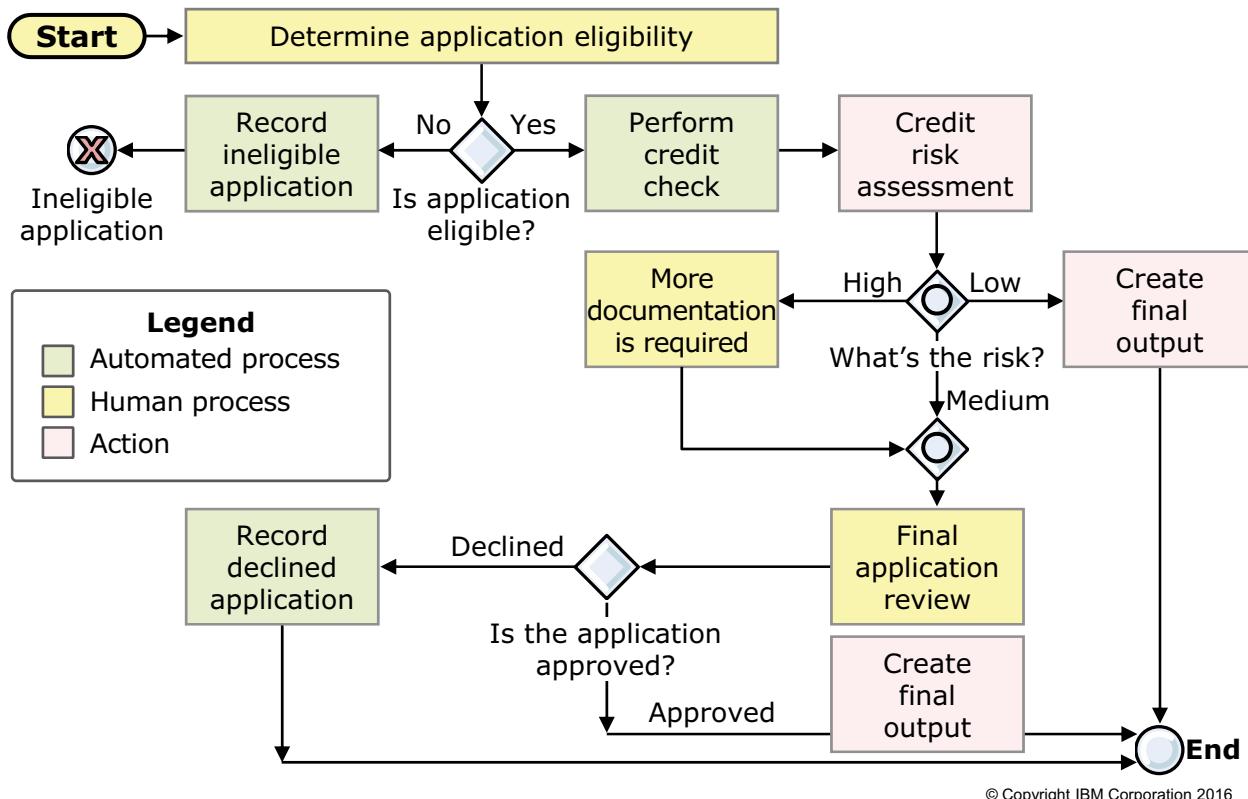


Figure 8-24. Account verification process: Overview

WB8601.2

Notes:

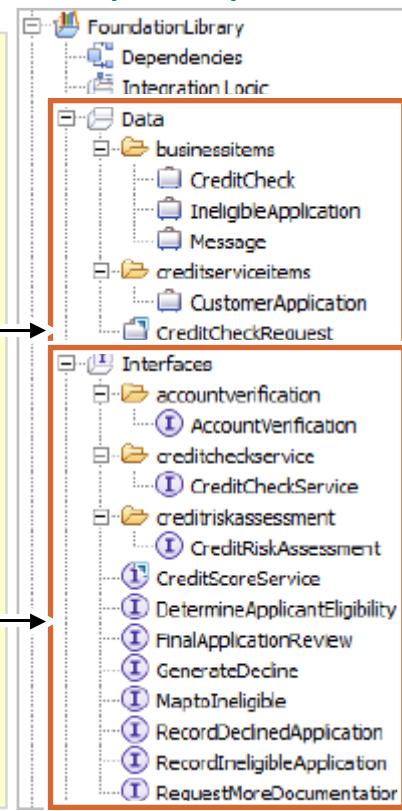
Why create the Account verification business process in IBM Integration Designer?

The business process application that you examined in IBM Process Designer in Exercise 1 is incomplete. Although the activities were in place, and in a later exercise you wired them together, each individual activity did not accomplish anything: they had no implementations. In this exercise and the subsequent exercises, you build this process in IBM Integration Designer; but more importantly, you focus on building the implementations for the individual activities.

Components that are required for Exercise 6 (1 of 7)

Prebuilt components that are imported in the lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary container for business objects:**
 - CreditCheck
 - IneligibleApplication
 - Message
 - CustomerApplication
 - CreditCheckRequest
- 4. FoundationLibrary container for interfaces:**
 - AccountVerification
 - CreditCheckService
 - CreditRiskAssessment
 - CreditScoreService
 - DetermineApplicantEligibility
 - FinalApplicationReview
 - GenerateDecline
 - MaptoIneligible
 - RecordDeclineApplication
 - RequestMoreDocumentation



© Copyright IBM Corporation 2016

Figure 8-25. Components that are required for Exercise 6 (1 of 7)

WB8601.2

Notes:



Components that are required for Exercise 6 (2 of 7)

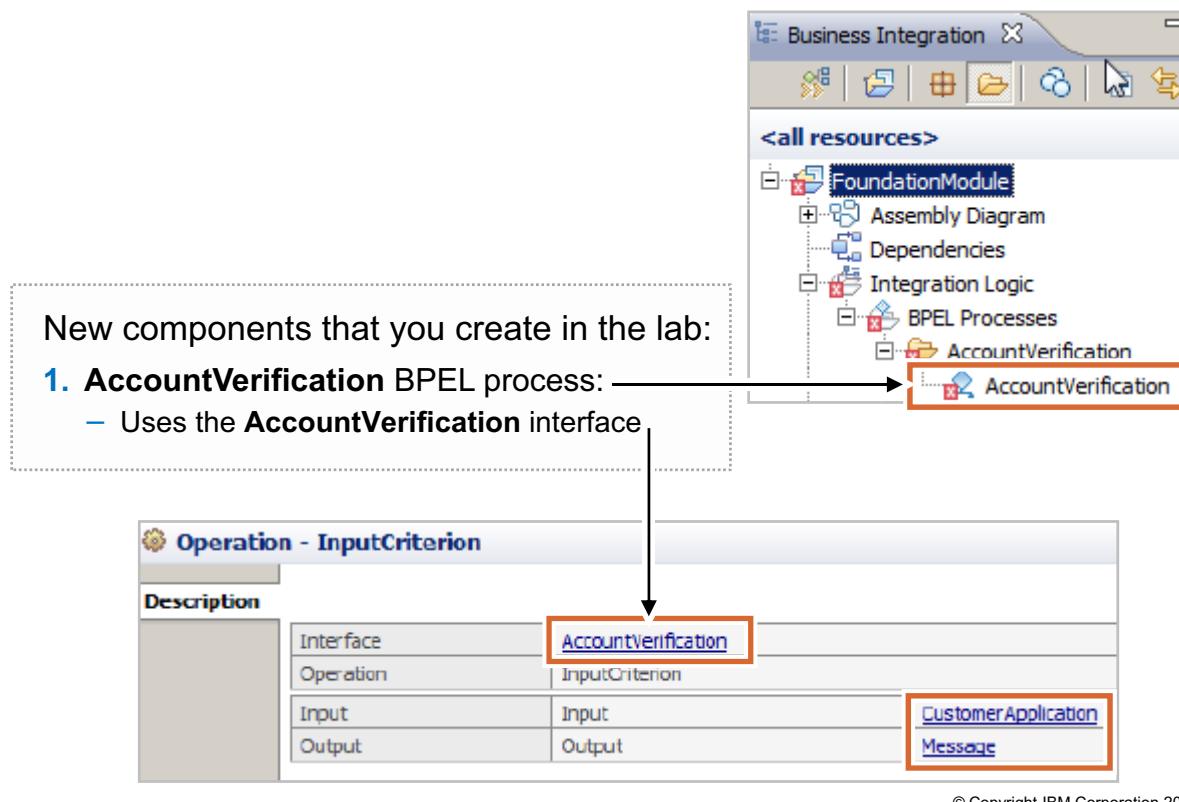


Figure 8-26. Components that are required for Exercise 6 (2 of 7)

WB8601.2

Notes:

An interface provides the input and output of a component. It is created independent of the internal implementation of the component. The AccountVerification interface dictates how to use the AccountVerification process. It specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions on the AccountVerification process.

The interface partner is a direct link to the interface where the partner is configured. An interface partner is the process interface, and it exposes operations that external users or services can call.

Components that are required for Exercise 6 (3 of 7)

New components that you create in the lab:

2. Global variables:

- CreditCheckVariable
- IneligibleApplicationVariable
- CustomerApplicationVariable
- CustomerApplicationVariable2
- MessageVariable

Variables
CustomerApplicationVariable
MessageVariable
CreditCheckVariable
IneligibleApplicationVariable
CustomerApplicationVariable2

© Copyright IBM Corporation 2016

Figure 8-27. Components that are required for Exercise 6 (3 of 7)

WB8601.2

Notes:

The strength of IBM Integration Designer is to use the Service Component Architecture, so other implementations can be placed in separate modules. Because implementations are spread throughout other modules, it maximizes reusability, and it necessitates intercommunication and passing variables between modules.

In this exercise, you add the following global variables to the AccountVerification business process. These variables store business objects for manipulation by the process activities:

- The global variable that is named CreditCheckVariable stores a CreditCheckRequest business object.
- The global variable that is named IneligibleApplicationVariable stores an IneligibleApplication business object.
- The global variable that is named CustomerApplicationVariable2 stores a CustomerApplication business object.
- CustomerApplicationVariable and MessageVariable correspond to the type of business objects that form the input and output variables for the AccountVerification process interface.

Components that are required for Exercise 6 (4 of 7)

New components that you create in the lab:

3. Reference partners:

- DetermineApplicationEligibilityPartner
- RecordIneligibleApplicationPartner
- CreditCheckServicePartner
- CreditRiskAssessmentPartner
- RequestMoreDocumentationPartner
- FinalApplicationReviewPartner
- GenerateDeclinePartner
- RecordDeclinedApplicationPartner
- MaptoIneligiblePartner

Reference Partners
DetermineApplicationEligibilityPartner
RecordIneligibleApplicationPartner
CreditCheckServicePartner
CreditRiskAssessmentPartner
RequestMoreDocumentationPartner
FinalApplicationReviewPartner
GenerateDeclinePartner
RecordDeclinedApplicationPartner
MaptoIneligiblePartner

© Copyright IBM Corporation 2016

Figure 8-28. Components that are required for Exercise 6 (4 of 7)

WB8601.2

Notes:

In this portion of the exercise, you add WS-BPEL reference partners to your process. Reference partners represent the service interfaces that your WS-BPEL activities call during process execution.

1. Add the DetermineApplicationEligibilityPartner reference partner with the DetermineApplicantEligibility interface.
2. Add the RecordIneligibleApplicationPartner reference partner with the RecordIneligibleApplication interface.
3. Add the CreditCheckServicePartner reference partner with the CreditScoreService interface.
4. Add the CreditRiskAssessmentPartner reference partner with the CreditRiskAssessment interface.
5. Add the RequestMoreDocumentationPartner reference partner with the RequestMoreDocumentation interface.
6. Add the FinalApplicationReviewPartner reference partner with the FinalApplicationReview interface.

7. Add the GenerateDeclinePartner reference partner with the GenerateDecline interface.
8. Add the RecordDeclinedApplicationPartner reference partner with the RecordDeclinedApplication interface.
9. Add the MapToIneligiblePartner reference partner with the MapToIneligible interface.



Components that are required for Exercise 6 (5 of 7)

New components that you create in the lab:

4. Interface partner

- AccountVerification interface partner →

The screenshot shows the 'Variables' component selected in the 'AccountVerification (AccountVerification)' palette. The palette also contains 'Interface Partners', 'Reference Partners', 'Correlation Sets', and 'Correlation Properties'. A red box highlights the 'Variables' item.

New BPEL Process

Select an Interface

You can create a new interface with one or more operations

Generate a new interface

Select an interface

Interface: **AccountVerification**

Select all of the operations that start the process. If all of the selected operations are not starting operations, then the generated BPEL process will contain errors.

Operations: **InputCriterion**

- During process creation, the **AccountVerification** interface is added to the process
- By default, this interface becomes an interface partner

© Copyright IBM Corporation 2016

Figure 8-29. Components that are required for Exercise 6 (5 of 7)

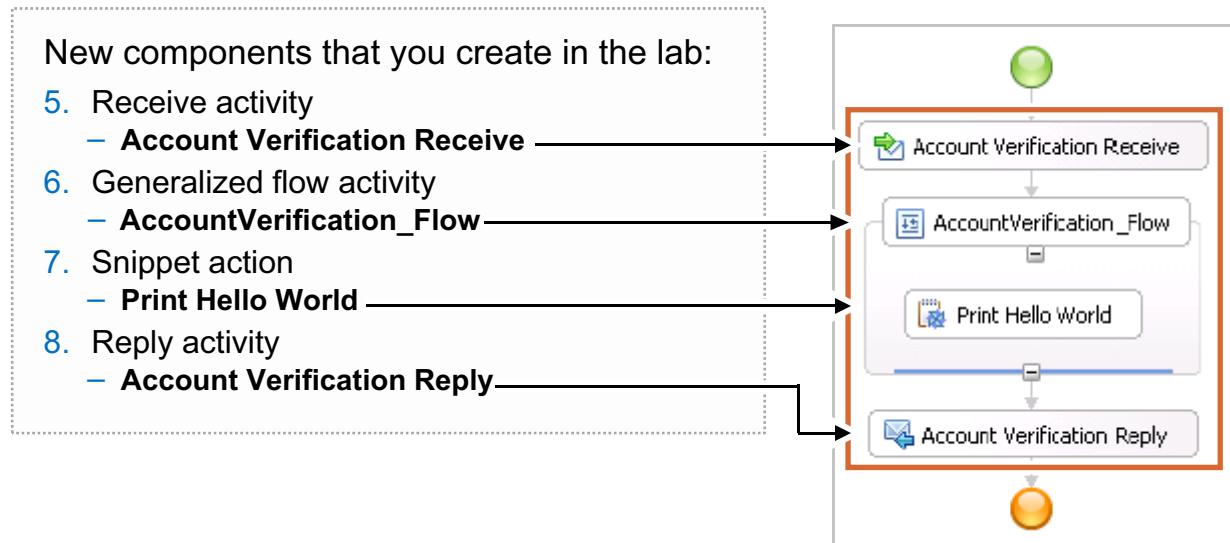
WB8601.2

Notes:

An interface provides the input and output of a component. It is created independent of the internal implementation of the component. The AccountVerification interface dictates how to use the AccountVerification process. It specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions on the AccountVerification process.

The interface partner is a direct link to the interface where the partner is configured. An interface partner is the process interface, and it exposes operations that external users or services can call.

Components that are required for Exercise 6 (6 of 7)



© Copyright IBM Corporation 2016

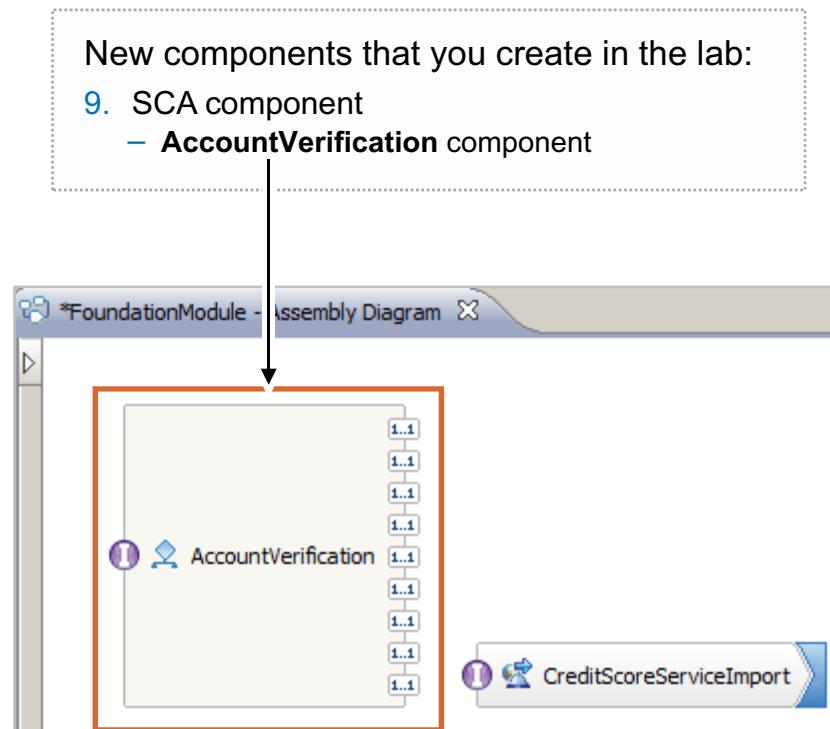
Figure 8-30. Components that are required for Exercise 6 (6 of 7)

WB8601.2

Notes:

In this exercise, you add a generalized flow that is a structured activity that acts as the container for the simple and complex BPEL activities.

Components that are required for Exercise 6 (7 of 7)



© Copyright IBM Corporation 2016

Figure 8-31. Components that are required for Exercise 6 (7 of 7)

WB8601.2

Notes:

Unit 9. Business process basic and structured activities

What this unit is about

This unit describes the various types of activities in BPEL, including basic and structured activities.

What you should be able to do

After completing this unit, you should be able to:

- List and describe the basic activities for business processes
- Define each of the available structured activities for business processes

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- List and describe the basic activities for business processes
- Define each of the available structured activities for business processes

© Copyright IBM Corporation 2016

Figure 9-1. Unit objectives

WB8601.2

Notes:



Topics

- WS-BPEL basic activities
- WS-BPEL structured activities

© Copyright IBM Corporation 2016

Figure 9-2. Topics

WB8601.2

Notes:

9.1. WS-BPEL basic activities

WS-BPEL basic activities



© Copyright IBM Corporation 2016

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

10.1

Figure 9-3. WS-BPEL basic activities

WB8601.2

Notes:

Overview of WS-BPEL activities

- Activities are the individual business tasks that operate with each other to implement the larger business goal, which is represented as the process that contains them
- An activity can be one of several different types: basic activities, structured activities, or activities that are associated with error processing
 - Basic actions are activities that have no structure and do not contain other activities: human task, snippet, and reply
 - Structured activities are activities that contain other activities: sequence and while loop
 - Activities for fault handling and compensation are activities that process expected and unexpected error conditions in processes
- BPEL also uses handlers with certain activities
 - Handlers contain other activities that run based on events or during error processing

© Copyright IBM Corporation 2016

Figure 9-4. Overview of WS-BPEL activities

WB8601.2

Notes:

As you add activities to a business process in IBM Integration Designer, you can add custom properties to each individual activity. The generated business user clients use the ones that can be queried, and put through a set of APIs.

Custom properties can be used to specify more properties for activities or the process (for example, to associate costs with activities). A custom property has a name and an optional (string) value. The value of custom properties can be specified during authoring time and at run time on a per-instance basis.

Basic activities: Receive, reply, and invoke

- The **receive** activity receives a message sent to a business process
 - Can start a new business process
 - Can restart an existing process
 - Request can be synchronous or asynchronous
- The **reply** activity responds to a message received
 - Typically used as a response to a synchronous request
 - Can return either a response message or a fault message
- The **invoke** activity calls a one-way or a request/response operation that a partner offers
 - Calls another service or business process



© Copyright IBM Corporation 2016

Figure 9-5. Basic activities: Receive, reply, and invoke

WB8601.2

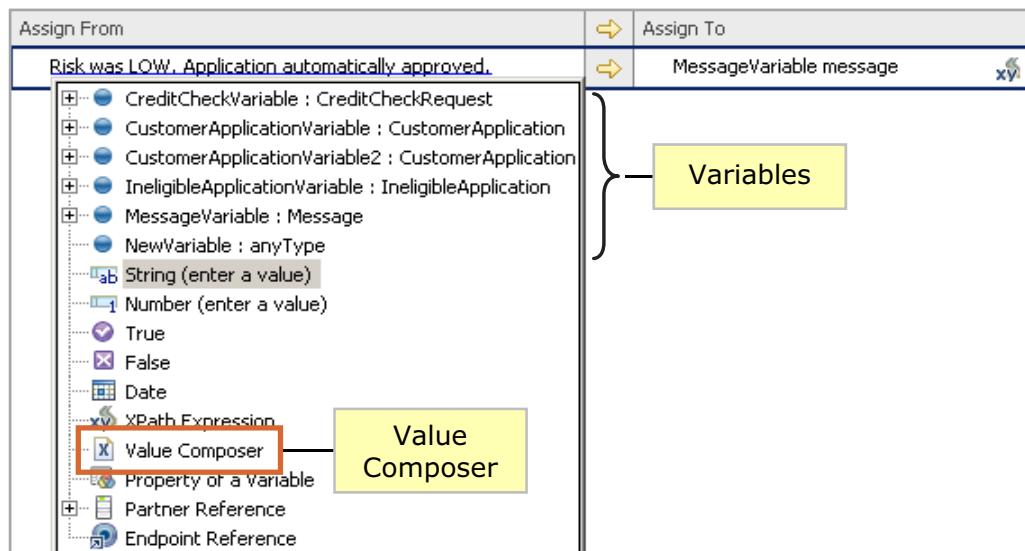
Notes:

The receive activity is one of the primary means of accepting information into a business process and is typically used at the beginning of the process to accept data and start an instance. The reply activity can be used only with a receive activity and can return a response message (indicating success) or any of various fault messages. Receive and reply can be used in long-running or non-interruptible business processes.

The invoke activity calls services by using partner links. Invocations can be one-way or request/response. It is not necessary to differentiate between synchronous and asynchronous invocations within the business process. The business process engine handles the invocation style for you.

Basic activities: Assign

- An **assign** activity is used to update the values of variables with new data
 - Values can be copied from source variables to destination variables
 - Full XPath support is provided
 - A Value Composer is provided to specify initial values for complex variables



© Copyright IBM Corporation 2016

Figure 9-6. Basic activities: Assign

WB8601.2

Notes:

Assign activities are used with variables. Variables represent the state of the business process and data that is sent to and received from the different services. Variables typically contain business objects.

Different services have different messages that must be populated with data, and the assign activity is the primary way to move information from one variable or message to another. Assign activities do basic mapping of information. For more complex types of mappings, the use of snippets is supported. The BPEL editor consists of a business object map capability, which can be used for mapping specific business objects.

IBM Process Server includes support for XPath V1.0. You can specify an XPath query string to retrieve information from business objects and messages.



Basic activities: Receive choice (1 of 2)

Select an Interface

You can generate a new interface with one operation, or use an existing interface with one, or more operations.

Generate a new Interface
 Select an existing Interface

Interface:

Select operations to start the process.

Operations:	<input checked="" type="checkbox"/> open
	<input checked="" type="checkbox"/> close
	<input checked="" type="checkbox"/> deposit
	<input checked="" type="checkbox"/> withdraw

Receive choice

- **Receive choice** selects one branch of activities to run based on a receive case
- Receive cases are interface operations

```

graph TD
    Start(( )) --> RC{Receive Choice}
    RC --> Open[Open]
    RC --> Close[Close]
    RC --> Deposit[Deposit]
    RC --> Withdraw[Withdraw]
    Open --> ReplyOpen[Reply-Open]
    Close --> ReplyClose[Reply-Close]
    Deposit --> ReplyDeposit[Reply-Deposit]
    Withdraw --> ReplyWithdraw[Reply-Withdraw]
    ReplyOpen --> Join(( ))
    ReplyClose --> Join
    ReplyDeposit --> Join
    ReplyWithdraw --> Join
  
```

© Copyright IBM Corporation 2016

Figure 9-7. Basic activities: Receive choice (1 of 2)

WB8601.2

Notes:

The receive choice activity is a combination of receive and choice. (Receive choice was previously called the pick activity). When a specific message is received, a matching path of activities is run.

Receive choice selects one branch of activities to run based on a receive case.

- Receive choice can be used to start a process instance.
- The activity waits for 1 of N possible messages.
- A correlation set is specified for each case.
- Each case can have different permissions.

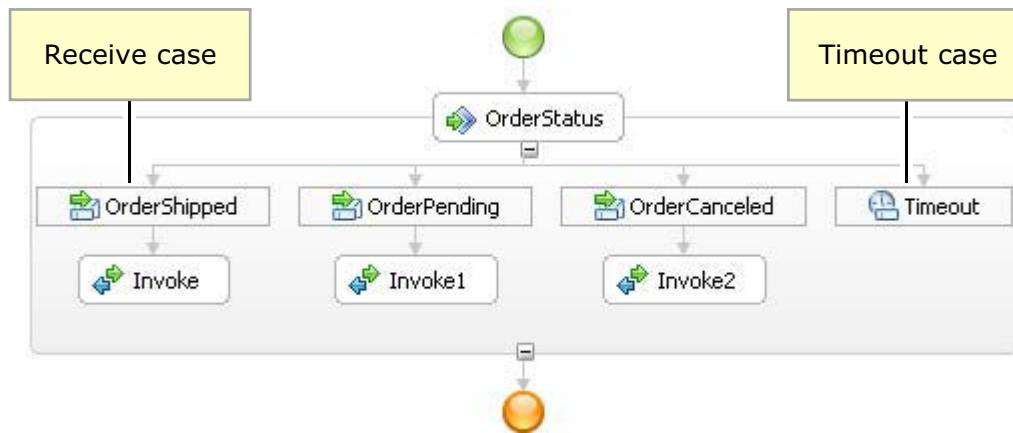
Receive cases are interface operations.

- If you select an interface with multiple operations when a process is created, it begins with a receive choice instead of a receive.

Basic activities: Receive choice (2 of 2)



- If a message is not received within a certain time, an expression evaluates a timeout case that runs a control path
 - Java and XPath expressions allow a date (visual, Java, or literal date-time) or a duration (visual or Java) value
 - Timeout expressions allow simple calendar, WebSphere CRON, or user-defined values



© Copyright IBM Corporation 2016

Figure 9-8. Basic activities: Receive choice (2 of 2)

WB8601.2

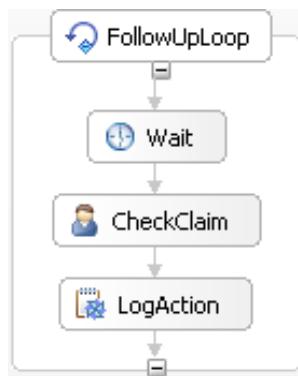
Notes:

The example that is shown here has three possible paths and a timeout in case none of the expected messages are received during the specific time period. If the order status receive choice activity receives an “order shipped” message, then the invoke activity runs. If an “order pending” message arrives, the invoke1 activity runs. If an “order canceled” message arrives, the invoke2 activity runs. While the receive choice activity is waiting for a specific message to arrive, the entire process instance is in a “sleep mode.” To avoid the process instance from using the server resources indefinitely, timeout logic can be defined. After a certain duration or on a certain date, the process stops waiting for a specific message and then continues.

Timeout is an IBM extension to BPEL. Calendar values are represented in Coordinated Universal Time (UTC). WebSphere CRON is a built-in calendar that uses a list of term expressions that represent elements of time to calculate the interval.

Basic activities: Wait

- A **wait** activity stops the business process for a specific amount of time that an expression evaluates
 - Java and XPath expressions allow a date (visual, Java, or a literal date-time) or duration (visual or Java) value
 - Timeout expressions allow simple calendar, WebSphere CRON, or user-defined values
 - Business calendars are also supported
- Wait is available only for long-running processes



© Copyright IBM Corporation 2016

Figure 9-9. Basic activities: Wait

WB8601.2

Notes:

The wait activity can be used to stop a business process on the current execution path for a specific time. The duration can be either a hardcoded value or calculated dynamically. XPath can be used, or Java can be used for a more complex calculation. Any length of time can be used for a wait activity; therefore, it is available only for use with long-running interruptible processes.

In the example on this page, the FollowUpLoop is used for follow-up handling that an insurance clerk triggers each time that the claim is checked. To repeat the follow-up handling until a decision is made, a while construct is used. When follow-up handling is triggered, the process waits for the time that is specified for follow-up; it is implemented as a wait activity.

The wait activity is available only for long-running (interruptible) business processes. Using the wait activity, you can make the process instance stop for a specific amount of time, or you can terminate the instance by defining a terminate activity.

It is possible to force completion of a BPEL wait activity by using the Business Process Choreographer Explorer.

Basic activities: Human task

- **Human task** activities are an IBM extension to BPEL
 - Human tasks are defined in the BPEL4People specification
- Human task activities in a business process are called inline human tasks
 - Inline tasks send process-related messages to humans for completion
 - Humans can send messages to processes by being given authorization to a receive or a case in a receive choice
- Assigning a task to a human involves the interaction between two editors in IBM Integration Designer
 - You use the process editor to compose a process that requires human interaction
 - You use the human task editor to configure the task



© Copyright IBM Corporation 2016

Figure 9-10. Basic activities: Human task

WB8601.2

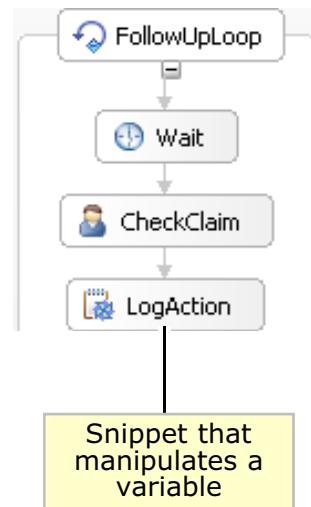
Notes:

For more information about human tasks, see Appendix C and Appendix D at the end of the Student Notebook.



Basic activities: Snippets

- The **snippet** activity allows execution of Java code in a BPEL process
- Java snippets are an IBM extension to BPEL
- The code runs locally in the BPEL process
 - Snippets in a BPEL process are compiled into a single class
- Snippets are typically used for working with the contents of variables
- Snippets are also used for:
 - Loop counters
 - Data validation
 - Data normalization
- Java snippets can be edited graphically (visual) or textually (Java)
 - A visual snippet can be converted to Java and edited textually
 - Moving back to a visual snippet after converting to Java is not supported



© Copyright IBM Corporation 2016

Figure 9-11. Basic activities: Snippets

WB8601.2

Notes:

The snippet activity is an IBM extension to the BPEL specification. Using snippet activities, you can work with business objects through SDO API calls. You can use an assign to do more robust processing on variables and business process states when the anticipated action cannot be completed. Although the BPEL specification does not support it, the snippet is one of the most useful activities. IBM Integration Designer includes a visual snippet editor, or you can use “plain” Java.

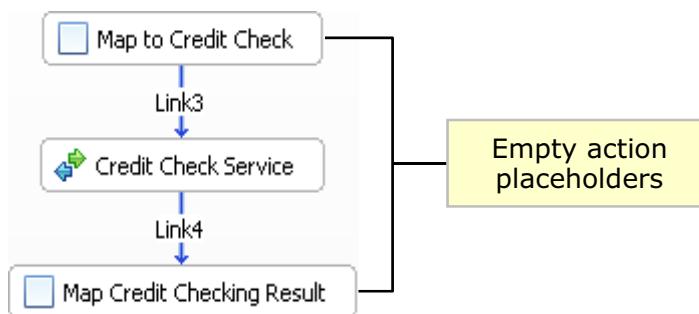
You can use the visual method to create the snippet and click the Java option to view the code that is used to implement that snippet. If you edit code in text mode, you are not able to revert to using the visual tool for that snippet. Snippets that are created in text mode cannot be viewed in visual mode.

A full set of library functions is available for snippets. You can also create your own custom snippets. Snippets can be reused and shared in a team programming environment.

Basic activities: Empty action



- An **empty action** activity acts as a “no-op” instruction in the business process
 - The empty action can be used as a synchronization point within the business process
 - To bring together parallel execution paths
 - Empty actions can also be used as placeholders
 - For a process activity implemented in the future
 - For development by someone else



© Copyright IBM Corporation 2016

Figure 9-12. Basic activities: Empty action

WB8601.2

Notes:

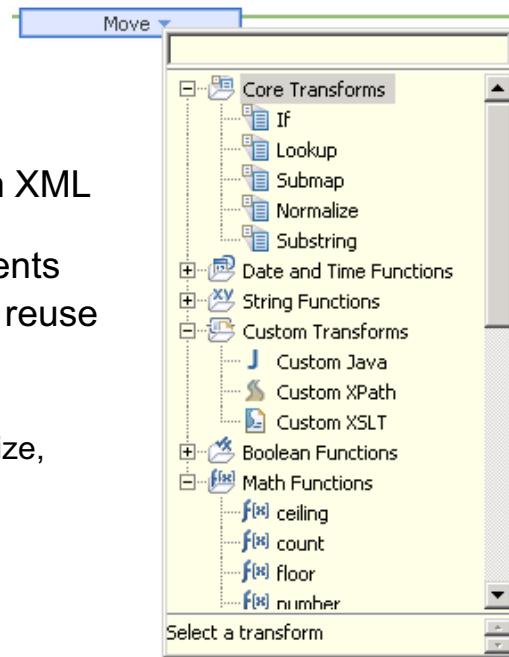
Use this activity as an undefined object to act as a placeholder within your process. You might do this activity if you are designing a process that you expect somebody else to implement, or if you are trying to synchronize the activities within a parallel activity.



Basic activities: Data map



- Data maps provide the structural and semantic transformation of business objects
- Maps can be business object maps or XML maps
 - Business object maps are needed only for relationships
- XSL stylesheets that are generated from XML maps are used in XSL Transformation primitives in the mediation flow components
- XML maps can be stored in catalogs for reuse
- Rich set of predefined transformations, including:
 - Core transforms: if, submap, lookup, normalize, and substring
 - Date and time functions
 - String functions
 - Boolean functions
 - Math functions
 - Custom transforms



© Copyright IBM Corporation 2016

Figure 9-13. Basic activities: Data map

WB8601.2

Notes:

A list of the supported transformations for data maps is included in the course appendixes. For a complete list of built-in transformation functions for XML maps, see the product documentation.

The purpose of a lookup function is to look up a key and return the specific value. This action is done through a comma-separated values (CSV) file, a properties file, or a custom function engine. For the custom function engine, it is easy to create your own .java file. The most frequent example for the lookup functions is the use of a hashtable to look up a key and retrieve another value.

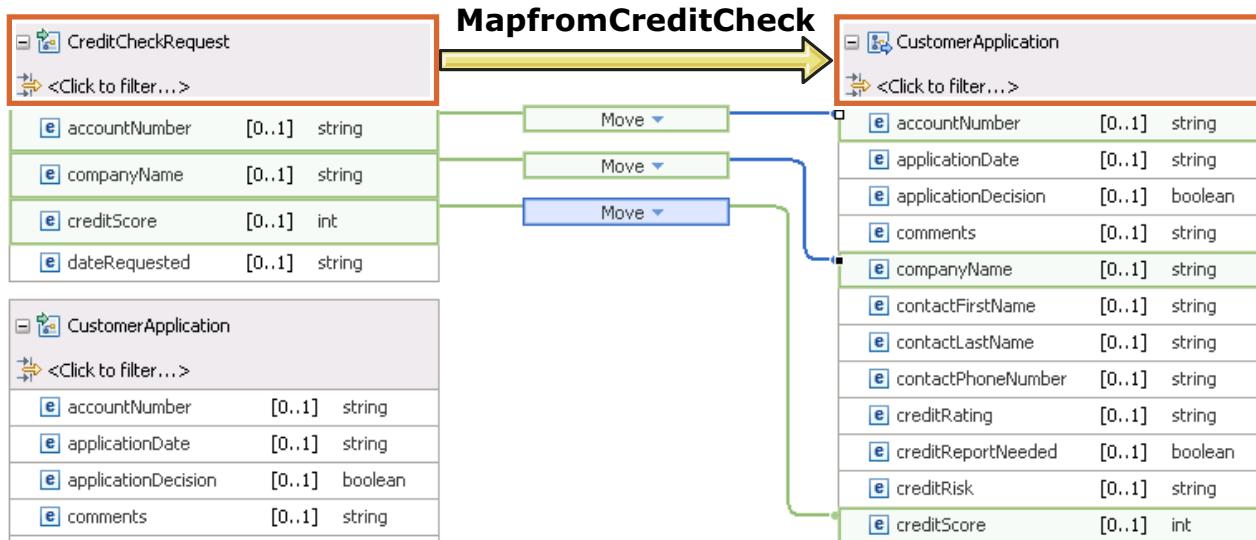
Using the map editor, you can quickly change a map transform into a submap and change from a submap into a local map. The reason for refactoring into a submap is to split up maps into reusable pieces. Large maps can be hard to manage and not reusable. Using submaps makes map content reusable and easier to manage. You can use the “refactor from submap” action to create a local map.

IBM Integration Designer includes a data map catalog. In the catalog, you can filter rows and sort by column headers like name, type, inputs, and outputs. You can also use a text search. By using the toggle buttons in the data map catalog, you can show and hide content tags and namespaces. In

addition, you can see other objects that use the map in the references view. Finally, you can create a data map from the data map catalog.

Relationships are covered in course WB858: *Developing Applications in IBM Business Process Manager V8.5.5 Advanced II*, which is the second week of training for the version 8.5.5 product.

Creating a data map



© Copyright IBM Corporation 2016

Figure 9-14. Creating a data map

WB8601.2

Notes:

To transform data between the source business object and the target business object, use a data map.

To create a data map:

1. In the business integration view, create two business objects.
2. To access the New Business Object wizard, click **File > New > Business Object**. Name these business objects: `BusinessObject1` and `BusinessObject2`
3. Create a BPEL process.
4. In the BPEL process editor, create two variables: `Variable1` and `Variable2`
5. For each variable, click the **Description** tab and in the Data type field, browse to one of the business objects that you created earlier, such that `Variable1` points to `BusinessObject1` and `Variable2` points to `BusinessObject2`.
6. Drop an empty action on the canvas.
7. Click the **Details** tab, and select the data map icon.

8. Select **XML map** or **Business object map** and click **Next**.
9. In the New Data Map wizard, give the new map a name and click **Next**.
10. Select one of the variables as the input and the other one as the output, and click **Finish**.

WebSphere Education

Common XPath editor

Simple XPath Expression Builder

Build a simple XPath Expression

Select a field and optionally add filters and conditions.

Expression:
\$checkInventory/orderDetails/orderItem[quantity > '5' and product = 'gizmos']/product = 'true'

Select a field:
<type to filter>

Aliases

- ID_OrderDetails : Business Item
 - ID_OrderDetails_orderItems (/orderItem) : Alias
 - ID_OrderDetails_numberOfItems (count(/orderItem)) : Alias
- orderItem : OrderItem [1..n]
 - ID_OrderItem : Business Item
 - product : ProductData
 - quantity : integer

The builder provides full XPath support and content assistance

Add an optional filter

Array Fields	Where	Operator	Value	and/or (optional)	Delete
orderItem	e quantity	>	5	and	
orderItem	e product	=	'gizmos'		

[Example...](#)

Add an optional condition

Operator	Value	Delete
=	true	

[Example...](#)

© Copyright IBM Corporation 2016

Figure 9-15. Common XPath editor

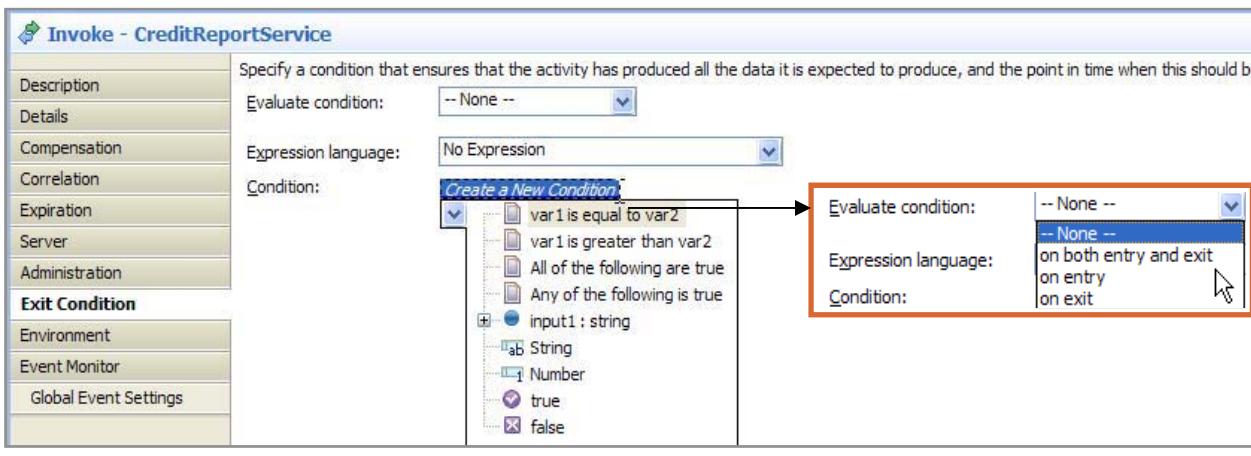
WB8601.2

Notes:

The common XPath editor is used throughout IBM Integration Designer wherever XPath expressions can be used. The XPath editor can be used for parameter transformations and for mediation flows and BPEL components (such as link conditions and timeout expressions). In the XPath expression builder, you first select a field with the ability to drill down or filter to find elements or business vocabulary aliases. Business vocabulary aliases give you the ability to assign aliases to XPath expressions so that you or your team does not remember complex XPath expressions, just aliases. Then, in the expression builder, you add optional filters or conditions on those elements. You receive immediate validation on the expression.

Activity exit conditions

- Most basic activities and human tasks support exit conditions
 - The **Receive choice** activity does not support exit conditions
- Exit conditions can be set:
 - On entry:** Determine whether an activity should be done
 - On exit:** Determine whether navigation continues as expected after completion
 - On both entry and exit**
- Expressions can be Java, simple (boolean), or XPath



© Copyright IBM Corporation 2016

Figure 9-16. Activity exit conditions

WB8601.2

Notes:

The following process activities support exit conditions: invoke, assign, receive, reply, wait, empty, snippet, data map, compensate, throw, rethrow, terminate, and human task. Exit conditions can be specified:

- On entry:** The criterion is evaluated immediately after the activity is initiated. If the exit condition evaluates to true, the activity is skipped; otherwise, the activity continues normally.
- On exit:** The criterion is evaluated before the activity is exited. If the exit condition evaluates to false, the activity is repeated; otherwise, the activity is completed and outgoing connectors are evaluated.
- On both entry and exit:** The criterion is evaluated immediately after the activity is initiated. If the exit condition evaluates to true, the activity is skipped. If the exit condition evaluates to false at entry, then the activity is run normally. When the activity is complete, the exit condition is evaluated again. If the condition evaluates to false, the activity is repeated; otherwise, the activity is finished and outgoing connectors are evaluated.

9.2. WS-BPEL structured activities

WS-BPEL structured activities



© Copyright IBM Corporation 2016

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

10.1

Figure 9-17. WS-BPEL structured activities

WB8601.2

Notes:

Structured activities: Sequences and parallel activities

- A **sequence** serializes the execution of nested activities
 - Activities run one at a time, in order
- A **parallel activity** depicts potentially parallel paths
 - Conditional logic is specified on interactivity links
 - Parallel activities in microflows run sequentially

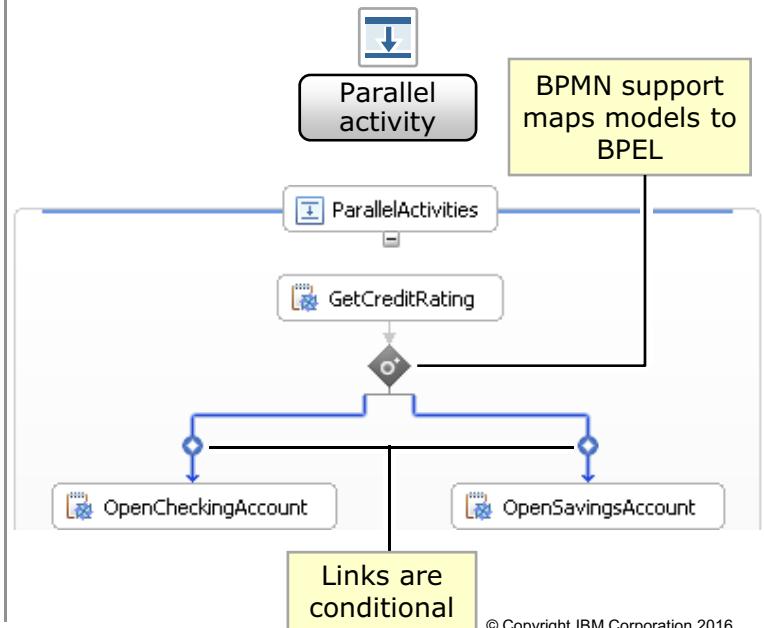
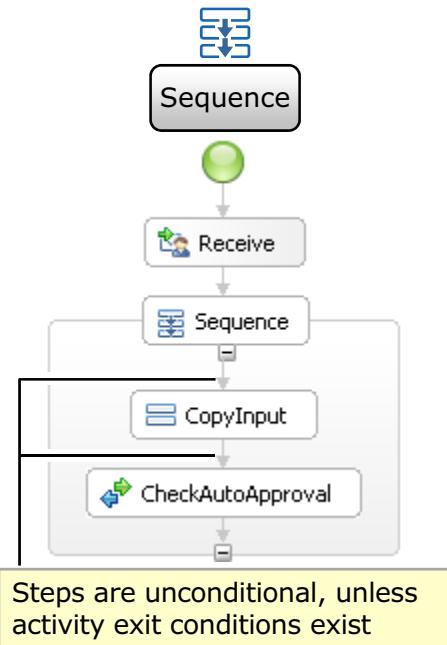


Figure 9-18. Structured activities: Sequences and parallel activities

WB8601.2

Notes:

Parallel activities run branches in parallel only in long-running processes. In a microflow process, all activities run on the same thread, even if they occur within a parallel activity. Parallel activities can use multiple threads in a long-running process. In a parallel activity, links are specified between activities. These links can include conditional logic. This logic can use variable state information from the business process to determine whether execution can proceed.

A sequence runs activities one after another, in order. In a sequence, an activity must complete successfully before the next one runs.

Links and link conditions

- Link conditions evaluate to true, false, or otherwise and are created by using:
 - Java expressions (text or visual)
 - XPath expressions
 - Simple expressions (basic true, false, or otherwise values)
- Multiple true links in structured activities can result in parallel execution paths in a long-running process
 - Parallel paths can be joined with conditions
 - Joins use the logical AND (process waits for execution from all paths), or the logical OR (wait for a single path), to determine when to continue to the next activity
 - If none of the links evaluate to true, a `joinFailure` fault is raised
 - Join failure settings can be used to suppress the fault, skip the activity that threw it, and continue process execution

© Copyright IBM Corporation 2016

Figure 9-19. Links and link conditions

WB8601.2

Notes:

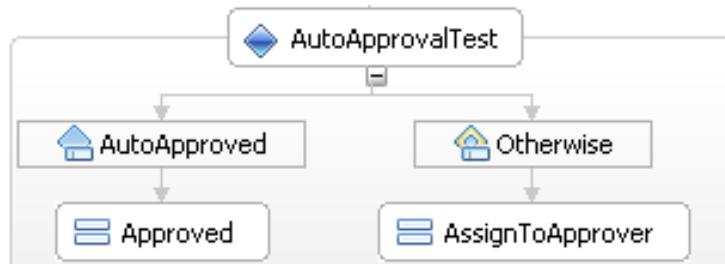
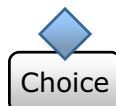
Links can include conditional logic to determine whether processing can proceed on a particular path. Links include support for XPath. Conditions can also be evaluated by using Java programming or a Boolean value. If multiple paths have conditional statements that originate from a single activity, an “otherwise” option can be set and used to run a path. If multiple conditions evaluate to true, parallel execution can be done during a long-running business process.

Join conditions are used to decide whether to wait for execution from all paths or just a single path before continuing a flow.

You can specify a display name for the BPEL link construct. Link names are displayed in the process editor and in the graphical process viewer.

Structured activities: Choice

- A **choice** activity selects one activity branch from a set of case elements that are based on a runtime condition
- The condition is created by using an expression language:
 - Java expressions (visual or text)
 - XPath expressions
 - Simple expressions (boolean true or false)
- At run time:
 - The first case element to evaluate to true is run
 - If no case element evaluates to true, the otherwise element runs
 - If otherwise is omitted, control passes to the activity after the choice activity



© Copyright IBM Corporation 2016

Figure 9-20. Structured activities: Choice

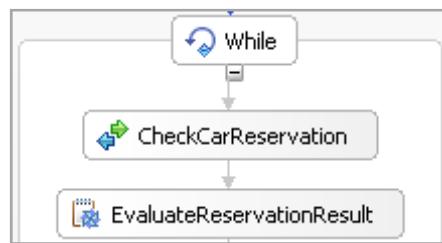
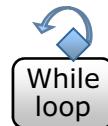
WB8601.2

Notes:

The *choice* activity was previously known as the *switch* activity. A choice activity is similar to a case statement. The first condition that evaluates to true determines the processing that takes place. Conditions are checked from left to right, and when one evaluates to true, the activities that are defined under that condition run. An “otherwise” case can also be used to account for situations where none of the conditions evaluate to true.

Structured activities: While loop

- If the condition evaluates to true, the **while loop** repeatedly runs the activities in its scope
- The condition is checked before the first iteration so it is possible that **none** of the nested activities run
- The condition can be created by using:
 - Java expressions (text or visual)
 - XPath expressions
 - Simple expressions (boolean true or false)
- A fault (handled or unhandled) terminates the loop
 - It is more efficient to end the loop by meeting the condition than by throwing an exception



© Copyright IBM Corporation 2016

Figure 9-21. Structured activities: While loop

WB8601.2

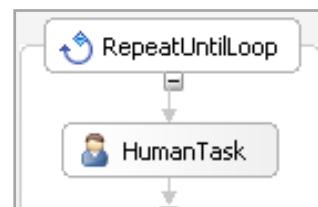
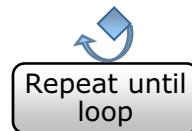
Notes:

The *while* activity is used to run activities iteratively. The activities in the scope of the while loop run while a defined condition remains true. The condition is evaluated before activity execution, so it is possible that none of the activities run.

A fault handler can be associated with the while activity to handle failure of any activity in the while loop that is not caught at the while loop construct level. If a fault does occur, any execution in that loop (including parallel activity) halts, and execution continues in the fault handler. Exiting a while loop in this manner is not the best way. You must try to meet the exit condition of the while loop, as more complexity is associated with throwing a fault. The use of faults must be reserved for handling failures in the business logic.

Structured activities: Repeat until loop

- The **repeat until loop** executes the activities in its scope at least one time and then continues to loop until the condition evaluates to true
- The condition is checked at the *end* of each iteration so the activities run **at least** one time
- The condition can be created by using:
 - Java expressions (visual or text)
 - XPath expressions
 - Simple expressions (boolean true or false)
- A fault (handled or unhandled) terminates the loop
 - It is more efficient to end the loop by meeting the condition than by throwing an exception



© Copyright IBM Corporation 2016

Figure 9-22. Structured activities: Repeat until loop

WB8601.2

Notes:

The *repeat until* activity provides for repeated execution of contained activities. The activities run until the condition evaluates to true. The condition is tested after each execution of the body of the loop, so the contained activities run at least once.

Structured activities: For each

- The **for each** activity enables the bundling of work
 - Requests that require a dynamic number of reviewers
 - Continue after some of the quotes are received
- Within the bundle, you control:
 - Whether a dynamic number of branches can be run serially or in parallel
 - Whether all branches are required for completion
 - Early exit criterion that specifies termination after a certain subset of branches

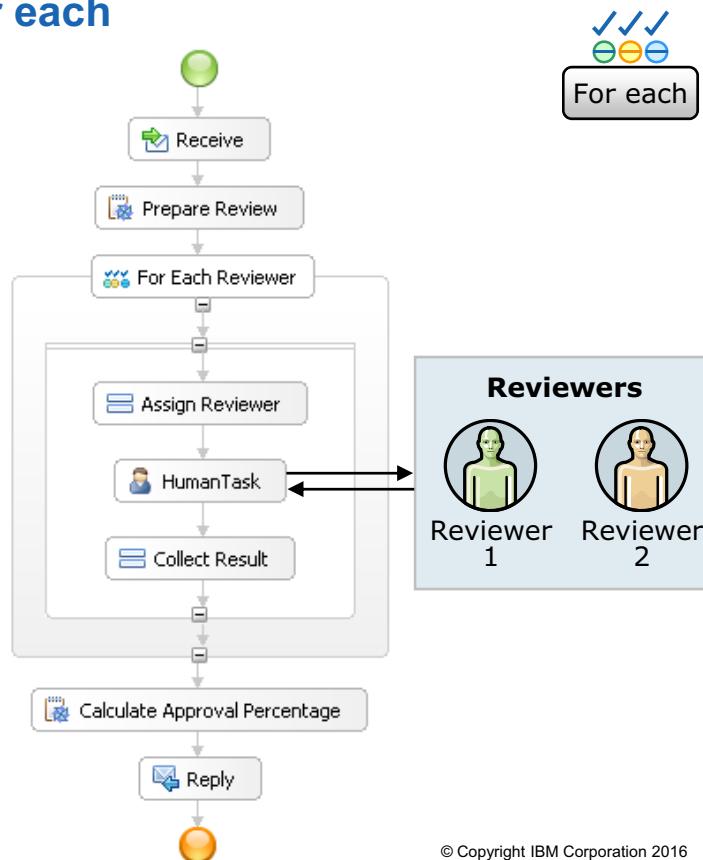


Figure 9-23. Structured activities: For each

WB8601.2

Notes:

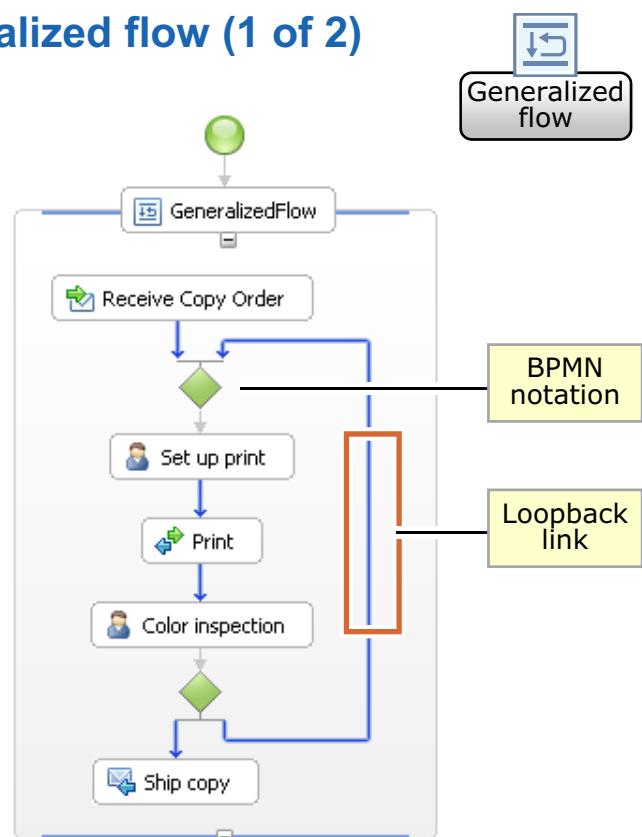
For each is useful in cases where sets of independent data are processed or where independent communication with different partners can be done in parallel. Unlike the parallel activity, the number of parallel branches in a “*for each*” is not known when the process is modeled. Therefore, a “*for each*” behaves like a flow with “n” similar child activities in which the links are constrained. A specified counter variable is used to iterate through a number of parallel branches that a start value and a final value control.

Exit criteria can be specified when the process does not require the completion of all branches (when enough recipients respond or when other reviews are not required because one reviewer rejects the document), for example:

- The author of a document starts a review process human task.
- The reviewers are unknown at authoring time. They are determined during process start through the input message.
- The document review happens in parallel.
- When enough reviews are complete, the “*for each*” is exited.

Structured activities: Generalized flow (1 of 2)

- Generalized flows:
 - Allow easy “backward” links in the process (that go back to a previous step): A loop back condition controls when the “back flow” occurs
 - Can use “catch” or “catch all” fault links instead of fault handlers
 - Support parallel execution paths
- Generalized flows support important modeling scenarios
 - Model arbitrary loop logic (return to a previous activity)
 - Frequently used with human tasks
 - Example: A claims process where you must arbitrarily jump back to redo previous activities



© Copyright IBM Corporation 2016

Figure 9-24. Structured activities: Generalized flow (1 of 2)

WB8601.2

Notes:

Generalized flows are an IBM extension to BPEL. In prior versions, generalized flows were termed cyclic flows.

In this example, the order of a color copy comes in through the “receive copy order” activity. A human task is then assigned to an employee in the “set up print” activity to set up the source file for printing. Part of this setup process includes adjusting the colors so that they are a close match to the original. When this setup process is complete, the flow continues to the “print” activity, which produces the copy. The output is then directed to another human task, the “color inspection” activity. This employee is responsible for deciding whether the colors are a match to the original. If they are not, then the evaluation is set to “false” and the flow is returned to the “set up print” activity. This loop continues until the employee at the “color inspection” activity decides that the color is acceptable, and the process is allowed to progress to the “ship copy” activity.

The generalized flow activity is similar to a parallel activity in that you can nest other process activities within it, and then control the execution order of those activities through links. The generalized flow and parallel activities can also be used to model a business process in a graph-oriented manner. The main difference between the two activities is that the generalized flow

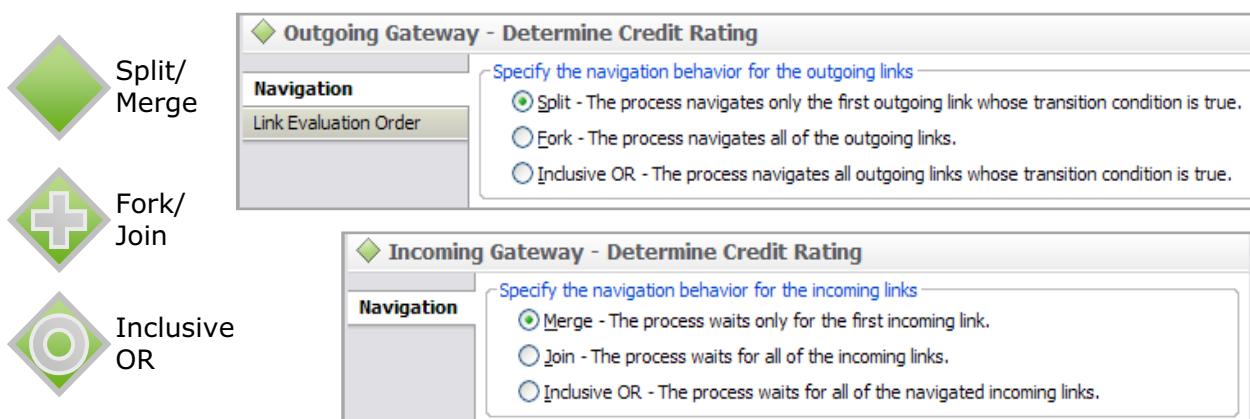
supports cycles (conditional links to loop back to previous activities in the sequence) and fault links. The parallel activity does not.

When faults occur in your business process, fault handlers are typically engaged to deal with the fault. The generalized flow activity offers a simplified “fault handling” procedure. 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, the 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 a “catch” fault link does not cover. If multiple fault links are modeled for the same activity, “best match” decides which fault link to follow. The “fault catching” rules are the same as for fault handlers. No more than one fault link can catch a fault.

You can terminate the fault link at any activity within the generalized flow. For example, you can choose to direct the fault link to a terminate activity to have the business process stop if a fault occurs. You might also 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.

Structured activities: Generalized flow (2 of 2)

- In a generalized flow, gateways are used to define the diverging and converging behavior when more than one link is the target or source of an activity
 - Split/Merge, Fork/Join, and Inclusive OR
- Gateways use Business Process Model and Notation (BPMN) to support IBM Process Designer notation



© Copyright IBM Corporation 2016

Figure 9-25. Structured activities: Generalized flow (2 of 2)

WB8601.2

Notes:

Business Process Model and Notation (BPMN) is a standardized graphical notation for creating diagrams of business processes. You can learn more about BPMN at: <http://www.bpmn.org/>

The visual representation of “graph constructs” in BPEL (parallel activities) means that for each activity with more than one incoming or outgoing link, a (BPMN-like) diamond (or gateway) represents the incoming and outgoing navigation behavior.

Structured activities: Scope

- A **scope** is a behavioral container for one or more activities in your process
 - Encapsulates variables and correlation sets (state)
- By definition, your entire process is contained within a single global scope
- You can nest other scopes within the global scope, forming a hierarchy
- Each scope can have its own:
 - Fault handlers (for expected error processing in the scope)
 - Event handlers (for event processing in the scope)
 - Compensation handlers (for unexpected error processing in the scope)
 - Local variables
- Scopes support dynamic runtime behavior



© Copyright IBM Corporation 2016

Figure 9-26. Structured activities: Scope

WB8601.2

Notes:

Fault handlers, compensation handlers, and event handlers are all associated with a particular scope. Scopes are supported as defined in the BPEL specification. Scopes encapsulate correlation sets and the state of a business process to make particular values available to a specific set of activities. Scopes can be established to define event handlers and compensation handlers.

Scopes can also be nested, allowing variables with the same name to exist in the business process as a unique instance. If a variable is defined at a particular scope, that variable is accessed when called by name. Variables that are defined at a parent scope are available while the name is different. When the scope ends, the variable is no longer available to downstream activities.

Structured activities: Collaboration scope (1 of 2)

- **Collaboration scope** is the preferred tool for the case paradigm
- A “case” is the product of a workflow or part of a workflow
- A case can be any number of things, including:
 - Evaluation of a job application
 - Ruling on an insurance claim
- Any business process can be handled as a case, but it is ideally suited to situations where the task owner uses knowledge and experience to:
 - **Expedite the process:** Close a case without going to trial
 - **Trigger subprocesses:** A doctor orders an extra blood test
 - **Repeat a number of activities:** A second interview
 - **Take an alternative path through the main flow of activities:** An insurance adjuster sees a claimant for added details
- The case model creates workflows that are well-defined but give the staff the flexibility to use their skills and judgment to adapt the flow to business needs
 - These business processes are also called “enhanced dynamic workflows”



© Copyright IBM Corporation 2016

Figure 9-27. Structured activities: Collaboration scope (1 of 2)

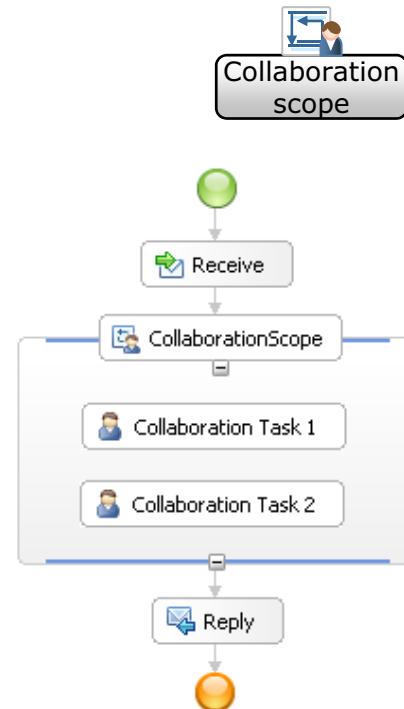
WB8601.2

Notes:

The collaboration scope activity allows enhanced dynamicity for knowledge workers by supporting case handling systems. Case handling systems assist knowledge workers in processing a business case (rather than constraining them). In a “case,” the high-level process is well-defined; however, many exceptions cannot be efficiently predefined. The case does not necessarily run in a straightforward manner, but might require you to jump (redo or skip steps).

Structured activities: Collaboration scope (2 of 2)

- Case handling support is provided by combining dynamicity features and the collaboration scope activity
 - BPEL Process Choreographer Explorer and Business Space support process dynamicity at run time
- You can create business logic within your collaboration scope by adding basic activities
 - You cannot include structured activities in a collaboration scope
- Exit conditions can be used to automatically skip and repeat steps
- You set the administrators for the collaboration scope to specify the people who do manual skips and jumps at run time
 - The default setting is everybody



© Copyright IBM Corporation 2016

Figure 9-28. Structured activities: Collaboration scope (2 of 2)

WB8601.2

Notes:

Use Collaboration scopes to create enhanced dynamic workflows: business processes in which the business logic can be adapted at run time. For example, the assigned worker can decide to repeat an activity, start a subtask, or skip some steps in the business process.

You can configure a scope activity so that users who interact with a runtime instance of the process have administrative authority over the activities that are nested within the scope activity. You can achieve enhanced dynamic behavior more directly by using a “collaboration scope” in which the associated administration task is automatically generated when the collaboration scope is added to the business process.

In the runtime environment, the business process can stop at an activity that is nested within a collaboration scope or a scope for which dynamicity is configured. In that case, an authorized individual can skip, undo, or redo that nested activity. These options are available through a client such as the Business Process Choreographer Explorer. The ability to dynamically modify a process that is already deployed to a runtime environment is especially useful in cases where the process describes a series of steps that are not always necessary.

Unit summary

- List and describe the basic activities for business processes
- Define each of the available structured activities for business processes

© Copyright IBM Corporation 2016

Figure 9-29. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. What is the value of having a generalized flow activity?
2. What is the difference between the parallel activity and the sequence activity?
3. List and define three basic process activities.
4. What type of process paradigm is the collaboration scope designed to support?
5. What is the primary purpose of the “for each” activity?
6. How does a choice activity function?

© Copyright IBM Corporation 2016

Figure 9-30. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Checkpoint answers

1. The generalized flow activity can be used for conditional links to loop back to previous activities in the process.
2. Parallel activities can potentially run activity paths in parallel in interruptible (long-running) processes. Sequences consist of a serialized execution of activities that occur one right after another.
3. Any of the following activities:
 - receive (receive message)
 - reply (respond)
 - invoke (call partner)
 - assign (update variable)
 - wait (pause execution)
 - empty (no-op)
 - empty snippet (Java code)
 - human task (create work items for people)
 - data map (transform business object)
 - receive choice (choose branch, which is based on receive case)
4. The “case” paradigm.
5. The “for each” activity enables the bundling of work.
6. A choice activity selects one activity branch from a set of case elements based on a runtime condition.

© Copyright IBM Corporation 2016

Figure 9-31. Checkpoint answers

WB8601.2

Notes:

Exercise 7

Creating a business process, part II



© Copyright IBM Corporation 2016

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

10.1

Figure 9-32. Exercise 7

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Implement basic BPEL activities in a business process
- Implement structured activities in a business process
- Compare the BPEL to the BPD in IBM Process Designer

© Copyright IBM Corporation 2016

Figure 9-33. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 7

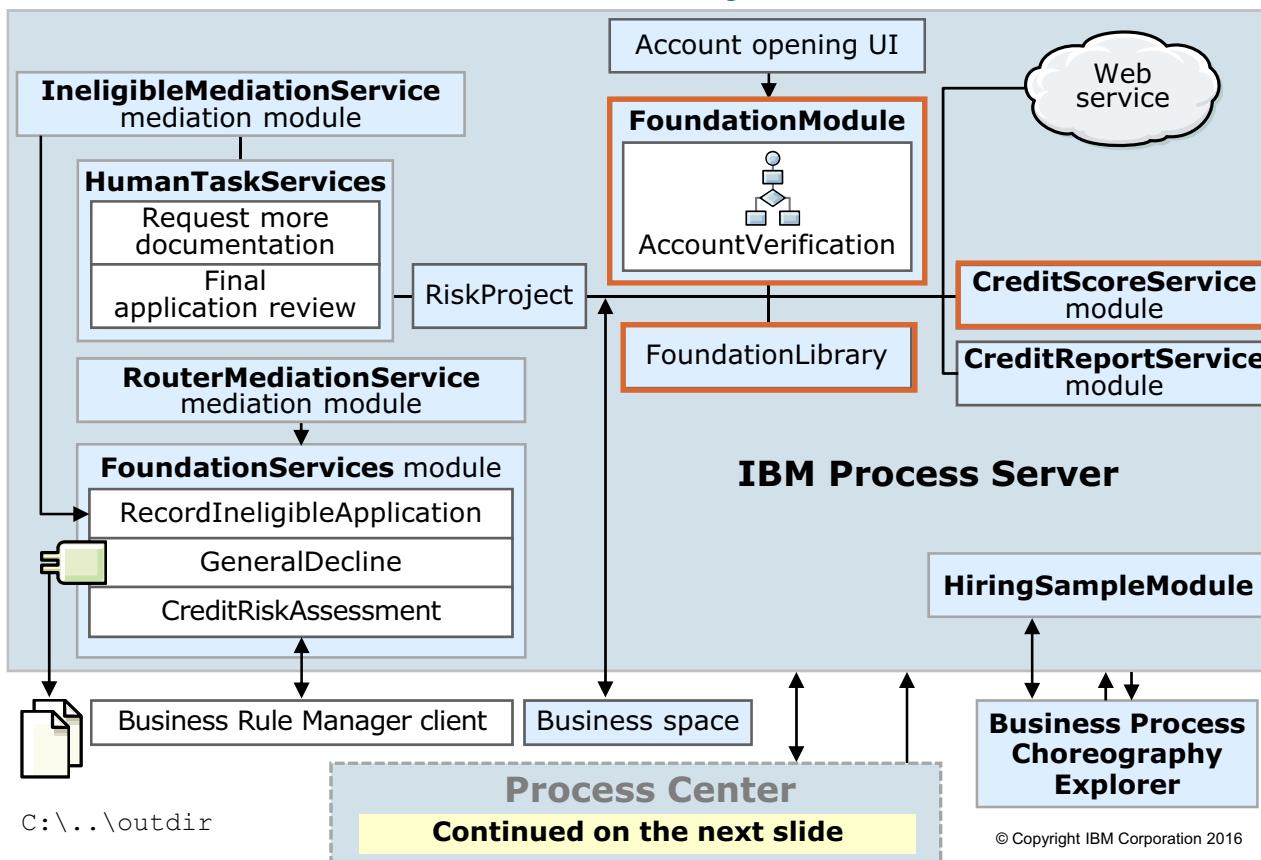


Figure 9-34. Course business scenario: What you work with in Exercise 7

WB8601.2

Notes:

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

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **AccountVerification**
 - BPEL process that you started in Exercise 6
 - You complete building the process in this exercise

New components that you create in the lab:

1. **AccountVerification**
 - BPEL process that you started in Exercise 6
 - You complete building the process in this exercise

Note:

The next slide lists all of the subcomponents of the **AccountVerification** process that are built in this lab.

© Copyright IBM Corporation 2016

Figure 9-35. Components that are required for Exercise 7 (1 of 2)

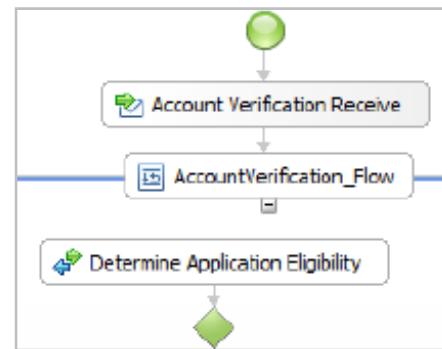
WB8601.2

Notes:

Components that are required for Exercise 7 (2 of 2)

Activities that are created in the **AccountVerification** process:

- Invoke activity
 - **Determine Application Eligibility**
 - **Map to Ineligible**
 - **Record Ineligible Application**
 - **Credit Check Service**
 - **Credit Risk Assessment**
 - **Request More Documentation**
 - **Final Application Review**
 - **Generate Decline**
- Empty action activity
 - **Map to Credit Check**
 - **Map Credit Checking Result**
- Assign and loop activity
 - **Create Output** assign activity
 - **While More Documents Required**
while loop activity



© Copyright IBM Corporation 2016

Figure 9-36. Components that are required for Exercise 7 (2 of 2)

WB8601.2

Notes:

1. You create the first activity: **Determine Application Eligibility** from where the **AccountVerification** process flow starts. The **Determine Application Eligibility** activity examines the customer application to determine whether the customer is eligible for an account. It invokes the **InputCriterion** operation of the **DetermineApplicantEligibility** interface of **DetermineApplicationEligibilityPartner**.
2. The **Map to Ineligible** invoke activity invokes the **InputCriterion** operation of the **MaptoIneligible** interface of **MaptoIneligiblePartner**.
3. The **Record Ineligible Application** activity archives any customer application that is determined to be ineligible. You implement the application that archives the application in a later exercise.
4. The **Map to Credit Check** transforms the data from a customer application business object into a format suitable for the credit score service to understand.
5. The **Credit Check Service** activity invokes the **calculateCreditScore** operation of the **CreditScoreService** interface of **CreditCheckServicePartner**.

6. After the credit score is returned, the data is again transformed. In this case, **Map Credit Checking Result** transforms it back into a customer application business object.
7. The **Credit Risk Assessment** activity examines the credit score that is received and does one of the following three actions:
 - If the credit risk is high, more documentation is requested before final review.
 - If the credit risk is medium, final employee review is requested before approval.
 - If the credit risk is low, the application is approved.
8. The **While More Documents Required** activity continues to request documentation while the comment attribute is equal to None. When an employee reviews the application, the comment attribute is populated and the loop terminates.
9. The **Request More Documentation** activity invokes the **InputCriterion** operation of the **RequestMoreDocumentation** interface of **RequestMoreDocumentationPartner**.
10. The **Final Application Review** activity invokes the **InputCriterion** operation of the **FinalApplicationReview** interface of **FinalApplicationReviewPartner**. This activity invokes a human task that allows an employee to review an application.
11. The **Generate Decline** activity invokes the **InputCriterion** operation of the **GenerateDecline** interface of **GenerateDeclinePartner**.
12. The **Record Declined Application** activity invokes the **InputCriterion** operation of the **RecordDeclinedApplication** interface of **RecordDeclinedApplicationPartner**.
13. The **Create Output** activity sets the process output message to: "Application was approved." The process returns the message when creditRisk is HIGH or MED and the person who reviews the application approves it. This **Create Output** activity sets the process output message to: "Risk was LOW. Application automatically approved." The process returns the message when creditRisk is LOW and the application flows directly from Credit Risk Assessment to Account Verification Reply.

WebSphere Education

Code snippets that are created in the AccountVerification process in Exercise 7

- Determine App Eligibility → Map to Ineligible code snippet
- Determine App Eligibility → Map to Credit Check code snippet
- **Assign Variable code snippet**
- Credit Risk Assessment → Assign Variable code snippet
- Merge Assign code snippet
- Credit Risk Assessment → Final Application Review code snippet
- Final Application Review → Generate Decline code snippet
- Final Application Review → Create Output code snippet
- Credit Risk Assessment → Create Output code snippet

Snippet - Assign Variable (Snippet)

Details

Visual Java

```
if (CustomerApplicationVariable2 == null) {
    com.ibm.websphere.bo.BoFactory factory = (com.
    CustomerApplicationVariable2 = factory.create(
}
CustomerApplicationVariable2=CustomerApplicationVar
```

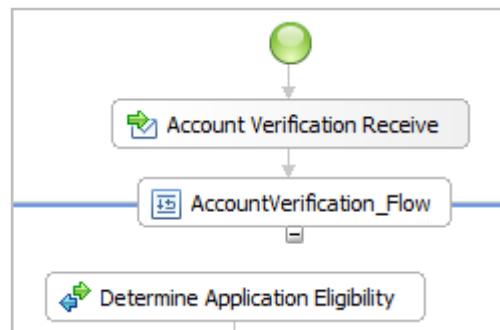
© Copyright IBM Corporation 2016

Figure 9-37. Code snippets that are created in the AccountVerification process in Exercise 7

WB8601.2

Notes:

Complete Account verification process (1 of 4)



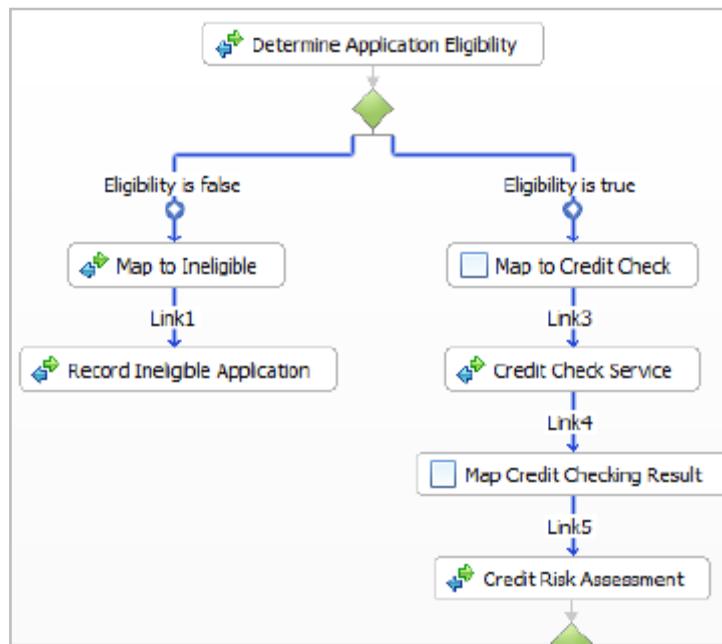
© Copyright IBM Corporation 2016

Figure 9-38. Complete Account verification process (1 of 4)

WB8601.2

Notes:

Complete Account verification process (2 of 4)



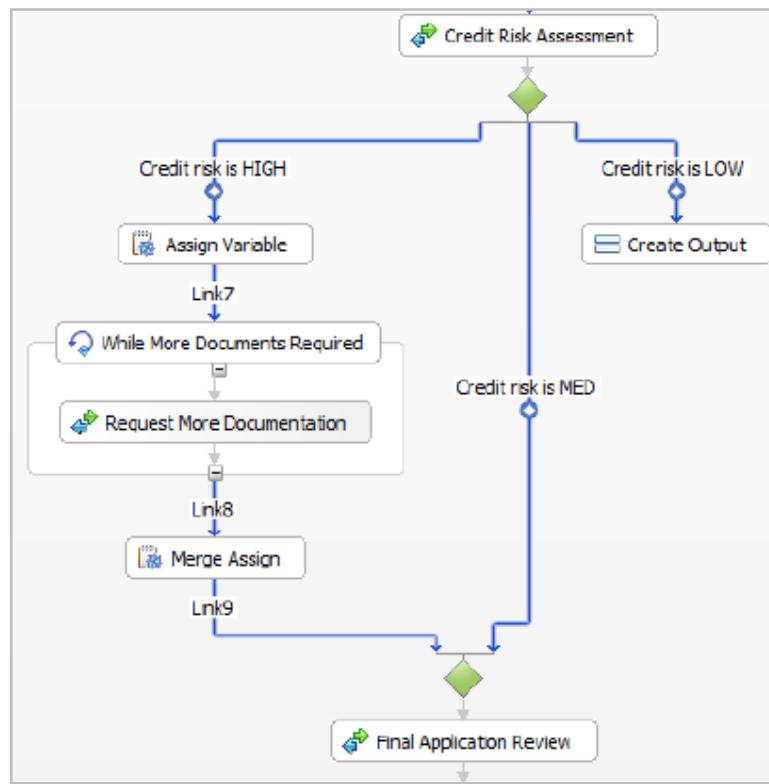
© Copyright IBM Corporation 2016

Figure 9-39. Complete Account verification process (2 of 4)

WB8601.2

Notes:

Complete Account verification process (3 of 4)



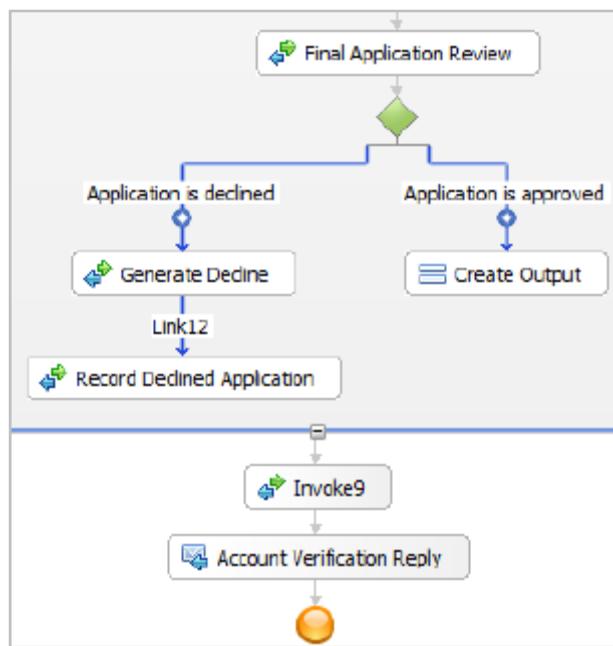
© Copyright IBM Corporation 2016

Figure 9-40. Complete Account verification process (3 of 4)

WB8601.2

Notes:

Complete Account verification process (4 of 4)



© Copyright IBM Corporation 2016

Figure 9-41. Complete Account verification process (4 of 4)

WB8601.2

Notes:



References

- OASIS WS-BPEL Technical Committee
 - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- BPEL4WS and WS-BPEL specifications
 - <http://bpel.xml.org/specifications>
- Business Process Choreographer samples
 - <http://publib.boulder.ibm.com/bpcsamp/v7r5/index.html>
- Versioning and dynamicity with WebSphere Process Server:
 - http://www.ibm.com/developerworks/websphere/library/techarticles/0602_brown/0602_brown.html
- Versioning and dynamicity Part 1: Creating multiple versions of a business process with WebSphere Process Server
 - http://www.ibm.com/developerworks/websphere/library/techarticles/0802_fasbinder/0802_fasbinder.html

© Copyright IBM Corporation 2016

Figure 9-42. References

WB8601.2

Notes:

Unit 10. Business process handlers, runtime behavior, and clients

What this unit is about

This unit describes the various types of activities in BPEL, including basic and structured activities.

What you should be able to do

After completing this unit, you should be able to:

- List and describe the available handlers and error-processing activities
- Describe the runtime behavior of business processes
- Describe the administrative options and types of client access that are available for business processes

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- List and describe the available handlers and error-processing activities
- Describe the runtime behavior of business processes
- Describe the administrative options and types of client access that are available for business processes

© Copyright IBM Corporation 2016

Figure 10-1. Unit objectives

WB8601.2

Notes:



Topics

- WS-BPEL handlers and error processing activities
- Runtime aspects of BPEL processes

© Copyright IBM Corporation 2016

Figure 10-2. Topics

WB8601.2

Notes:

10.1.WS-BPEL handlers and error processing activities

WS-BPEL handlers and error processing activities



© Copyright IBM Corporation 2016

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

10.1

Figure 10-3. WS-BPEL handlers and error processing activities

WB8601.2

Notes:

Handlers and error processing activities: Throw and rethrow

- A throw activity signals a condition that the logic cannot handle (a fault)
- Faults are matched:
 - By fault name alone (the parameter might not match); parameter can contain added fault information
 - By optional parameter alone (not suggested)
 - Or both (best choice for a good match)
- Throws can raise a built-in or user-defined fault
 - The engine throws built-in faults to signal low-level failures in the system
 - BPEL-defined standard faults for common failures (such as joinFailure)
 - User-defined faults that a business partner or process throws
- The rethrow activity takes the current fault in a fault handler and raises it to the next enclosing scope
 - Rethrow raises a new fault with the same name and parameters as the fault that the containing handler caught
 - Indicates that the error state remains
 - Rethrow is available only in fault handlers

© Copyright IBM Corporation 2016

Figure 10-4. Handlers and error processing activities: Throw and rethrow

WB8601.2

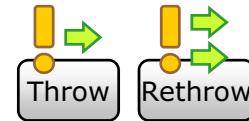
Notes:

A throw activity in a business process can throw any fault, including standard faults, but the intended usage pattern is to throw business faults. A throw activity throws an exception that can be caught and handled within the business process. If a throw activity throws a fault and it is not handled within a process with a request/response interface, either business or standard fault, it is returned as a runtime exception to the process caller.

You cannot return a business or standard fault with a throw activity. You must use a reply activity to return a business fault to the process client. A reply activity can return a business fault that is defined only on the interface that the process implements. This business fault is returned to the process caller; you cannot catch and handle it within the process.

A rethrow activity can be used in a fault handler to rethrow the fault to the next enclosing scope. This action might be useful when you want to do some fault handling on the current scope, such as triggering specific compensation handlers, and still want to make the enclosing scopes aware of this issue.

You can also use a rethrow activity when the current fault handler cannot handle the fault and wants to propagate it to an outer-scoped fault handler. In the absence of a rethrow activity, a fault that uses a throw activity to propagate to a higher level is a new fault. When a rethrow activity is

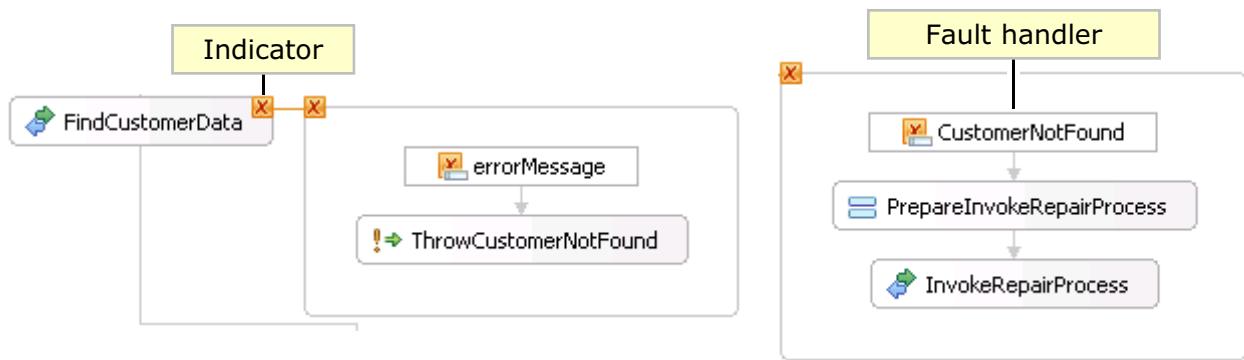


invoked, the fault is the same instance. The rethrow activity is available only within a fault handler because only an existing fault can be rethrown. A rethrow activity is especially useful in a catch-all fault handler because the current fault data cannot be easily retrieved when a throw activity is used.

The Web Services Business Process Execution Language (WS-BPEL) specification defines standard fault types for common system failures. You can add these built-in fault types to your process definition. These faults are available only within BPEL processes, and they do not have an equivalent in the SCA programming model.

Handlers and error processing activities: Fault handler

- A fault handler is a collection of specific activities that run when a fault (expected error) is thrown on the activity or scope 
- Fault handlers use “catch” and “catch all” elements to handle error conditions in a process
 - “Catch” elements process specific faults
 - “Catch all” elements process faults that a “catch” element does not intercept
- A scope can have multiple fault handlers
 - Different kinds of faults can have different fault-handling activities



© Copyright IBM Corporation 2016

Figure 10-5. Handlers and error processing activities: Fault handler

WB8601.2

Notes:

Fault handlers are designed to detect and signal a failure in the execution of a business process. They are designed to catch business exceptions such as a customer who requests an item that is no longer available. The fault handler can catch these faults and either handle the exception and continue processing or rethrow the fault. A fault handler is associated with an invoke activity, a scope of activities, or a process. When a handler catches a fault, any activities in progress are terminated and execution switches to the fault handler.

Fault handlers can be nested like scopes. Each scope can contain its own fault handler. If a fault is not caught at the current scope, it propagates up to the next scope, and the fault handler for that scope catches it. If the fault reaches the uppermost scope of the business process and is not handled, the entire business process fails. Within a fault handler, you can also define compensate activities that trigger a compensation handler.

When a fault is thrown

- A fault means that processing in the current scope cannot continue
 - All active work in the scope fails
- If a matching fault handler is defined in the scope:
 - The scope is in a FAILED state, but the enclosing scope continues with normal processing
 - The fault is handled as specified in the “catch” block
- If a handler is defined but does not match the fault, or if no fault handler is defined:
 - The scope is in a FAILED state, and the fault is rethrown to the next enclosing scope
 - If a fault escalates to the top-level scope of the process, it is considered unhandled, and the entire business process is in a FAILED state
- If a fault is not caught on the enclosing scope or in a fault handler, the “continue on error” setting determines how the process proceeds
 - “No”: The process is STOPPED and a work item is created for the process administrator
 - “Yes”: The fault is rethrown and if it reaches outermost scope, the process is in a FAILED state

© Copyright IBM Corporation 2016

Figure 10-6. When a fault is thrown

WB8601.2

Notes:

A fault handler or fault link can catch a specific fault name, fault type, or both. When a fault occurs, the Business Flow Manager uses the following rules to match the fault with a fault handler or fault link on the enclosing scope, or on the activity where the fault occurred.

- If an invoke activity without a fault handler or any other basic activity is the source of one or more fault links, the Business Flow Manager tries to find a matching fault link. If a fault link is not available, it then tries to find a matching fault handler on the enclosing scope.
- If an invoke activity or a scope with one or more fault handlers is the source of one or more fault links, the Business Flow Manager tries to find a matching fault handler. If a fault handler is not available, it runs the default fault handler and then tries to find a matching fault link. If a matching fault link is not available, it tries to find a matching fault handler on the enclosing scope.
- If the fault does not have any associated fault data, the Business Flow Manager uses a fault handler or fault link with the matching fault name. If a fault handler or fault link is not found, it uses the catch-all fault handler or fault link if one is available. A fault handler or fault link with a fault variable that is defined cannot catch a fault that does not have any data.

- If the fault has fault data that is associated, the Business Flow Manager uses a fault handler or fault link with the matching fault name. It also uses a fault variable with a type that matches the type of the fault data. If a fault handler or fault link is not found that matches the name and fault data type, it uses a fault handler or fault link without a fault name. It also uses a fault variable with a type that matches the type of the fault data. If a suitable fault handler or fault link cannot be found, it uses the catch-all fault handler or fault link if one is available. If a fault handler or fault link does not have a fault variable that is defined, it cannot catch a fault that has data.

If a fault is raised that does not match any of the fault handler or fault link definitions, the default fault handler is started. The default fault handler is not specified explicitly. The default fault handler runs all of the available compensation handlers for the immediately enclosing scopes in the reverse order of completion of the corresponding scopes. If the scope is the source of one or more fault links, the Business Flow Manager then tries to find a matching fault link. If a matching fault link is not available or the scope is not the source of any fault links, the default fault handler rethrows the fault to the next level. The next level is the enclosing scope of the process. On this next level, the Business Flow Manager again tries to match the fault to the fault handlers or fault links that are available.

If any of the specific fault handlers or fault links do not catch the fault – for example, the catch-all fault handler or catch-all fault link – the fault reaches the process scope. The process ends in the FAILED state. Even if a fault handler catches the fault on the process scope and handles it, the process still ends in the FAILED state.

When you define a business process, you can specify what happens when an unexpected fault is raised and a fault handler is not defined for that fault. You can use the “continue on error” setting when you define your process to specify that it is to stop where the fault occurs.

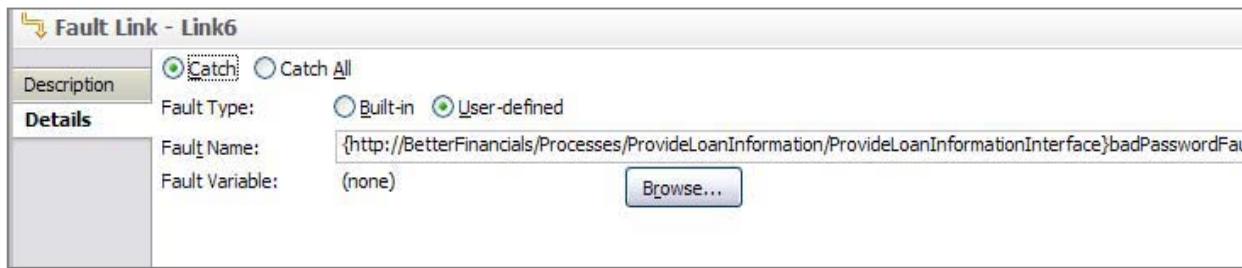
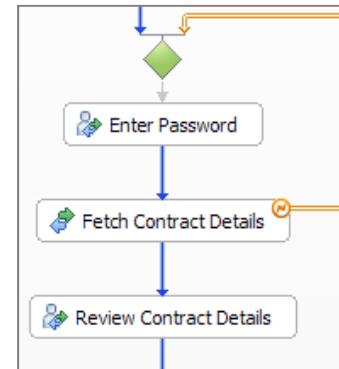
For most activities, the continue on error behavior is the same as for the process. You can specify the continue on error behavior explicitly for invoke, Java snippet, custom, and human task activities. By default, the continue on error behavior of these activities is also the same as for the process.

If an unexpected fault is detected, fault handling of the activity is started. If the continue on error setting is set to “yes,” then the standard fault handling is applied. The continue on error setting for the activity or the process might be set to “no.” In this case, if a fault handler on the immediately enclosing scope or a fault link that leaves this activity does not handle the fault, then the activity stops. If a catch-all fault link or fault handler is defined on the immediately enclosing scope, the value of the continue on error setting has no effect. The fault is always handled, and the activity is never stopped.

For activities that stop because of an unexpected fault, you can use the stop reason property of the activity, which determines the cause of the fault and the actions that you can take to repair it.

Error processing: Fault links

- Fault links are run after a fault occurs in the source activity of the link
 - When a fault occurs and is caught, only the fault link is navigated
 - The fault path can merge back into the regular flow
 - Evaluation order of multiple fault links can be specified
- Fault catching rules are the same as for fault handlers
 - A fault name and fault variable to catch can be specified (Catch), or all faults can be caught (Catch All)
- Fault links are available in generalized flows



© Copyright IBM Corporation 2016

Figure 10-7. Error processing: Fault links

WB8601.2

Notes:

When faults occur in your business process, fault handlers are typically engaged to deal with the fault. The generalized flow activity offers a simplified fault handling procedure. 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, the 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. The catch-all fault link must be followed when any fault occurs that a catch fault link does not cover. If multiple fault links are modeled for the same activity, best match decides which fault link to follow. The fault that catches rules is the same as for fault handlers. No more than one fault link can catch an occurred fault.

You can terminate the fault link at any activity within the generalized flow. For example, you can choose 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.

Handlers and error processing activities: Compensate

- Compensation activities allow business processes to define activities that place the business back into a balanced state
 - A balanced state is an outcome acceptable to both the client and the business
 - Only scopes or activities that completed normally can be compensated
- The runtime behavior and objective of compensation differ:
 - Compensation in a long-running process attempts to restore balance after one or more transactions are committed
 - In a microflow, the entire process runs within a single transaction
 - To set up compensation for a microflow, you store the properties for the invoke activities
 - If the process cannot be committed and must be rolled back, the original data can then be restored
 - If a microflow fails, the runtime engine crawls back through the invoke activities in reverse order, restoring each to its previous state
 - To create compensation logic for an invoke activity in a microflow, you must define a location to store the process state details, called a compensation sphere
- The compensate activity is available in a fault handler, a compensation handler, a collaboration scope, or a generalized flow activity

© Copyright IBM Corporation 2016

Figure 10-8. Handlers and error processing activities: Compensate

WB8601.2

Notes:

The compensate activity can be used in a fault handler or a compensation handler. The compensate activity points to a specific compensation handler, which contains the steps necessary to restore a process to a balanced state. In addition, compensation activities can be added to a generalized flow activity or a collaboration scope.

Previously, compensate activities were used only in compensation handlers and fault handlers (as specified by the BPEL specification). When a compensate activity was added to a generalized flow or collaboration scope, the entire generalized flow or collaboration scope was compensated; you could not specify a specific compensation target in these activities.

Compensation has at times been described as a means of undoing an action, but this description is not entirely accurate. More specifically, it is a service that runs when a state is reached in your process that you deem undesirable. The goal is not always to return to a previous condition, but instead to maintain a balanced and consistent state and to compensate any committed operations that conflict with this state. Two types of compensation exist: business compensation and technical compensation.

- **Business compensation:** This type of compensation is used in a transactional process where an operation is already committed and cannot be reversed. Business compensation is another operation that, when run, creates a balanced state where both business partners are satisfied.

For example, suppose that something goes wrong at any time during a typical business transaction. It is a simple matter of replacing the object on the store shelf, and halting all communication between the purchaser and the vendor. However, if the transaction is committed (money is exchanged and a receipt issued), then canceling it is not possible. You cannot return the object to the shelf. A different procedure (a refund) must take place to return the conditions to a balanced state. The operations that already took place must be compensated to return to a situation in which both partners are satisfied. It is not necessarily the same state that existed before (for example, if the customer paid in cash, the customer can receive a store credit in return). Nonetheless, it is one that is balanced and consistent.

- **Technical compensation:** Technical compensation is used in transactions that fail before they are committed, when one of the operations cannot be reversed. For example, imagine that a customer requests that an item is personalized in some way. The vendor complies, but before money is exchanged, something unexpected happens, and the transaction is canceled. The object cannot be returned to the shelf; another procedure must run to consider the personalization that took place. In another example, imagine that in your process, one of the activities within a transaction sent out an email, but the transaction was canceled before it was committed. You cannot undo the sending of an email, so you must compensate in some other way.

The two ways in which you can compensate business processes are:

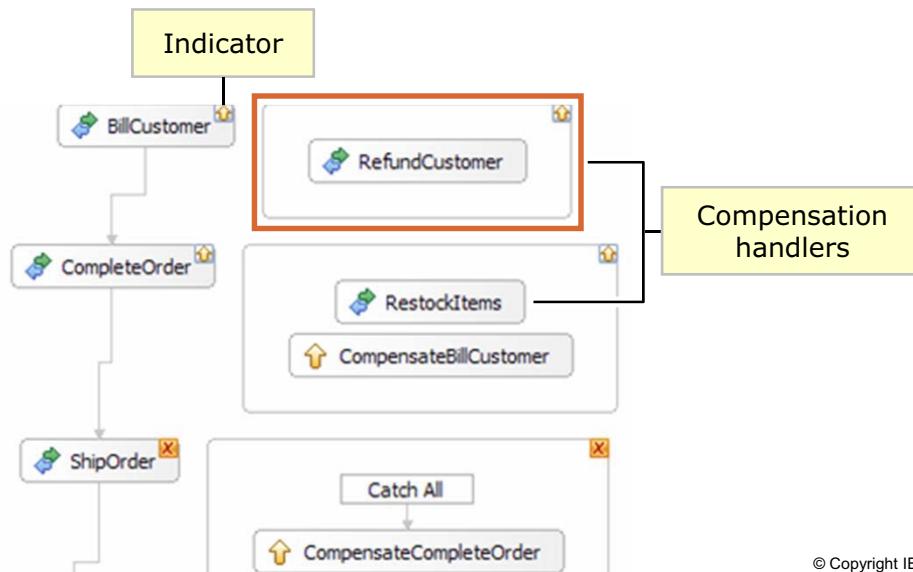
- **Compensation pairs:** Compensation pairs are the original properties of each of the individual parts of a business process. These properties are saved so that they can be restored if the process cannot be committed and must be rolled back. The original status of the activity is stored in an operation, and its value is stored in a variable. If IBM extensions are not enabled for this process, then you cannot use compensation pairs and must use a compensation handler.
- **Compensation handler:** A compensation handler is a series of isolated activities. The activities in the handler run only when a fault is thrown, and after the parent activity is already committed. The goal of a compensation handler is to return a failed process to a balanced state.

You cannot use a compensation handler with a microflow. Because microflows run within a single transaction, you must use compensation pairs to store the original properties of each invoke activity if the process fails.

In a long-running process, you can use either of these options. If the compensation characteristics of each activity are fairly simple (compensation can be achieved in a single step), then consider the use of compensation pairs. However, if you require compensation that uses more complicated logic, assign a compensation handler to each activity.

Handlers and error processing activities: Compensation handlers

- Compensation handlers contain actions that reverse operations that are designed to handle unexpected system problems and bring the process back to a known, stable state
- Compensation handlers can be added to invoke, scope, generalized flow, collaboration scope, or human task activities



© Copyright IBM Corporation 2016

Figure 10-9. Handlers and error processing activities: Compensation handlers

WB8601.2

Notes:

In this example, if an order fails to ship for some reason, a fault is thrown and is caught by using a “catch-all” block in a fault handler. The compensation activity in the fault handler then calls the appropriate compensation handler. This handler restocks the items on the order. The next compensation handler is then called, which would handle refunding the customer. Although the compensation handlers in this example are attached to individual activities, they can be attached to scopes to include multiple activities. During compensation:

- All variables and partner links are accessible because all variable values are persisted in the database when the completed activity commits. Each completed activity creates a row in the database that contains a snapshot of the process variable values.
- The values (such as a confirmation number) can be used in the compensation handler to “undo” the completed activity.

For a detailed discussion of compensation and business processes, see course WB858: *Developing Applications in IBM Business Process Manager V8.5.5 Advanced – II*, which is the second week of training for the version 8.5.5 product.

Handlers and error processing activities: Event handlers

- **Event handlers** accept outside requests for a particular scope
 - **OnEvent** elements allow the process to consume messages from the client
 - **Timeout** elements allow the process to work on a task after a specified length of time (a timer starts when a scope is entered)
- Any type of processing can occur in the event handler (sequence or parallel)
- Requests can be one-way or request/response, and events can be repeated until the activity or scope is completed
- Restrictions can be set to limit who can send events
- Correlation sets must be enabled on event handlers to direct events to the correct business process instances

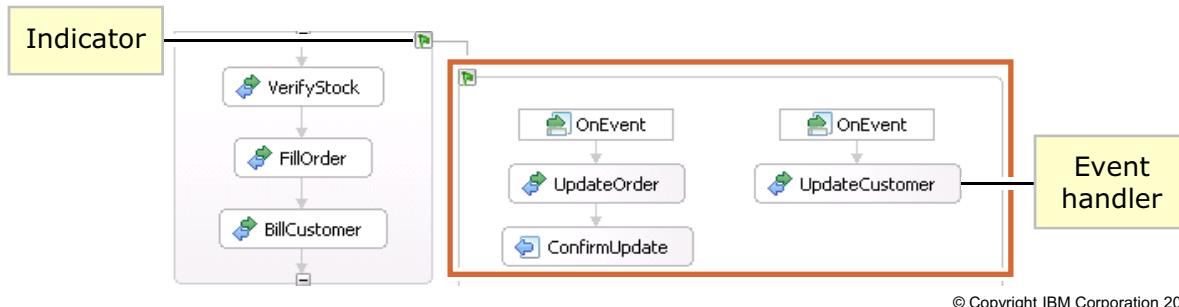


Figure 10-10. Handlers and error processing activities: Event handlers

WB8601.2

Notes:

Event handlers were originally defined under the BPEL V1.1 specification. Event handlers are available only for a long-running business process. They accept outside requests from external clients into the business process.

In this example, while the specified scope is active, a customer can add or remove products from the order or update the customer information.

You can define and configure an event handler either on individual scopes within the process, or for the process in its entirety. An event handler can be used to process requests while the process is running. To process the event, the handler has access to business process state information such as variables.

An event handler has two parts. The first part defines the conditions under which an event handler is invoked, and it defines the type of the event handler. You have the following two choices:

- **OnEvent** element: Use this element to create a control path and specify the operation that causes this path to be followed.
- **Timeout** element: Use this element to create a control path that is followed when a specified time is either reached or elapsed. This element is used on a single path, and is configured to

specify a specific date or time. When the process is running, this path is chosen unless input is received within this time period, or by the specified date.

The second part of an event handler defines the implementation of the event handler; specifically, the action that the event handler is to take when it runs. To specify this action, populate that handler with the necessary process activities from the palette.

Event handlers use correlation to match requests to the correct process instance. When a handler receives an event, a set of steps that is defined in the handler runs in a separate thread from the business process. When the event handler activities are complete, the thread stops.

Each event handler is enabled when the process or scope that it is associated with starts. Each event handler is disabled when the process or scope that it is associated with ends. Multiple instances of each event handler can be started while the event handler is enabled. If an event comes in before this event handler is available (when the scope is reached), the business process engine holds that event and waits until that scope is reached. If the event is received after the scope is ended, a runtime exception is generated, indicating that the event cannot be handled.

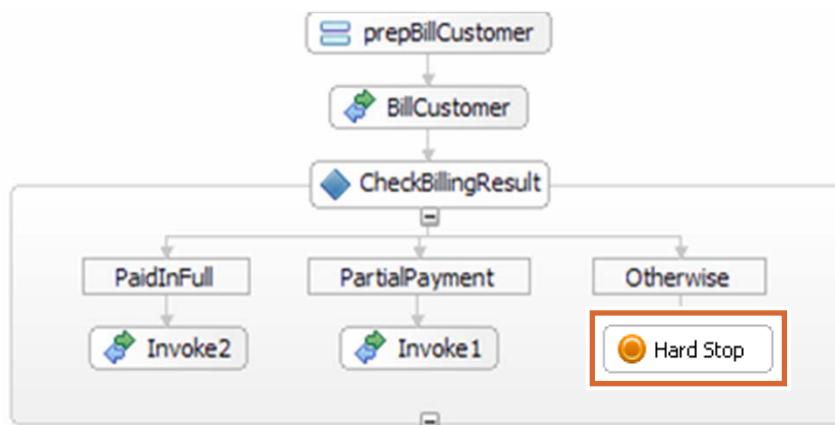
The event activity runs on its own thread, created by the process container upon dispatching the event.

Handlers and error processing activities: Terminate

- A **terminate** activity ends processing of the business process instance (a hard stop) or the structured activity
 - Can be used to allow the administrator to make repairs
- All execution ends, including parallel execution paths
- Compensation of terminated activities does not occur



Terminate



© Copyright IBM Corporation 2016

Figure 10-11. Handlers and error processing activities: Terminate

WB8601.2

Notes:

The terminate activity is the equivalent of pulling the plug on an electrical device. It is a hard stop, signifying that the business process must be halted. When it is used, all activities that are currently active are halted without any fault handling or compensation behavior.

10.2.Runtime aspects of BPEL processes

Runtime aspects of BPEL processes



© Copyright IBM Corporation 2016

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

10.1

Figure 10-12. Runtime aspects of BPEL processes

WB8601.2

Notes:



Business process runtime support

- Service integration buses support process communication and navigation
 - Java Message Service (JMS) messaging resources for your business processes are created during application deployment
 - Process transactions can also be triggered by using the work manager
 - In work-manager-based navigation, work-manager threads are used to reduce JMS use and improve performance
- Process state and activity-related information can be stored in a database for persistence, recovery, and data queries
 - Database is called BPEDB by default
 - Database must be created manually in IBM Process Server, by using a supported production database, such as DB2 or Oracle
 - Database creation scripts are available at `<WPS_Install>\dbscripts\ProcessChoreographer`
 - Database creation is automatic in IBM Integration Designer
 - DB2 Express is the data store for IBM Integration Designer
- If your business process uses human tasks, you must also install the human task container

© Copyright IBM Corporation 2016

Figure 10-13. Business process runtime support

WB8601.2

Notes:

The business process container is the resource that is responsible for running the business processes. It is a Java EE application front end that provides the APIs that you can use to create robust clients. The business process container can be installed by default when you install IBM Process Server. It is automatically installed with IBM Integration Designer as part of the test environment.

If you elect not to install the container when you install a stand-alone IBM Process Server, you can configure it later by using the IBM Process Server administrative console. The administrative console includes a wizard that guides you through the container configuration options such as the database type, messaging resources, and security parameters.

In IBM Integration Designer, the container is configured to use a Derby database by default, which is adequate for development and testing. A scalable, enterprise database such as DB2 is suggested for production environments. DB2 on z/OS, Oracle, and a number of other database products are supported. Scripts are provided to set up the database with the tables that the business process container needs. The data store holds the business process templates, along with business process instance and state information for the container. Messaging resources for business processes are enabled and configured in the environment as well.

A lightweight navigation style, called work-manager-based navigation, is also available. In work-manager-based navigation, work-manager threads are used. The quality of service is the same as in JMS navigation. Communication to the database is reduced, which results in performance improvements. For more information about work-manager-based transactions in business processes, see the topic “Improving the performance of business process navigation” in the product documentation.

Business process templates and instances

- Process templates describe the business process model
 - The Business Flow Manager uses the template to create instances of the business process at run time
 - Process templates are deployed and installed on IBM Process Server
- Process instances are entities that exist at run time
 - An instance represents one running business process, its specific data, and its state
 - The Business Flow Manager can run multiple process instances at the same time
- Creation of a template version is supported by using the template's valid-from date
 - The valid-from date is used to decide which process template to use when creating a process instance
 - When the instance is created, it runs against that version of the template
 - If a new template is deployed, in-flight process instances can be upgraded
- Process templates and instances are persisted to a database
 - In IBM Integration Designer, the database is in DB2 Express
 - For production environments, database creation scripts are provided per platform

© Copyright IBM Corporation 2016

Figure 10-14. Business process templates and instances

WB8601.2

Notes:

By using the process instance migration, you can migrate in-flight process instances to the new version at run time. To create a version of your process, you can define a process migration specification, which provides process instance migrations at run time. You deploy and install the new process version (with a new validFrom date) and the process migration specification to your runtime environment. An administrator can then migrate existing instances of the process in the runtime environment to the new version of the process. The processes can be migrated in two ways: the BPEL Process Choreographer Explorer application for select business processes, or a script for batch migration.

Of critical importance: The two versions must have the same name and namespace; but with different valid-from dates, the correlation sets for the process versions must be the same, and the interfaces must be the same. By using the process instance migration tools, you can update versions of running instances of processes in a late-binding situation. With early binding, a client is hardwired to a particular process version.

For an example of process instance migration, see:

<http://publib.boulder.ibm.com/bpcsamp/index.html>

Microflow and long-running processes at run time

- The runtime behavior of a business process depends upon whether it is long-running or microflow (non-interruptible)

	Microflow (non-interruptible)	Long-running (interruptible)
Transactions	One transaction for the entire process	Transaction boundaries can be set between activities
Persistence	None	Information about process activity and state is persisted to a database
Crash recovery	None (execution is transient; if the server crashes, the process state is lost)	Process can resume from the last checkpoint
Parallelism	None: strictly single-threaded	Activities can run in parallel
Interruptible	No	Yes
Asynchronous	No	Yes
Correlation	No	Yes
Handlers	No support for fault, event, or compensation handlers	Handlers are supported

© Copyright IBM Corporation 2016

Figure 10-15. Microflow and long-running processes at run time

WB8601.2

Notes:

In this course, transactions and persistence are not covered in detail. However, it is still necessary to point out these key differences between microflow and long-running business processes. While they are both considered business processes, a microflow is contained within a single transaction, where a long-running process can be multi-transactional and can run over an extended period. If your process requires more than one transaction, it must be a long-running process.

If your process can be designed either way, consider the following options:

- If your process must stop at any point and wait for external input, in the form of either an event or a human task, then you must use a long-running process. Microflows are not interruptible.
- If you do not have IBM extensions that are enabled for this process, then you cannot make it a microflow. A microflow is an IBM enhancement of the BPEL programming language. If these extensions are disabled, a microflow is not an option.
- If you have a short series of steps to model and want them to run quickly in the runtime environment, then use a microflow. Microflows can also be used as subprocesses in a larger business process.

- If you have elements in your process that you would like to run in parallel, use a long-running process. Keep in mind that to work properly, each of the parallel paths must run within its own transaction, and its transactional behavior must be set to be either “commit before” or “requires own.”

For a detailed discussion of transactions and business processes, see course WB858: *Developing Applications in IBM Business Process Manager V8.5.5 Advanced – II*, which is the second week of training for the version 8.5.5 product.

If IBM extensions are disabled, the **Process is long-running** check box is not displayed on the Details tab of a long-running process. However, you can enable the extensions by clearing the **Disable IBM Process Server BPEL extensions** check box on the Details tab. When you do so, you are warned that this step cannot be undone. If you proceed, then the **Process is long-running** check box is displayed on the **Details** tab. It can be cleared to convert the process to a microflow.

Clients for business processes

- Several clients are available for working with process instances
- Business Process Choreographer Explorer:
 - Is a customizable application for running basic administrative tasks on business processes
 - Reporting is a customizable application for reporting on business processes
- Generic EJB, JMS, and web service APIs are used to create clients that make direct calls to the business process container
 - Can query business process instances
 - Can interact with instances (start, send events, and invoke activities)
 - APIs are documented in the IBM Integration Designer product documentation
- Clients interact with processes by using different service interfaces and bindings:
 - SCA, JMS, and web services
 - Clients start processes (through a receive or receive choice activity) or send events

© Copyright IBM Corporation 2016

Figure 10-16. Clients for business processes

WB8601.2

Notes:

The BPEL Process Choreographer Explorer reporting component requires the Common Event Infrastructure EVENT database to compile statistics.

The Business Process Choreographer web services API provides two separate web service interfaces:

- The Business Flow Manager API allows client applications to interact with microflows and long-running processes, for example:
 - Create process templates and process instances
 - Claim existing processes
 - Query a process by its ID
- The Human Task Manager API allows client applications to:
 - Create and start tasks
 - Claim existing tasks
 - Complete tasks

- Query a task by its ID
- Query a collection of tasks

Client applications can use either or both of the web service interfaces.

The Enterprise JavaBeans (EJB) APIs are provided as two stateless session enterprise beans. Business process applications and task applications access the appropriate session enterprise bean through the home interface of the bean.

The `BusinessFlowManagerService` interface provides the methods for business process applications, and the `HumanTaskManagerService` interface provides the methods for task-based applications. The application can be any Java application, including another Enterprise JavaBeans (EJB) application.

With these Enterprise JavaBeans (EJB) APIs, you can create client applications to do the following tasks:

- Manage the lifecycle of processes and tasks, from starting them to deleting them when they complete
- Repair activities and processes
- Manage and distribute the workload over members of a workgroup

The EJB APIs are provided as two stateless session enterprise beans:

- The `BusinessFlowManagerService` interface provides the methods for business process applications.
- The `HumanTaskManagerService` interface provides the methods for task-based applications.

For more information about the EJB APIs, see the Javadoc in the `com.ibm.bpe.api` package and the `com.ibm.task.api` package.

You can use the generic JMS client interface (referred to as the “JMS API”) to develop client applications that asynchronously access business processes that run in the Business Process Choreographer environment. The JMS API allows client applications to asynchronously interact with microflows and long-running processes. The JMS API exposes the same interface as the web services API, with the following exceptions:

- With the web services API, the call operation can be used to invoke microflows only.
- However, by using the JMS API, the call operation can be used to invoke both microflows and long-running processes.

All Tasks

Use this page to work with task instances for which you have access rights.

Priority	Task Name	State	Kind	Owner
5	Request More Documentation	Ready	To-do Task	

Items found: 1 Items selected: 0 << Page 1 of 1 >>

- Business Process Choreographer Explorer is included for basic process administration
- Business Explorer is built with reusable JavaServer Faces (JSF) components

© Copyright IBM Corporation 2016

Figure 10-17. Business Process Choreographer Explorer client

WB8601.2

Notes:

Business Process Choreographer Explorer is included for basic process administration, as follows:

- View installed process templates
- View, start, terminate, delete, compensate, suspend (including “suspend until” and “suspend for”), try process instances again, transfer ownership

Business Process Choreographer Explorer is built with reusable JavaServer Faces (JSF) components, as follows:

- Starting point for customized administrative clients
- Ability to create customized views

The Business Process Choreographer Explorer client provides administrative capabilities for business processes and human tasks. It is installed when you configure the business process container. You can do basic functions such as starting, terminating, and deleting instances. You can compensate activities that fail within the business process, and you can try them again. Security can be enabled for this application to limit who can work with or administer processes and tasks.

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 the Business Process Choreographer Explorer.

The Business Process Choreographer Explorer includes support for selective deletion of completed process instances and their associated data from the Business Process Choreographer runtime database.

Deletion criteria include:

- Process instances of 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 – that is, finished, terminated, or failed

You can also use the Business Process Choreographer Explorer to suspend process instances for an amount of time or until a specified time and then automatically resume them. For example, if a client asks to put an order on hold, you can suspend the process for seven days, or until a specific date and time.

Suspending a business process instance means that the navigation of the process instance is put on hold. If you suspend a process instance, you can specify a duration or point in time when the process instance resumes automatically. This capability is exposed through all renderings of the generic BFM interface, that is, to JMS clients, web services clients, and EJB clients. Suspension of a process applies to long-running, top-level processes, and is propagated to all child processes.

You can also use the Business Process Choreographer Explorer to transfer process ownership. For example, if the owner of a process is no longer with your company, you can use this feature of the Business Process Choreographer Explorer to transfer process ownership to another employee.

In the IBM Integration Designer test environment, the Business Process Choreographer Explorer URL is: <http://localhost:9080/bpc>



Runtime process dynamicity (1 of 2)

- Scopes and collaboration scopes support dynamic modification by using the process widget in Business Space or Business Process Choreographer Explorer
 - Provides greater flexibility and allows processes to adapt to changing situations
 - Supports dynamic human workflow scenarios and case handling scenarios
- Dynamicity allows business users and solution administrators to override the navigation of a long-running process
 - Jump forward and backward between activities in a running process
 - Skip specific activities within a running process
 - Incorporate process-relevant data changes at run time

Skip Activities

Use this page to skip activities and, optionally, to view and modify the variables

Dynamically skip activities and set variables

Skip	Set Variables	Cancel
Process Instance Name	LoanApp2	
Description		
State	Running	
Activity Names	Receive CopyInput CheckAutoApproval	

© Copyright IBM Corporation 2016

Figure 10-18. Runtime process dynamicity (1 of 2)

WB8601.2

Notes:

Dynamic modification includes the ability to modify process data at run time. You can modify business information, such as customer name and address. CEI events can be generated during dynamic modification for auditing purposes.

Runtime process dynamicity (2 of 2)

- Jumping (forward and back) is supported between activities in a generalized flow, sequence, or single thread or branch of a parallel flow
 - Source of a jump is a basic activity
 - Target of a jump can be a basic or structured activity
 - Jumping into nested constructs is not supported
- You can skip active or future basic activities in a process
 - Can immediately skip active activities (in a non-terminal state: running or ready)
 - Future activities can be marked to be skipped
 - Can skip basic activities only
 - Cannot skip structured activities (such as scope or sequence)
 - Can combine skip with jump
- If the activity is not reached, skipping an activity can be undone
- Specific business users and administrators can be authorized to initiate jumps, or to skip certain activities in a scope or collaboration scope
- You can update process variables at run time
 - Can use BPEL Explorer or Business Space to set variables for running instance
 - Useful in repair scenarios where inconsistent data must be changed

© Copyright IBM Corporation 2016

Figure 10-19. Runtime process dynamicity (2 of 2)

WB8601.2

Notes:

For a complete list of repairable items in business processes, see the product documentation.

Installing and uninstalling business process applications

- In the IBM Integration Designer test environment:
 - Server runs in “development mode” by default
 - Applications are installed and uninstalled easily by using add or remove projects
 - Removing applications does not require terminating and deleting instances, or stopping process templates
- On a production server:
 - Not necessary to stop a process template or task template before uninstalling an application; the template is stopped automatically
 - If uninstallation fails, the template is restarted
 - If instances are running, uninstallation fails
 - A check is done after the template is stopped
 - Applications can be removed by using a Jacl script even if instances exist in various states (it cleans up templates and instances, but should be used only during testing)

General Properties	
Name	server1
Node Name	widNode
<input checked="" type="checkbox"/> Run in development mode	

© Copyright IBM Corporation 2016

Figure 10-20. Installing and uninstalling business process applications

WB8601.2

Notes:

Business processes, as deployable artifacts, are contained within an EAR file and exposed as SCA components. They are installed in the normal application installation process. No more steps are necessary when installing an application that includes a business process. If you uninstall an application that contains a business process, you must first stop all running instances of that business process. This check is done to prevent 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 are removed.
2. Stop the template that represents that business process. (It is done automatically when the administrative console is used to uninstall the application.)
3. Uninstall the application.

This process is made easier in the development environment by using the “Run in development mode” option. By using this option, you can uninstall and update the application without having to terminate all business process instances. You can enable this feature for testing in the IBM Integration Designer environment, but it is not necessary to enable it in a production environment.

By using the `bpcTemplate.jacl` script, you can use the `-force` flag to uninstall applications that comprise processes or tasks with running instances. It is not meant for production environments except as a last resort. Removing applications along with running instances must be done during testing only.

Deletion of completed instances

- Deletion of completed instances as specified in process or task properties
 - **Yes:** Delete an instance when in a FINISHED, TERMINATED, or FAILED state
 - **On successful completion** (default): Delete only if not in FAILED state
 - **Completed after:** Delete instances that are complete after the specified date or time
 - **No:** Do not delete
- Cleanup service allows scheduled deletion of instances and tasks
 - Specify administratively which instances should be deleted and when
- Cleanup service configuration specifies:
 - When and how long (in minutes) the cleanup service should run
 - How many instances to delete in one transaction
 - Which instances should be deleted
- Cleanup job configuration specifies:
 - Instances that are going to be deleted
 - Time instances are kept after completion before deletion by the cleanup service
 - Jobs that run in the order that is listed

© Copyright IBM Corporation 2016

Figure 10-21. Deletion of completed instances

WB8601.2

Notes:

The following are many ways to delete process instances:

- Deletion of completed instances can be specified when defining business processes in IBM Integration Designer.
- Completed process instances can be deleted administratively either by using the `deleteCompletedProcessInstances` script, or by using the corresponding MBean interface with the `wsadmin` command-line tool.
- Administrators can use the BPEL Process Choreographer Explorer to delete selected instances (not ideal for handling large numbers of instances).
- Business Process Choreographer APIs (EJB, JMS, web services, REST) can be used to delete instances.
- You can use the cleanup service to delete instances.

Options for automatic deletion in IBM Integration Designer include:

- Do not automatically delete the process instance upon completion.
- Delete the process instance upon its completion (state: FINISHED, FAILED, or TERMINATED).
- Delete the process instance only if it completed successfully (state: FINISHED).



Unit summary

- List and describe the available handlers and error-processing activities
- Describe the runtime behavior of business processes
- Describe the administrative options and types of client access that are available for business processes

© Copyright IBM Corporation 2016

Figure 10-22. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. What is a business process template?
2. What client would you use to view information about the state of running processes?
3. List three differences between the runtime behavior of a long-running process and a microflow.
4. What is the purpose of the cleanup service?
5. What is the purpose of the fault handler?
6. What is the purpose of a compensation handler?

© Copyright IBM Corporation 2016

Figure 10-23. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Checkpoint answers

1. Process templates describe the process model. The Business Flow Manager uses the template to create instances of the process.
2. BPEL Process Choreographer Explorer or Business Space.
3. Microflows do not persist. No crash recovery is available for microflows. Microflows do not support parallelism. Microflows cannot be interrupted, are synchronous, and do not support correlation.
4. It allows the scheduled deletion of process instances and tasks administratively so you do not have to specify deletion behavior in the application.
5. A fault handler is a collection of specific activities that run when a fault (expected error) is thrown on the activity or scope. Fault handlers use catch and catch all elements to process error conditions in a process.
6. If an unexpected problem occurs, compensation handlers provide rollback capability for long-running processes and consist of all the steps that must be taken to restore the process to a balanced state.

© Copyright IBM Corporation 2016

Figure 10-24. Checkpoint answers

WB8601.2

Notes:

Exercise 8

Creating a business process, part III



© Copyright IBM Corporation 2016

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

10.1

Figure 10-25. Exercise 8

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Use data maps to transform process data
- Use context variables to create a runtime process description
- Assemble an SCA application that contains a business process
- Test a business process in the IBM Integration Designer test environment

© Copyright IBM Corporation 2016

Figure 10-26. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 8

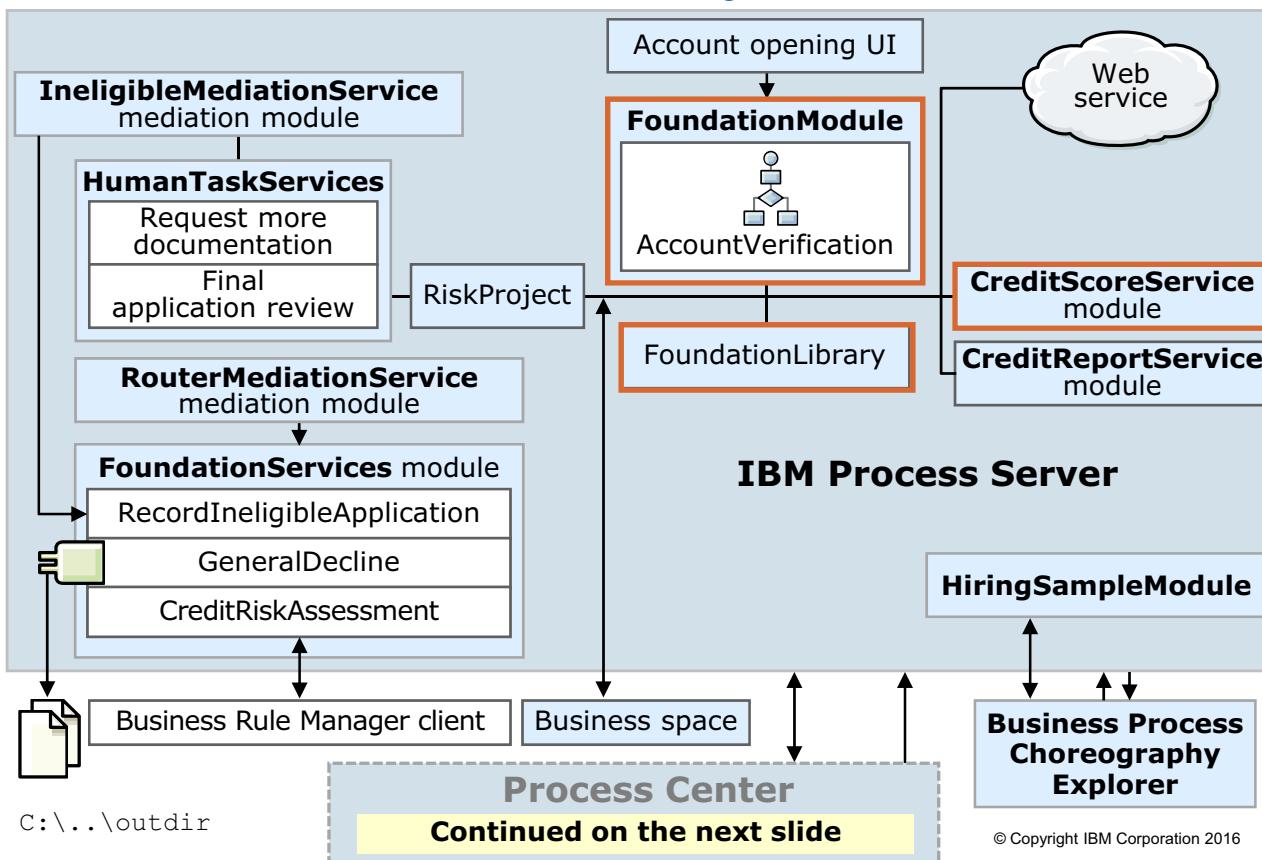


Figure 10-27. Course business scenario: What you work with in Exercise 8

WB8601.2

Notes:

Components that are required for Exercise 8

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **AccountVerification**
 - BPEL process that you built in Exercises 6 and 7

New components that you create in the lab:

1. **AccountVerification**
 - BPEL process that you completed in Exercise 7
 - You complete implementation of the process in this lab

Note:

The next slide lists all of the subcomponents of the **AccountVerification** process that are built in this lab.

© Copyright IBM Corporation 2016

Figure 10-28. Components that are required for Exercise 8

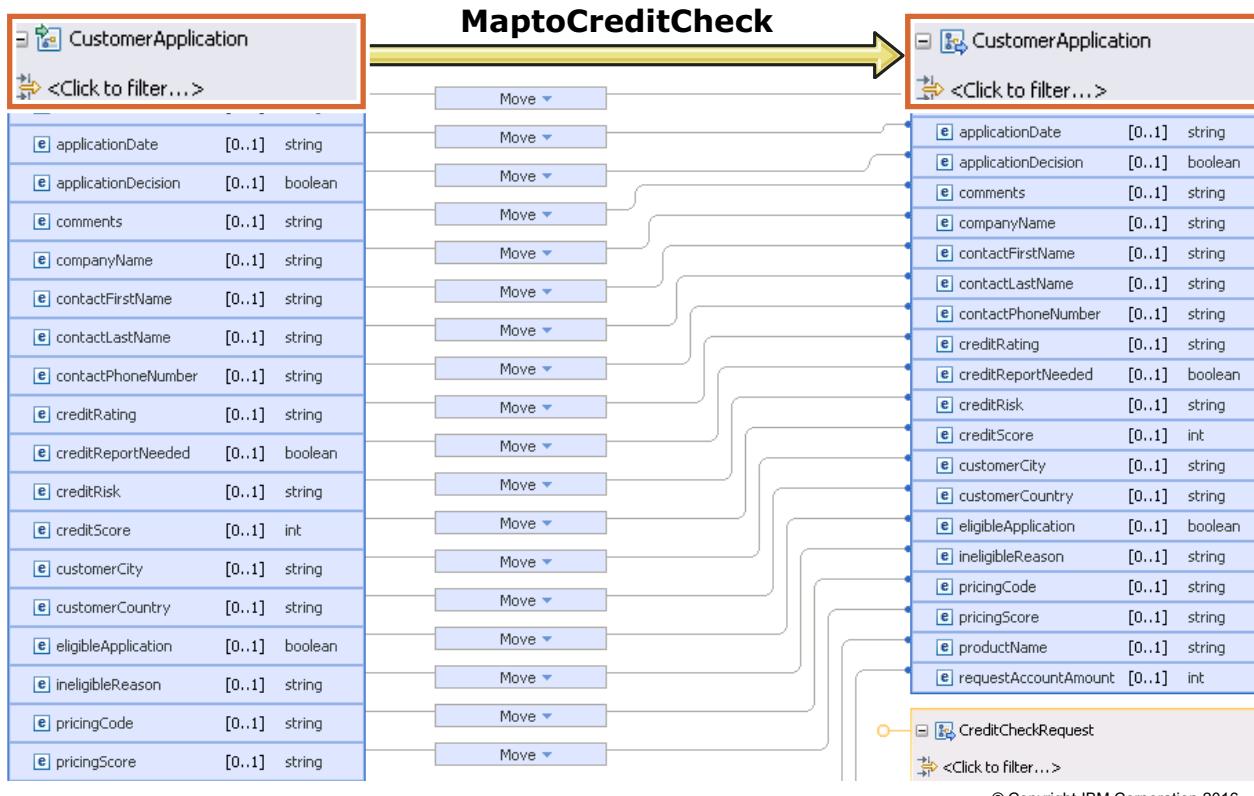
WB8601.2

Notes:

In this exercise, you create data maps to transform business objects between service requesters and providers. You also complete the core business logic for several of the services that the **AccountVerification** business process invokes. After you complete the service logic, you assemble the application that contains the **AccountVerification** process, and you test it.



Create data map in Exercise 8 (1 of 2)



© Copyright IBM Corporation 2016

Figure 10-29. Create data map in Exercise 8 (1 of 2)

WB8601.2

Notes:

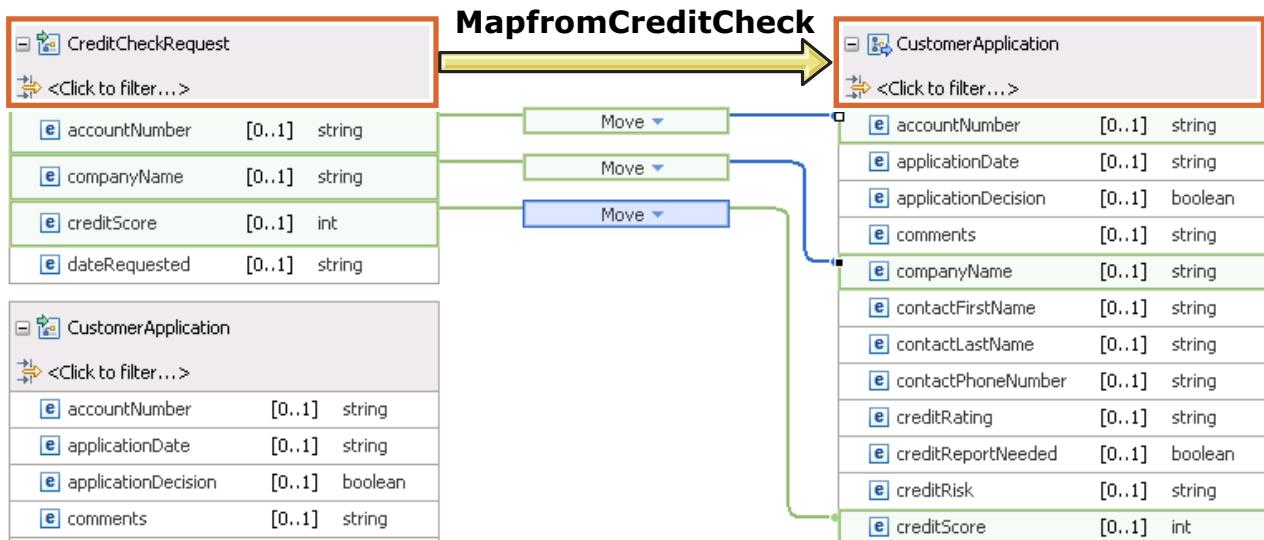
The AccountVerification business process that you are implementing receives and operates on a CustomerApplication business object input. During process execution, your business process calls an external web service, CreditScoreService, which provides the credit score that is needed to determine the customer's credit risk. Unlike the AccountVerification process (which uses a CustomerApplication business object input), the CreditScoreService interface uses a CreditCheckRequest business object for both the input and the output.

Because of this disparity, before you can invoke the CreditScoreService, you must transform the CustomerApplication business object into a CreditCheckRequest business object input. When the credit score is returned, you must transform the output from a CreditCheckRequest business object into a CustomerApplication business object. Transforming the data before and after service invocations in a business process can be done easily by using the data map activity in IBM Integration Designer.

The MapToCreditCheck data map moves data from the CustomerApplicationVariable input to both a CustomerApplicationVariable output and a CreditCheckVariable output.

You transform the data into both objects so that you can preserve the existing data in CustomerApplicationVariable and merge the response data back into it.

Create data map in Exercise 8 (2 of 2)



© Copyright IBM Corporation 2016

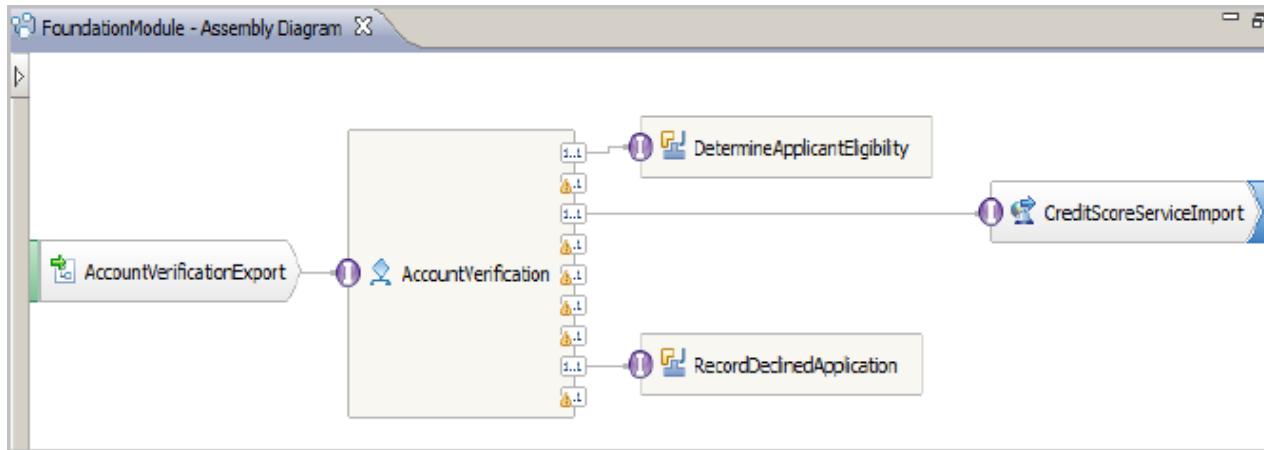
Figure 10-30. Create data map in Exercise 8 (2 of 2)

WB8601.2

Notes:

You create a **MapfromCreditCheck** data map that merges data from the **CreditCheckVariable** and **CustomerApplicationVariable** inputs into a **CustomerApplicationVariable** output.

Assemble the SCA application that contains the business process in Exercise 8



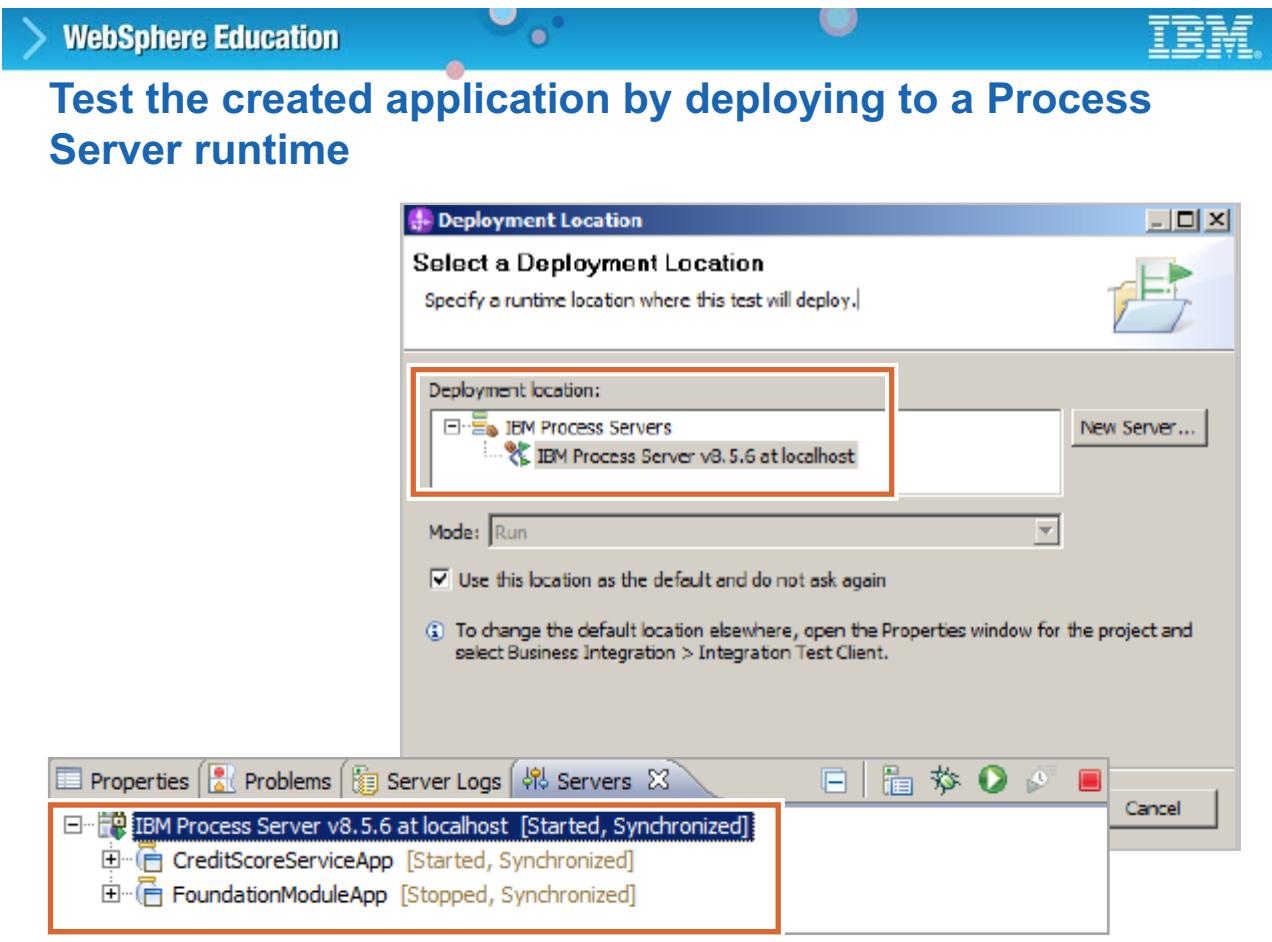
© Copyright IBM Corporation 2016

Figure 10-31. Assemble the SCA application that contains the business process in Exercise 8

WB8601.2

Notes:

In the assembly diagram, you wire the SCA components into an integrated application that is deployed to the runtime environment.



© Copyright IBM Corporation 2016

Figure 10-32. Test the created application by deploying to a Process Server runtime

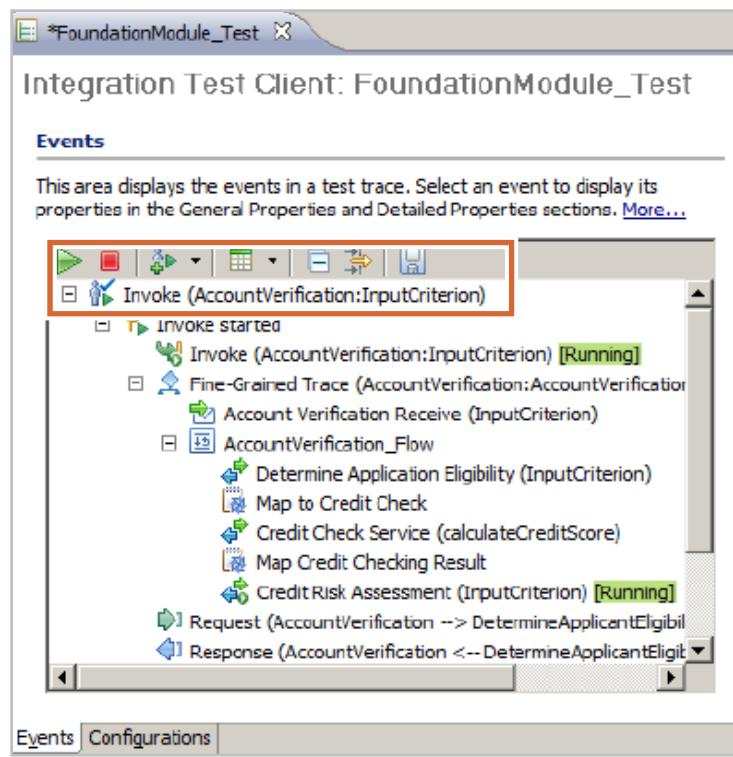
WB8601.2

Notes:



Use the test client in Exercise 8

- Use the Integration Test client to run and test the **AccountVerification** process



© Copyright IBM Corporation 2016

Figure 10-33. Use the test client in Exercise 8

WB8601.2

Notes:



References

- OASIS WS-BPEL Technical Committee
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- BPEL4WS and draft WS-BPEL specifications from IBM developerWorks
 - <http://bpel.xml.org/specifications>
- Business Process Choreographer samples
 - <http://publib.boulder.ibm.com/bpcsamp/v7r5/index.html>
- Versioning business processes and human tasks
 - http://www.ibm.com/developerworks/websphere/library/techarticles/0808_smolny/0808_smolny.html

© Copyright IBM Corporation 2016

Figure 10-34. References

WB8601.2

Notes:

Unit 11. Business rules

What this unit is about

This unit describes the purpose and function of business rules as an SCA component in IBM Process Server.

What you should be able to do

After completing this unit, you should be able to:

- Define the purpose and business value of using business rules
- Describe the function of a rule group and list the rule group components
- Define the concepts of rule sets and decision tables
- Describe the runtime behavior of a rule group component
- Identify the IBM Process Server administrative capabilities for importing, exporting, and auditing business rule changes in the runtime environment

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Define the purpose and business value of using business rules
- Describe the function of a rule group and list the rule group components
- Define the concepts of rule sets and decision tables
- Describe the runtime behavior of a rule group component
- Identify the IBM Process Server administrative capabilities for importing, exporting, and auditing business rule changes in the runtime environment

© Copyright IBM Corporation 2016

Figure 11-1. Unit objectives

WB8601.2

Notes:



Topics

- Overview of business rules
- Runtime aspects of rule groups

© Copyright IBM Corporation 2016

Figure 11-2. Topics

WB8601.2

Notes:

11.1. Overview of business rules

Overview of business rules



© Copyright IBM Corporation 2016

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

10.1

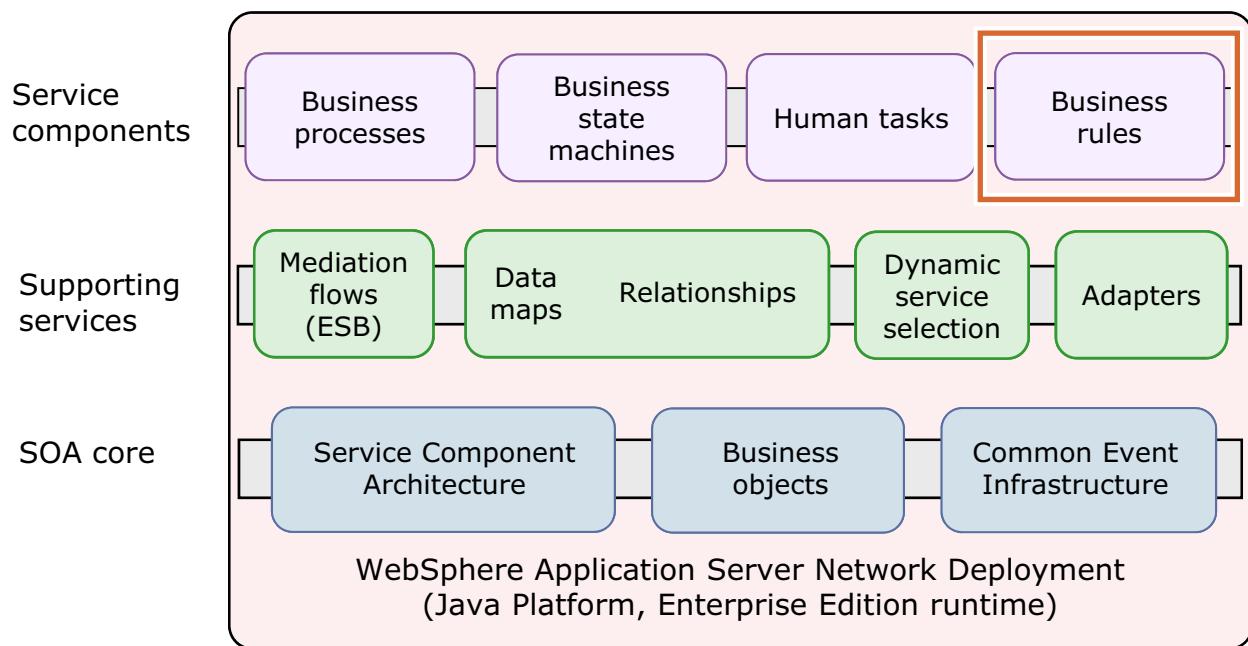
Figure 11-3. Overview of business rules

WB8601.2

Notes:

Business rules are a service component

- Business rules are part of the service components layer



© Copyright IBM Corporation 2016

Figure 11-4. Business rules are a service component

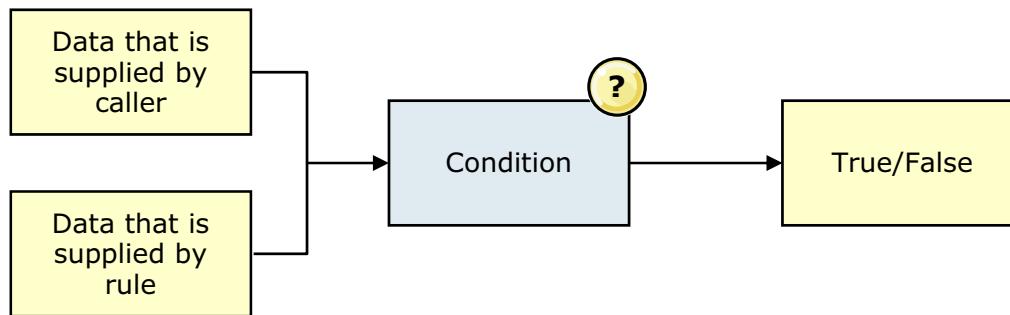
WB8601.2

Notes:

Business rules externalize frequently altered business policies. Business rules are an implementation type that is exposed as an SCA component. As an SCA component, business rules are exposed as services and can be called like any other SCA component.

What is a business rule?

- A business rule captures and implements business policies and practices by using one or more if-then statements
 - For example: If `orderTotal >=1000`, then the discount = 0.10
- A business rule consists of a condition (an expression that uses data that the caller and the rule supply) and one or more actions
 - The condition is the “if” portion of the statement
 - Evaluation of the condition is either true or false
 - The action is the “then” portion of the statement



© Copyright IBM Corporation 2016

Figure 11-5. What is a business rule?

WB8601.2

Notes:

When a business rule runs, a series of conditions are evaluated and a series of actions done. The outcome of the evaluation of the condition is a true or false value. An action is a simple piece of logic that can invoke a service, update a local variable, or as is most commonly the case, modify data that is returned to the caller. The action has access to both data supplied by the caller and data that is supplied by the rule.

A business rule is used to help abstract the client from implementation of the business logic. A calling client does not care how a business rule carries out its duties; it sends in the inputs and expects a result. Organization of rules is covered next, but to set the stage, business rules are assembled into business rule groups and can be implemented in one of two ways, rule sets or decision tables.

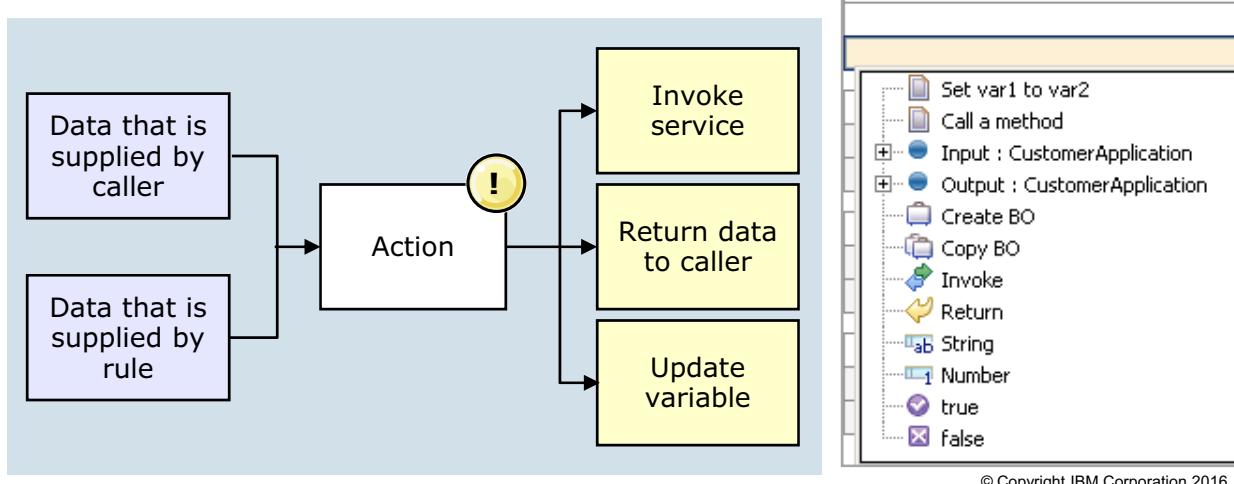
Examples of business rules include:

- A rental car company is able to change corporate discount amounts during peak dates and times: summer and holidays.
- An insurance agency can invoke a process for an insurance claim that happened in the past. It can happen because of a company merger or through a change in company policy.

- Shipping rates might change based on destination, package weight and size, or delivery priorities (overnight or next day).

Business rule actions

- Possible actions that result from the evaluation of the condition:
 - Invoking a service
 - Updating a local variable
 - Modifying data that is going to be returned to the caller
 - Creating or copying a business object
 - Returning (stopping execution of the rules early)



© Copyright IBM Corporation 2016

Figure 11-6. Business rule actions

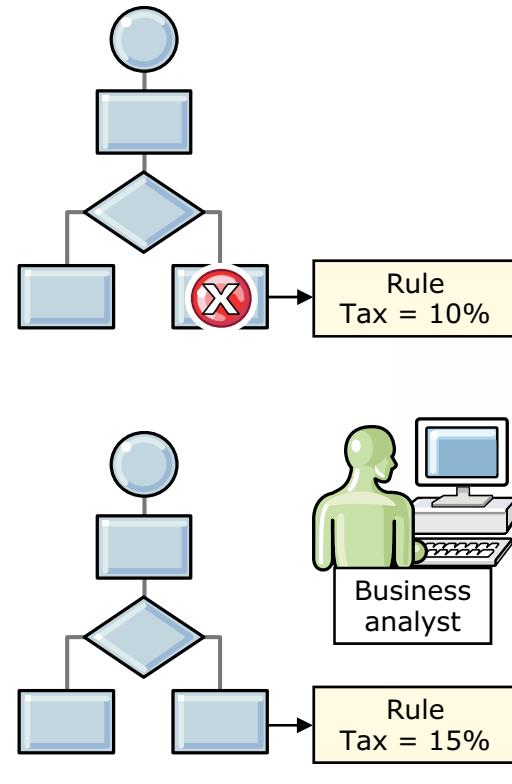
WB8601.2

Notes:

When the return statement runs, evaluation of rules stops. Output variable values at the return point become the output of the rules.

Business value of rules

- By using rule groups to expose rules as services, rules are separated from processes that use them
 - Multiple business processes can use the same sets of business rules
 - Rules are no longer in application code
- The business analyst can quickly change exposed rules at run time
 - If rules are not in application code, you are no longer bound to IT development cycles
 - The developer is needed only for more complex changes
- Rule groups are SCA components
 - As an SCA implementation type, it abstracts and decouples the rule implementation
- Two ways to implement business rules
 - Rule sets
 - Decision tables



© Copyright IBM Corporation 2016

Figure 11-7. Business value of rules

WB8601.2

Notes:

Hardcoding values and decision points into a business process is undesirable. Business values that are subject to change would result in manual recoding of the process and redeployment of the solution. These manual activities are error prone and take lot of time. IBM Process Server provides a much more elegant solution through business rules.

Decisions or values that affect the operation of a process can be externalized outside of the process. When these values are needed, the process asks for these values in a dynamic fashion by invoking the business rule service. IBM Process Server maintains the business rule values in a secure data store. They can be designed, configured, or modified at a high level through a specialized administrative web client. In principle, suitable staff members can modify these values without having to know anything about the underlying technical characteristics of the solution.

Why do you need a business rule service component when BPEL has a choice activity? In BPEL, you can use activities such as choice or receive choice; however, you are hardcoding the decision-making rules in the application. In a hardcoded logic, changes cannot be made easily or quickly.

By having a business rules component, the decision-making logic can be separated from the business process. Multiple services can invoke the rules, and business rules become reusable.

Changes in business rules (for example, the discount rate changes from 10% to 12%) do cause changes in business processes.

Business rules help abstract the implementation (the actual work done) from the client (the object that calls the work). Abstraction allows a company to change, update, and reuse code easily. This model of abstraction fits naturally within loosely coupled SOA processes. When they are decoupled, it is easy for other SCA services to access business rules. For example:

- An analyst can change loan approval criteria without redeploying the application.
- Flexibility allows for quick responses to changing business conditions and customer demand.
- Rules are reusable: multiple business processes can use the same set of business rules.

Rule sets defined

- Two types of rules can be used in a rule set: if-then and action
 - An if-then rule evaluates a condition, and then performs one or more actions
 - An action rule always performs actions, regardless of the input, without using a condition
- All rules in a rule set are evaluated sequentially, first to last
- Each **condition** that evaluates to true causes an **action** to be performed
 - More than one rule can be fired
- If-then and action rules can be exposed as template rules
 - Templates are exposed at run time in a natural language format
- The Business Rules Manager web client or Business Space client can be used to edit template rule parameters at run time
 - The developer can constrain parameter values

© Copyright IBM Corporation 2016

Figure 11-8. Rule sets defined

WB8601.2

Notes:

Business rules can be implemented in two ways: rule sets and decision tables. A rule set is a set of one or more if-then condition or action statements that are evaluated sequentially. Therefore, the first business rule that is listed is going to be the first one evaluated when called at run time. The second rule is evaluated next, continuing until the last one is evaluated. Rule sets hold any number of action or if-then conditions. A rule set can evaluate multiple conditions and can process multiple rules.



Rule sets in IBM Integration Designer (1 of 2)

- Rule sets in IBM Integration Designer contain:
 1. **Name (and Display Name)**
 2. **Interface**: Provides the operation, inputs, and outputs
 3. **Variables**: Rule actions can update local variables that you define

The screenshot shows the IBM Integration Designer interface for a rule set named "checkCustomerTypeRS".

- Rule Set:** Shows the name "checkCustomerTypeRS" and a "Display Name" field.
- Interface:** Shows an interface named "CheckCustomerType" with one operation "checkCustomerType". It also defines two variables: "inputCustomerType" (string type) and "outputValidType" (boolean type).
- Variables:** A table for defining local variables, currently empty.

(Rule set contents continue on the following page)

© Copyright IBM Corporation 2016

Figure 11-9. Rule sets in IBM Integration Designer (1 of 2)

WB8601.2

Notes:

The areas of the rule set editor in the figure are as follows:

1. The rule set properties: Use this icon to see detailed properties for your rule set.
2. The interface area: This area displays the interface that the rule set currently references.
3. The variables area: This area displays the variables that the rule set uses to store its data. To create a variable, click the icon; or to remove one, highlight it and click the icon.

Rule sets in IBM Integration Designer (2 of 2)

- Rule sets in IBM Integration Designer also contain:
 4. **Rules:** If-then rules and action rules
 5. **Templates:** Rules with editable parameters that are exposed in the runtime environment

The screenshot shows two panels of the IBM Integration Designer interface. The top panel is titled 'Rules' and contains a table with one row. The table has columns for Name (Rule1), Template (Template_Rule1), and Presentation (If inputCustomerType.equalsIgnoreCase(Competitor) == [False] then outputValidType = [true]). A toolbar with four icons is at the top of this panel. A yellow circle with the number 4 is next to the title. An arrow points from the right side of the toolbar to a callout box. The callout box is titled 'The Rules icon palette includes:' and lists: Add If-Then Rule, Add Action Rule, Add Template Rule, and Convert Rule to a Template.

The bottom panel is titled 'Templates' and contains a table with one row. The table has columns for Name (Template_Rule1), Presentation (If inputCustomerType.equalsIgnoreCase(stringParam0) == [booleanParam1] then outputValidType = [booleanParam2]), and Description. Below the table is a table for 'Parameters' with three rows: stringParam0 (string, None), booleanParam1 (boolean, None), and booleanParam2 (boolean, None). A toolbar with three icons is at the top of this panel. A yellow circle with the number 5 is next to the title. An arrow points from the right side of the toolbar to another callout box. This callout box is titled 'The Templates icon palette includes:' and lists: Add If-Then Template and Add Action Template. Another arrow points from the right side of the 'Parameters' table to a callout box. This callout box is titled 'Exposed parameters can be constrained to a predefined set of values' and has an arrow pointing to the 'Constraint' column in the 'Parameters' table.

© Copyright IBM Corporation 2016

Figure 11-10. Rule sets in IBM Integration Designer (2 of 2)

WB8601.2

Notes:

The remaining portions of the rule set editor are:

4. The icons in the rules area toolbar, which have the following functions:
 - Create an if-then rule
 - Create an action rule
 - Create a template rule
 - Create a template that is based on the current rule
5. The templates area: Use this area to graphically compose templates for a rule set. The icons in the templates area toolbar have the following functions:
 - Create a template for an if-then rule
 - Create a template for an action rule

A constraint is a restriction that is placed in a template by the person who creates it. This constraint limits how much a specified parameter can be modified. The two main types of constraints are:

- Range constraints: Range constraints apply to numeric types that are used within rules. For example, an authorized business user is allowed to adjust the discount on some merchandise, but these changes must be within 5% – 30%.
- Enumeration constraints: Enumeration constraints take the form of a list of either a numeric or a string type. The authorized business user must choose from one of the options available in the list, for example, upgrading a customer's credit rating from "silver" to "gold."

Decision table

- Decision tables represent multiple rule conditions in a table
 - Each axis of the table is an “if” condition
 - A single “then” action is defined at the intersection of the conditions
 - Only one rule is fired that meets all the conditions
 - A special “otherwise” condition can be used for the case when none of the conditions are met
- An example table is shown:
 - Axes represent package weight and volume conditions
 - The action that is taken is the intersection of the weight and volume criteria
- Decision table rules can also be exposed as templates

Conditions		Weight condition		
package.weight	< 1	> 1 and < 5	> 5	
package.volume	shippingAndHandlingCharge	shippingAndHandlingCharge	shippingAndHandlingCharge	
< 9	2	5	7	
> 9	7	7	7	
Actions				

© Copyright IBM Corporation 2016

Figure 11-11. Decision table

WB8601.2

Notes:

The nested tree structure of a decision table evaluates multiple rules efficiently. The order of condition evaluation is not specified. If more than one alternative is possible, the choice is non-deterministic.

A decision table represents a multi-dimensional nested if-then structure. It is a rule set that can handle more complex decisions than a simple if-then decision. The decision table is a set of “if” conditions with “then” actions that are defined at the intersection points of the table. Conditions are evaluated in a nested order (navigation tree). A decision table evaluates one or more conditions, but processes only one rule.



Decision tables in IBM Integration Designer (1 of 2)

- Decision tables in IBM Integration Designer contain:
 - Name (and Display Name)**
 - Interface:** Provides operations, inputs, and outputs for the decision table

The screenshot shows the IBM Integration Designer interface. At the top, there's a toolbar with various icons. Below the toolbar, the main area is divided into two sections: 'Decision Table' and 'Interface'.

Decision Table:

Name	calcShipping	Display Name	calcShipping
------	--------------	--------------	--------------

Interface:

(I) Interface	CalculateShipping	
Operation	calcShipping	
Input	packageType	Package
Output	shippingCost	int

Two yellow circles with numbers 1 and 2 are overlaid on the screenshot, pointing to the 'Decision Table' section and the 'Interface' section respectively.

(Decision table contents continue on the following page)

© Copyright IBM Corporation 2016

Figure 11-12. Decision tables in IBM Integration Designer (1 of 2)

WB8601.2

Notes:

The areas of the decision table editor are as follows:

- The decision table properties: Use this icon to see detailed properties for your decision table.
- The interface area: This area displays the interface that is being referenced, and the inputs and outputs that you can use in the decision table.

Decision tables in IBM Integration Designer (2 of 2)

- Decision tables also contain:
 - Initialize:** An action rule that automatically works on an operation when data is first passed to the decision table, such as copying a business object or initializing a variable; only one initialize rule is allowed
 - Table:** Contains the multi-dimensional if-then rules

The screenshot shows the IBM Integration Designer interface. At the top, there are two toolbars: 'Initialize' (numbered 3) and 'Table' (numbered 4). The 'Initialize' toolbar contains icons for creating initialization rules and converting rules to templates. The 'Table' toolbar contains icons for adding conditions, values, actions, changing orientation, and converting rules or cells to templates. Below these toolbars is a decision table editor. The table has four columns under the 'Conditions' header: 'package.weight' with values '< 1', '> 1 and < 5', and '> 5'; 'package.volume' with values 'shippingAndHandlingCharge' and '2'; and '< 9' and '> 9'. The last column is labeled 'Actions'. A callout box for the 'Initialize' toolbar lists: 'Create initialization rule' and 'Convert rule to template'. Another callout box for the 'Table' toolbar lists: 'Add a condition', 'Add a condition value', 'Add an action', 'Change orientation', and 'Convert rule or cell to template'.

Conditions	< 1	> 1 and < 5	> 5
package.weight			
package.volume	shippingAndHandlingCharge	shippingAndHandlingCharge	shippingAndHandlingCharge
< 9	2	5	7
> 9	7	7	7

© Copyright IBM Corporation 2016

Figure 11-13. Decision tables in IBM Integration Designer (2 of 2)

WB8601.2

Notes:

The areas of the decision table editor are continued:

- The initialize area: Use the icons in the initialize toolbar to do the following functions:

- Add an initialization action rule (an operation that takes place when data first enters a decision table)
- Create a template from this action rule

An initialization action rule automatically does an operation of some kind when data is first being passed to a decision table. Typically, an initialization action rule is used for the following reasons:

- To set an initial value, for example, to copy input business objects to output business objects.
- To initialize an output business object, for instance, to set the initial value of the business object attributes. More importantly, if the business object has attributes that are also business objects, then you can create those business objects (and this process can be recursive if they in turn contain even more business objects).

Initialization action rules can also be made into templates so that they can be modified in the runtime environment.

4. The decision table area: Use the icons in the decision table toolbar to have the following functions:

- Add a condition
- Add a condition value
- Add an action
- Change the orientation of the condition
- Create a template for one of the expressions in this table

The **conditions area** is displayed in light blue. Use this area of the decision table to define the multiple conditions that evaluate the incoming inputs to fire a corresponding action.

The **actions area** is displayed in light gray. Use this area of the decision table to define the actions that fire when the conditions intersect.

Rule sets versus decision tables

- Use rule sets when:
 - You want to fire multiple rules
 - You have several simple rules, each with one condition
- Use decision tables when:
 - You want to fire only one rule
 - You have complex rules with multiple conditions
- Rule sets and decision tables can be used together
 - You can have several rule sets
 - You can have several decision tables
 - You can have combinations of rule sets and decision tables

© Copyright IBM Corporation 2016

Figure 11-14. Rule sets versus decision tables

WB8601.2

Notes:

Use decision tables when you have rules with multiple clauses or variables in the condition statements. More importantly, use a decision table when you want to process only one rule.

Use rule sets when you have rules with a few clauses or variables in the condition statements, or you must process multiple rules. (Evaluating sequentially can be inefficient when many rules exist.)

A developer typically does not have the authority to make business decisions that involve rules (such as how much of a discount to give to certain types of customers). When you develop business rules, you must typically involve a business analyst.

Rule groups

- A rule group is an SCA component that is used to dynamically invoke rule sets and decision tables, which are based on set criteria
 - The criterion is a date and time range
 - Only one target runs based on a date selection criterion
 - Start and end date/time criteria, and target rule sets or decision tables, can be modified at run time
 - Criteria can also be based on the content of the business object input, but this type of choice must be created programmatically
 - New rule sets and decision tables cannot be created at run time, but new rules can be created from exposed templates
- Groups organize rules that share a common business purpose
 - Groups are searchable at run time
 - A rule group might contain any number of rule sets or decision tables
- Rule groups are presented like any other SCA component

© Copyright IBM Corporation 2016

Figure 11-15. Rule groups

WB8601.2

Notes:

A business rule group is the highest level implementation component for a business rule. The main idea behind a rule group is to gather rule sets or decision tables that share a common business focus. It also serves as the gateway to the business rules because it is exposed in the runtime environment as an SCA service. All other services invoke a business rule on a request that a client (BPEL, selector) sends through the rule group. Therefore, a rule group must be established before a rule set or decision table can run.

One of the most important functions of the rule group is to define a date and time range for business rule execution. For example, during the time between 1/1/05 12:00 AM and 1/1/06 12:00 AM, use decision table one. The date and time that are provided during invocation determine which business rule to use.

Rule implementations can be “scheduled” for a point in the past, present, or future. This scheduling gives a company the ability to invoke a process as it was in the past, or to switch over to the next version of a business rule on a future date.

Business rule groups can be searched. In each search data field, you can select one of four query operators: **is equal to**, **is like**, **is not equal to**, **is not like**. The **is like** and **is not like** operators can contain wildcard characters, such as the percent sign, to specify a wildcard for any number of

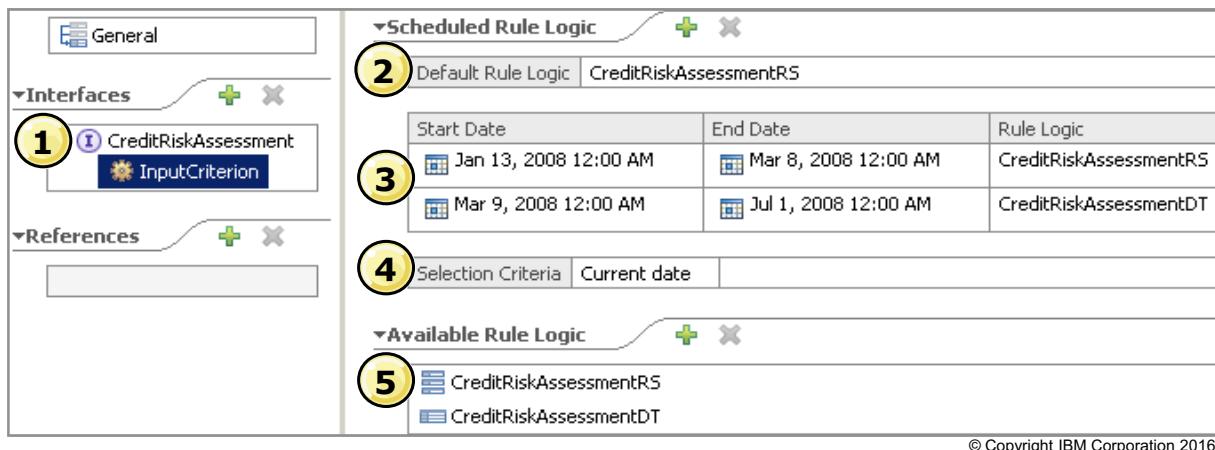
characters and an underscore to specify a single wildcard character. The wildcards must follow SQL syntax.

You can also select from the user-defined properties to add to the search context, and you can combine these properties by logical operators “and,” “or” and “not.” When you add, delete or modify properties on the search page, it applies only to the search context. It does not affect the properties of any rule object inside the business rules manager.



Rule group components

- Rule groups are composed of:
 1. An interface
 2. A default destination (ensures a target)
 3. One or more target destinations corresponding to the start and end date/time ranges (dates cannot overlap)
 4. A date selection criterion (the value is compared to the target start-end dates)
 5. A set of available destinations (rule sets and decision tables)



© Copyright IBM Corporation 2016

Figure 11-16. Rule group components

WB8601.2

Notes:

A rule group has five basic components:

1. An interface with an operation and operation parameters (for example, input1 and output1).
2. A default destination (a rule set or decision table to use when none of the active destinations apply).
3. A set of active destinations that are composed of rule sets and decision tables with a set of start and end date ranges for each of the active destinations.
4. A selection criterion that is used to evaluate which destination is used. The three criteria options are:
 - To use the current date and time (that is, the current date and time on the server that runs the application)
 - To “mine” for a date and time by using XPath or a Java snippet
 - To use custom Java code to return a date and time
5. A set of available destinations that can be added to the list of active destinations.

The example shows only one rule set and one decision table, but you can have multiples of only rule sets, or decision tables, or mixtures of both.

Business rule group properties

- Two types of rule group properties: system and user-defined
- System properties are read-only, and IBM Integration Designer and IBM Process Server use them internally
- User-defined (custom) properties are read and write
- Custom properties are used for the management of rule groups
 - Number of user properties that you can define is unlimited
 - Add and delete in IBM Integration Designer, and modify in Business Rules Manager
 - Can be queried through Business Rules Manager and the public business rules management API
 - Can hold customer-specific information

Name	Value
Contact	john.doe@us.ibm.com

© Copyright IBM Corporation 2016

Figure 11-17. Business rule group properties

WB8601.2

Notes:

Two types of business rule group properties exist: system properties, which are IBM-use only and read-only; and user-defined or “custom” properties, which are read/write. Custom property names and values are case-sensitive.

Business rule group properties are defined in IBM Integration Designer but are modifiable in the business rules manager web client.

You can use custom properties on business rule groups for searches to retrieve subsets of business rule groups that you want to view and modify. You add new custom properties, delete existing properties, or modify existing properties through the editing pages of business rule groups. The number of custom properties on a business rule group is unlimited.

Display names and description fields for rule groups

- The **Description** field is for rules, rule sets, decision tables, and rule groups
- The **Display Name** field is for rule sets, decision tables, and rules that are used to support documentation
- Display names:
 - Can be any string value with special characters
 - Do not have to be unique

General Information

Display Name	CreditRiskAssessmentRS	<input checked="" type="checkbox"/> Synchronize with the name										
Last Published	Jan 24, 2008 13:00 (Local Time)	Status										
Description	<input type="text"/> <div style="border: 1px solid black; padding: 5px; width: 100%;"> You can add a description for the rule set and for individual rules </div>											
Rules <input type="button" value="New Rule from Template"/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Display Name</th> <th>Rule</th> <th>Description</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>RiskHIGH</td> <td>RiskHIGH</td> <td>If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/></td> <td><input type="button"/></td> <td><input type="button"/> <input type="button"/> Delete <input checked="" type="checkbox"/> Synchronize Name</td> </tr> </tbody> </table>			Name	Display Name	Rule	Description	Action	RiskHIGH	RiskHIGH	If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/>	<input type="button"/>	<input type="button"/> <input type="button"/> Delete <input checked="" type="checkbox"/> Synchronize Name
Name	Display Name	Rule	Description	Action								
RiskHIGH	RiskHIGH	If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/>	<input type="button"/>	<input type="button"/> <input type="button"/> Delete <input checked="" type="checkbox"/> Synchronize Name								

© Copyright IBM Corporation 2016

Figure 11-18. Display names and description fields for rule groups

WB8601.2

Notes:

The display names replace the current name values of rule groups, and they are used everywhere that the name values are currently shown, including the left navigator and when the artifacts are displayed in detail. The display names for rule groups are not required to be unique inside the business rule artifacts of the same type (business rule groups). Names of the business rule artifacts are still required to be unique in any use cases.

WebSphere Education

Business rules: Tools (1 of 2)

- IBM Integration Designer tools:
 - Easy to use tools for defining, executing, and managing business rules
 - Eclipse based tools for Business Rule development
 - Focused on the more technical developer role
 - Business Integration view is used mostly for Business Rule interaction

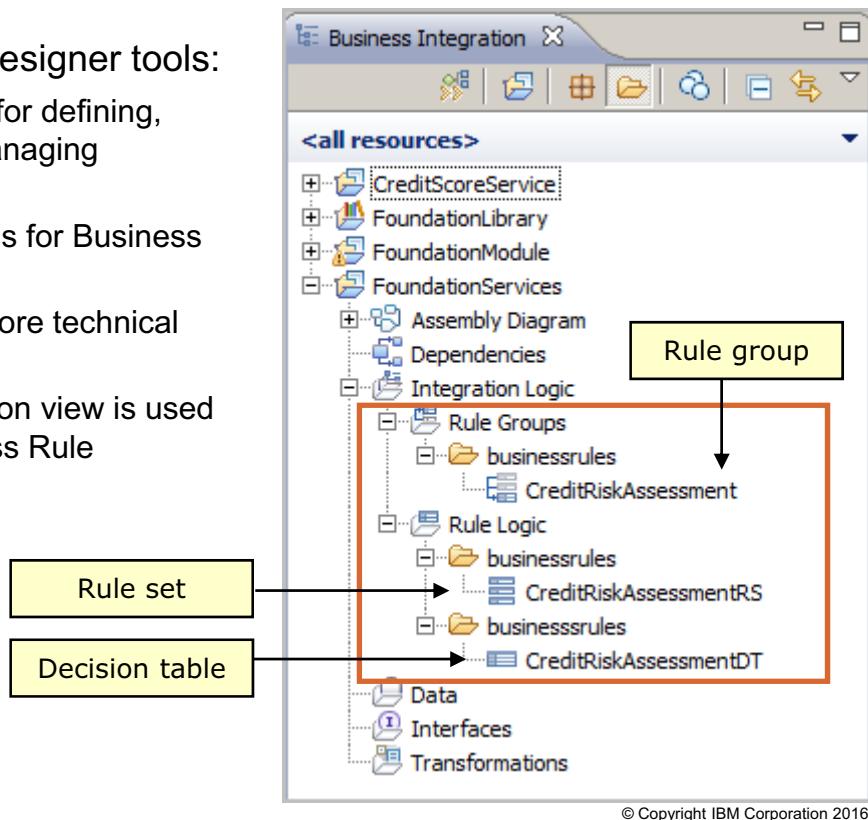


Figure 11-19. Business rules: Tools (1 of 2)

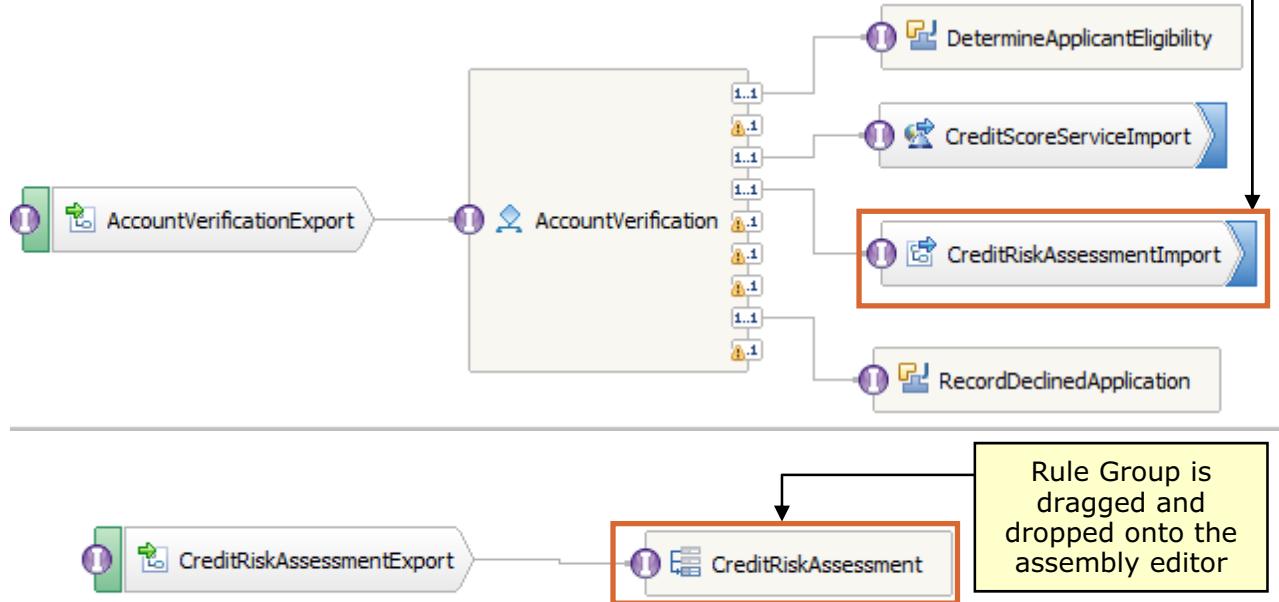
WB8601.2

Notes:

Business rules: Tools (2 of 2)

- Assembly editor is used to connect Business rules to calling SCA Components

Import used to connect Business rule to another module



© Copyright IBM Corporation 2016

Figure 11-20. Business rules: Tools (2 of 2)

WB8601.2

Notes:



Business rules in IBM Process Designer

- An activity implementation can contain business rules in IBM Process Designer
 - Called a “decision service”
 - Written in Business Action Language (BAL)
- The following lists differences between rules in IBM Process Designer and IBM Integration Designer:
 - Decision services implement activities or actions only; they do not assign values or run self-contained code
 - BAL is a declarative language, and business rules in IBM Integration Designer support declarative presentations and programmatic rules
 - Rule sets and decision tables are not programmatically selected in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 11-21. Business rules in IBM Process Designer

WB8601.2

Notes:

11.2. Runtime aspects of rule groups

Runtime aspects of rule groups



© Copyright IBM Corporation 2016

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

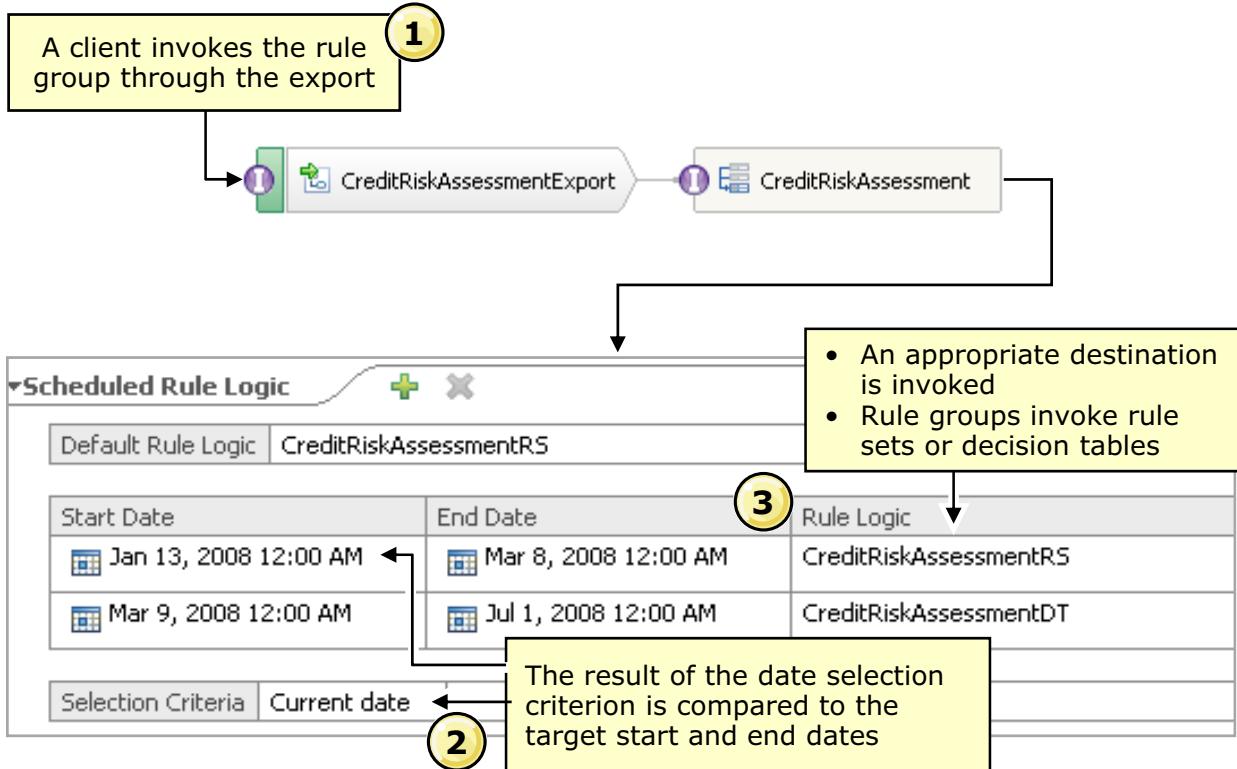
10.1

Figure 11-22. Runtime aspects of rule groups

WB8601.2

Notes:

Business rules at run time



© Copyright IBM Corporation 2016

Figure 11-23. Business rules at run time

WB8601.2

Notes:

Rule groups function in the following way at run time:

1. A client (JSP, POJO, SCA component) calls the rule group component interface through the export component.
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 invoked.
 - Rule group targets are rule sets and decision tables.
 - If the date criterion falls outside the start and end dates for the rule sets and decision tables, the default destination is used.

More runtime aspects of business rules

- The IBM Process Server runtime handles many business rules
 - However, advanced rule engines like Operational Decision Manager can handle complex rule inference and visualization
- The runtime uses service calls and SCA to support integration with other rule engines
- Business Rules Manager web client and Business Space are used to interact with rules at run time
 - Business Rules Manager is installed with IBM Business Process Manager Advanced
- Business rule information is stored in the IBM Process Server common database (CMNDB) for all supported database types

© Copyright IBM Corporation 2016

Figure 11-24. More runtime aspects of business rules

WB8601.2

Notes:

IBM Process Server handles typical integration rules, but does not handle the other complex rules like inferencing. Complex rules are left to rule engines such as Operational Decision Manager.

The IBM Process Server runtime environment can use the business rules manager web client to edit and update business rules. In addition, Business Space contains a Real-Time Business Configuration widget that can be used to interact with business rules.

Designer and developer roles in business rules

- Exposing rules as templates at run time is based on the separation of responsibilities

Role	Tools used	Tasks
Business analyst (business skills)	Business Rules Manager and Business Space	<ul style="list-style-type: none"> • The business analyst (or architect) designs rules • Analysts adjust rule parameters at run time with business and customer needs in mind
Integration developer (technical skills)	IBM Integration Designer	<ul style="list-style-type: none"> • The developer codes the rules and the rule templates in IBM Integration Designer • SCA components and assembled applications are created by using SOA principles for flexibility and reuse

© Copyright IBM Corporation 2016

Figure 11-25. Designer and developer roles in business rules

WB8601.2

Notes:

The integration developer uses IBM Integration Designer to create or manage the technical details of the business rules. The developer works with the IT architect to implement the business process plan. The business analyst role is more in tune with the changing needs of the business, watching the market, and deciding upon how to provide services (promotional discounts, policy changes, services for a certain age group). Analysts use the business rules manager web client and the Business Space Real-Time Business Configuration widget to change the business rules whenever necessary. The business analyst also works with the developer to update and create business rules for the company.

The Business Rule Manager web tool and the Business Space Real-Time Business Configuration widget are built specifically for the business analyst. They have a manageable subset of full authoring functions that are found in IBM Integration Designer. The only way a business analyst is able to see any rules in the web tools is when the developers build the templates. These templates give the business analyst an easy way to edit business rules at run time by using only a browser.



Business rules manager web client

- The Business Rules Manager client is a browser-based application that can publish changes to business rules or revert to previous rules at run time
- WebSphere Application Server security authenticates the users
- You can change rule group targets, start and end dates, the default rule group, or the decision table

The screenshot shows the Business Rules Manager web client interface. On the left, there's a sidebar with navigation links: 'Publish and Revert' (disabled), 'Business Rule Groups' (selected), 'CreditRiskAssessment' (under Business Rule Groups), 'InputCriterion' (under CreditRiskAssessment), 'CreditRiskAssessmentD' (under InputCriterion), and 'CreditRiskAssessmentR' (under InputCriterion). The main panel is titled 'Business Rule Groups' and lists 'Business Rules Resources' with items 'CreditRiskAssessment', 'InputCriterion', and 'CreditRiskAssessmentDT'. Below this is a section titled 'Scheduled Rule Logic' with instructions: 'Click button to choose from specifying date, no start/end date, and continuous for automatic end date calculation.' It includes 'Add Selection Record' and 'Sort' buttons, and two sets of date/time input fields for 'Start Date/Time' and 'End Date/Time'. The 'Start Date/Time' fields show Jan 1, 2008, 00:00 and Mar 2, 2008, 00:01. The 'End Date/Time' fields show Feb 29, 2008, 24:00 and Mar 13, 2009, 24:00. A copyright notice at the bottom right reads '© Copyright IBM Corporation 2016'.

Figure 11-26. Business rules manager web client

WB8601.2

Notes:

The business rules manager web client is started through a browser, not the administration console. The business rules manager (BRM) is an enterprise application (a set of JSP pages and servlets), which manages the business rules that are running on the server. If you use the typical installation option to install IBM Business Process Manager Advanced, it is installed by default.

The Business Rule Manager web tool can be accessed by opening a browser and entering:

`http://localhost:9080/br/webclient/pages/index.jsp`

You can view all business rules that are exposed through the rule groups that run on the server.

Templates in the business rules manager

- Using the Business Rules Manager web client, you can:
 - Change the rule names and the exposed template parameters
 - Create rules from templates
 - Delete existing rules
 - Change the order of rule execution

Edit Mode: CreditRiskAssessmentRS - Rule Set

Save Cancel Messages:

General Information

Display Name	CreditRiskAssessmentRS	<input checked="" type="checkbox"/> Synchronize with the name
Last Published	Jan 24, 2008 13:00 (Local Time)	Status Original
Description		

Rules

You can create a rule from a template, change the execution order, or delete a rule

Name	Display Name	Rule	Description	Action
RiskHIGH	RiskHIGH	If the customer credit score is greater than <input type="text" value="0"/> and less than <input type="text" value="4"/> then the credit risk is <input type="text" value="HIGH"/>	<input type="checkbox"/>	<input type="button" value="Delete"/> <input checked="" type="checkbox"/> Synchronize Name
RiskMED	RiskMED	If the customer credit score is greater than <input type="text" value="3"/> and less than <input type="text" value="8"/> then the credit risk is <input type="text" value="MED"/>	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Up"/> <input checked="" type="checkbox"/> Synchronize Name

© Copyright IBM Corporation 2016

Figure 11-27. Templates in the business rules manager

WB8601.2

Notes:

The market changes quickly. If web applications do not respond to such changes in a timely manner, business can be lost to competitors. Any minor changes to code would require regression testing, uninstalling the old version of the application, and installing the new application. The business rules manager allows immediate responsiveness and flexibility by allowing a business analyst to change rule parameters at run time.

Exporting and importing rule changes

- Changes that are made to the runtime configuration of rules can be exported, and then imported into IBM Integration Designer
 - Use the command line to export the rules to a compressed file
 - Exported rules are imported into IBM Integration Designer to synchronize runtime and development environments
- Changes in IBM Integration Designer can be imported into the runtime without redeployment

© Copyright IBM Corporation 2016

Figure 11-28. Exporting and importing rule changes

WB8601.2

Notes:

You import rule groups after changing business rules in use by installed applications, and you are ready to bring those changes into another server. You can also use this facility to synchronize your development environment with changes in the production environment. You export rule group components after changing business rules, and you must synchronize your production environment with your development environment. You can use the command line to export and import rule group components. The import function is intended to allow business rule changes to be tested on a test server and then moved over to a production server after testing is complete.

Business rule auditing

- Changes to business rules at run time can be audited
- Audit logging for rules can be configured by using the server command line interface
- You can write changes to `SystemOut.log` or a custom audit log
- Auditing records:
 - Name of person who changes the rule
 - Location where the change request originated
 - Old business rule object
 - New business rule object

© Copyright IBM Corporation 2016

Figure 11-29. Business rule auditing

WB8601.2

Notes:

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 must examine 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 that are made to business rules. You can configure your server to automatically detect when changes are made to business rules and create an entry in a log file that details the changes. You can choose to have the log entries that are written either to 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 change is made logs the following information:

- The name of the person who incorporates the change
- The location from where the change request originated
- The old business rule object
- The new business rule object



Public business rules API

- The public business rules API can be used to create custom rule clients similar to the Business Rules Manager
- APIs can be used to:
 - Query business rule groups by name, namespace, or custom properties
 - Modify the business rule group selection table
 - Add and delete rules inside a rule set
 - Add or delete conditions and actions in decision tables
- Use with other APIs to build complete management clients
 - Manage business processes, human tasks, and business rules through a single application

© Copyright IBM Corporation 2016

Figure 11-30. Public business rules API

WB8601.2

Notes:

Unit summary

- Define the purpose and business value of using business rules
- Describe the function of a rule group and list the rule group components
- Define the concepts of rule sets and decision tables
- Describe the runtime behavior of a rule group component
- Identify the IBM Process Server administrative capabilities for importing, exporting, and auditing business rule changes in the runtime environment

© Copyright IBM Corporation 2016

Figure 11-31. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. True or False: All rules in a rule set are evaluated in the order that they are written.
2. True or False: You can change the parameters of a business rule at run time without redeploying the application.
3. True or False: A decision table fires only one rule; a rule set might fire several.
4. True or False: Runtime changes to business rules can be exported from IBM Process Server and imported into IBM Integration Designer.

© Copyright IBM Corporation 2016

Figure 11-32. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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

Checkpoint answers

1. True.
2. True: If the developer exposes the rule parameters by using a template.
3. True.
4. True

© Copyright IBM Corporation 2016

Figure 11-33. Checkpoint answers

WB8601.2

Notes:

Exercise 9



Creating business rules

© Copyright IBM Corporation 2016

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

10.1

Figure 11-34. Exercise 9

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create rule sets and decision tables that contain business rules
- Create a rule group component
- Incorporate a rule group component in an assembly diagram
- Test a business rule group in the integration test client
- Use the Business Rule Manager web client to interact with business rules at run time

© Copyright IBM Corporation 2016

Figure 11-35. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 9

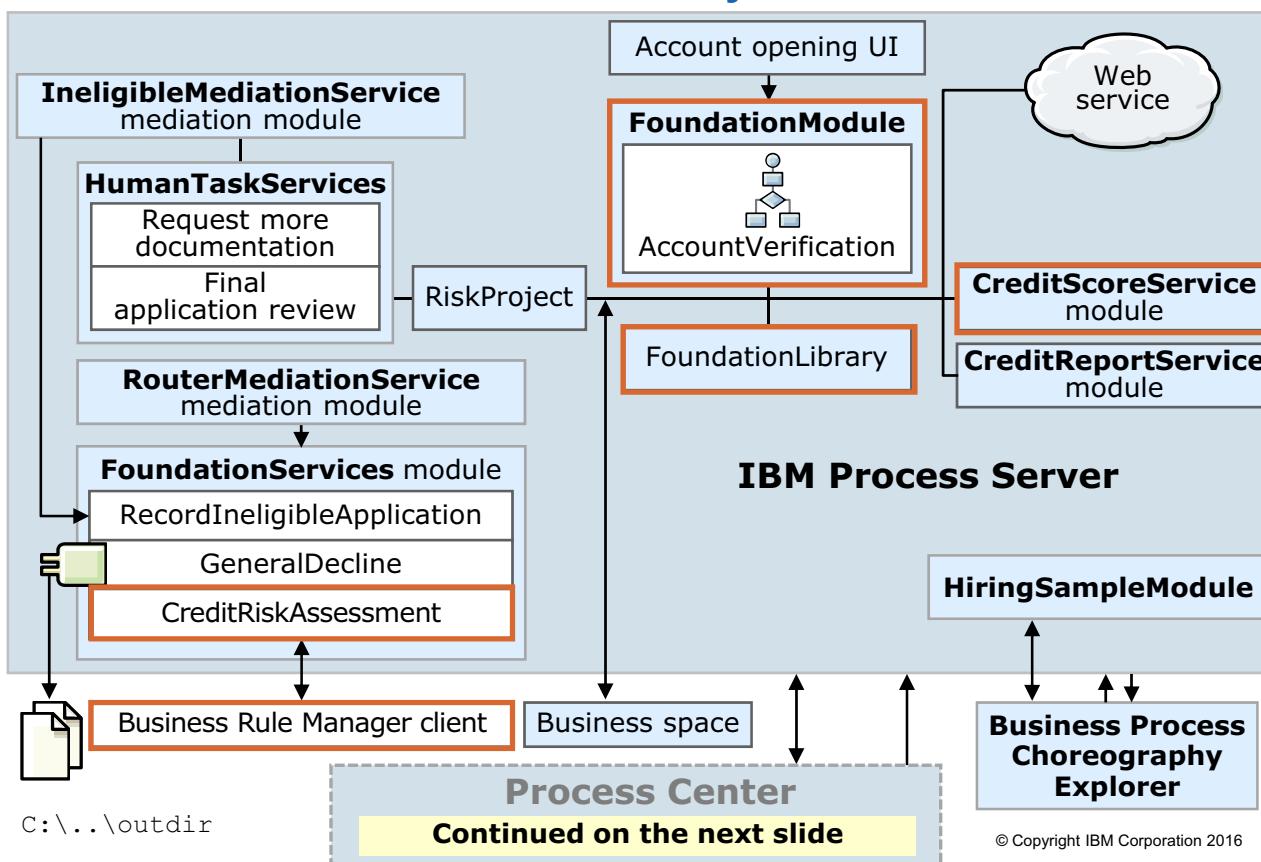


Figure 11-36. Course business scenario: What you work with in Exercise 9

WB8601.2

Notes:

Components that are required for Exercise 9

Prebuilt components that are imported in the lab:

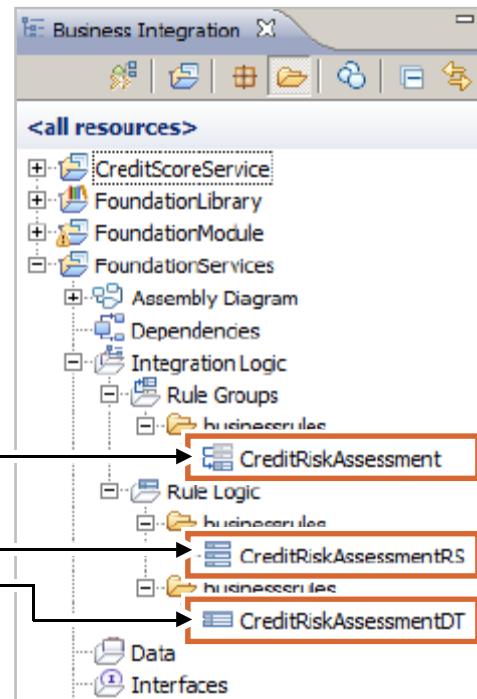
- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. AccountVerification**

- BPEL process that you completed in Exercise 8

New components that you create in the lab:

- 1. FoundationServices** module that contains:

- **CreditRiskAssessment** rule group
- **MapInputToOutput** rule
- **CreditRiskAssessmentRT** rule set
- **CreditRiskAssessmentDT** decision table



© Copyright IBM Corporation 2016

Figure 11-37. Components that are required for Exercise 9

WB8601.2

Notes:

In this exercise, you create rule sets and decision tables that contain business rules. You create a rule group component, incorporate the rule group component in an assembly diagram, and test it. Finally, you use the Business Rules Manager web client to interact with business rules at run time.

To determine the credit risk, a series of business rules are used to evaluate the credit score that the credit score service returns.

In this exercise, you create the business rules that the credit risk assessment service uses. If the creditScore value is less than 4, then the creditRisk is HIGH. If the creditScore value is in the range of 4 – 7, then the creditRisk is MED (short for medium). If the creditScore value is in the range of 8 – 11, then the creditRisk is LOW. Customer applications that are HIGH risk require more documentation and are subject to more review. Customer applications that are MED risk require more review but not more documentation. Customer applications that are LOW risk are automatically approved.

You create a **CreditRiskAssessment** business rule group in the **FoundationServices** module that uses the **CreditRiskAssessment** interface. You also create a **CreditRiskAssessmentDT** decision table in the rule group.

You then create an action rule that is named **MapInputToOutput** that copies the data from the input business object and assigns it to the output business object.

Create rule group in Exercise 9

Scheduled Rule Logic

Default Rule Logic	CreditRiskAssessmentRS
Start Date	Nov 3, 2015 12:00 AM
End Date	Nov 3, 2016 12:00 AM
Rule Logic	CreditRiskAssessmentRS

Rules

Name	RiskHIGH
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than 0 and less than 4 then the credit risk is HIGH
Name	RiskMED
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than 3 and less than 8 then the credit risk is MED
Name	RiskLOW
Template	CreditRiskTemplate
Presentation	If the customer credit score is greater than 7 and less than 12 then the credit risk is LOW

- Test the created application by deploying to a Process Server runtime

© Copyright IBM Corporation 2016

Figure 11-38. Create rule group in Exercise 9

WB8601.2

Notes:



References

- Business Rules Community
 - <http://www.brcommunity.com>
- Business Rules Group
 - <http://www.businessrulesgroup.org>
- SOA programming model for implementing web services, part 9: Integrating rules with SOA
 - <http://www.ibm.com/developerworks/webservices/library/ws-soa-progmodel9/index.html>
- Creating and deploying business rules
 - http://www.ibm.com/developerworks/websphere/library/tutorials/0610_kolban/0610_kolban.html

© Copyright IBM Corporation 2016

Figure 11-39. References

WB8601.2

Notes:

Unit 12. Adapters

What this unit is about

In this unit, you learn about WebSphere (JCA) adapters and how they are used to integrate solutions with other applications.

What you should be able to do

After completing this unit, you should be able to:

- Describe the purpose and business value of using adapters in applications
- Describe the capabilities of WebSphere (JCA) adapters
- List the advantages of using the JCA architecture for WebSphere Adapters
- Describe how to use the External Service wizard for WebSphere Adapters

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the purpose and business value of using adapters in applications
- Describe the capabilities of WebSphere (JCA) adapters
- List the advantages of using the JCA architecture for WebSphere Adapters
- Describe how to use the External Service wizard for WebSphere Adapters

© Copyright IBM Corporation 2016

Figure 12-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to adapters
- Using WebSphere (JCA) Adapters

© Copyright IBM Corporation 2016

Figure 12-2. Topics

WB8601.2

Notes:

12.1. Introduction to adapters

Introduction to adapters



© Copyright IBM Corporation 2016

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

10.1

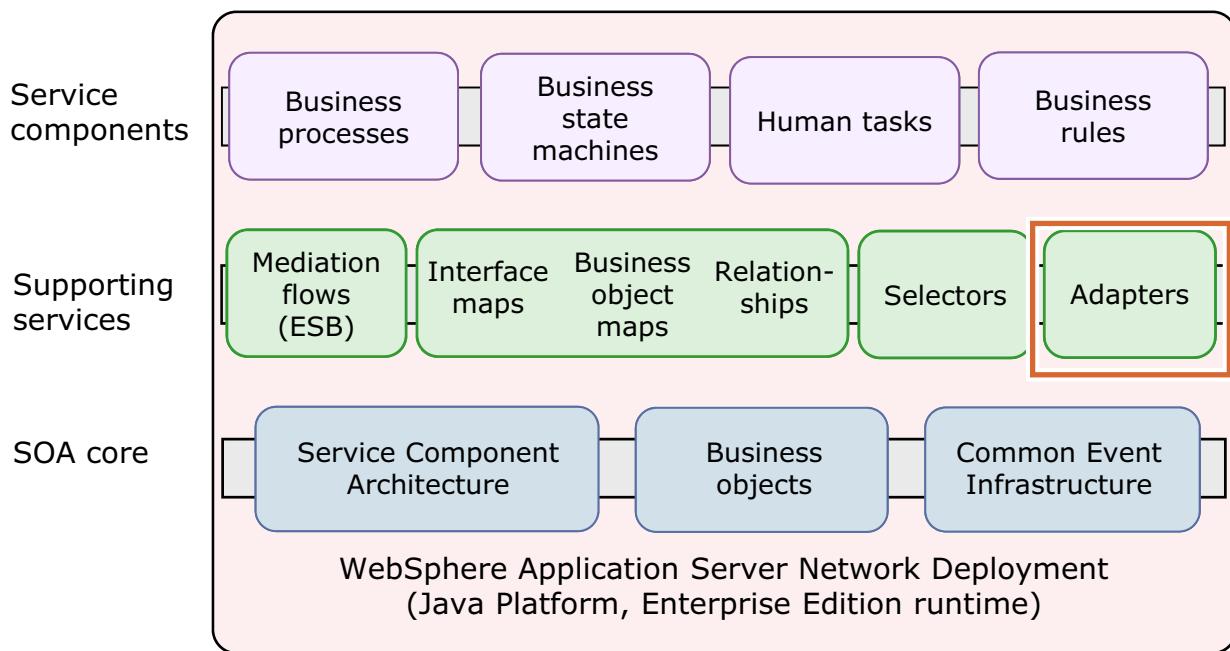
Figure 12-3. Introduction to adapters

WB8601.2

Notes:

Adapters are supporting services

- Adapters are part of the supporting services layer



© Copyright IBM Corporation 2016

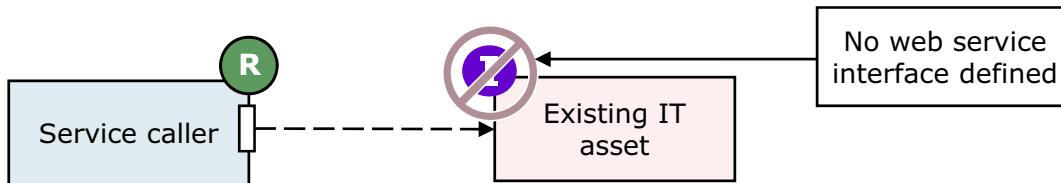
Figure 12-4. Adapters are supporting services

WB8601.2

Notes:

Introduction to adapters

- Adapters provide integration with enterprise information systems (EIS) without service interfaces
 - An EIS provides the information infrastructure for an enterprise by providing a set of services to clients:
 - Enterprise resource planning (ERP)
 - Customer relationship management (CRM)
 - Human resource systems (HR)
 - Adapters use existing IT assets in your SOA without significant programming, which simplifies integration
 - Adapters are presented as SCA components
 - EIS is invoked as a service through the adapter



© Copyright IBM Corporation 2016

Figure 12-5. Introduction to adapters

WB8601.2

Notes:

Adapters also provide other services, such as CEI event monitoring, transaction support, asynchronous communication, connection pooling, security (beyond web service), and a discovery utility for generating interfaces.

As you design and assemble your SOA solutions, on many occasions you must integrate SOA applications with an existing IT asset. Ideally, you are able to access that asset as a service just as you access other components. If the asset does not have a service interface, examine the environment to see whether the service interface can be added easily. On some occasions, the environment cannot directly support a service interface. You can use adapters to access and include such assets in your SOA solutions.

Organizations large and small invested in enterprise information systems over the past several decades, and as such are critically dependent upon their continued execution. Invariably any organization has a large proportion of its data that one, or more typically, many EIS systems manage. These systems fall into categories, such as ERP systems like SAP, and database systems like Oracle or DB2. Access to the data that these applications manage is provided through a set of exposed services, but adapters can also provide access to these services.

Adapter components

- Adapters connect EIS systems and applications to a central server
 - Adapter is a mediator between an asset and a broker (IBM Process Server)
- Adapters provide a layer of abstraction to simplify connectivity
- An adapter is divided into two components, usually in a single JAR file
 - One face looks like a service, providing a consistent framework for access to back-end systems and technologies
 - One face looks like a traditional user, providing application-specific access

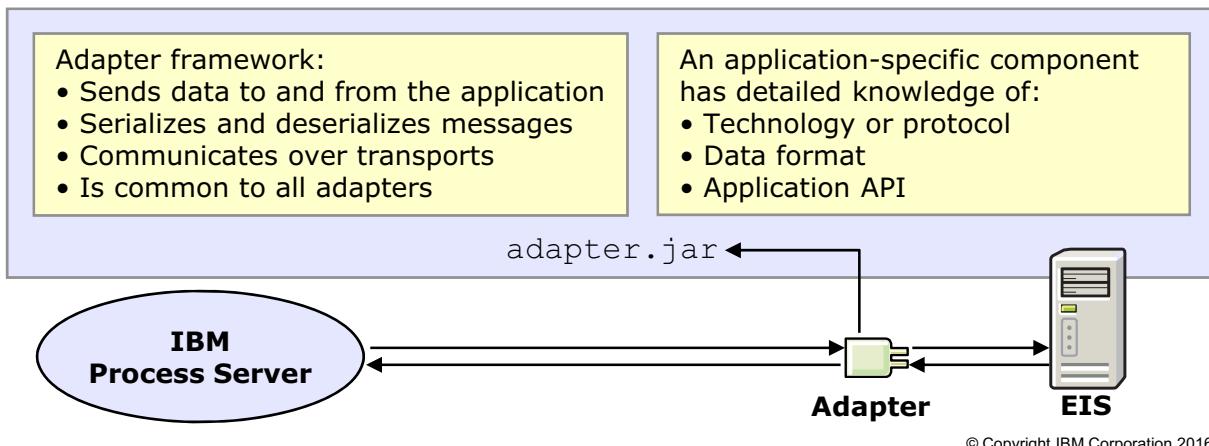


Figure 12-6. Adapter components

WB8601.2

Notes:

An adapter acts as a mediator between your SOA world and the asset. Access to the asset is not achieved directly. A component that is called an adapter presents a service interface to the SOA components, providing the service that the asset must expose. This interface is one face of the adapter. The other face of the adapter communicates directly with the asset by using whatever technology is appropriate for that asset. This communication can be achieved through proprietary APIs or various other techniques.

Adapters are specific to the version of the application for which they are written. If the application changes its API, a new adapter is developed. However, if the application data schema changes, the adapter code is not required to be changed. An advantage of the adapter framework is that when coding an adapter, it is not necessary for the developer to know the means of connecting with the message queues or configuration files. The framework automatically handles that connection. Instead, the developer writes the interaction only between the application programming interface and the adapter API methods to process business objects.

The adapter is written in the language that the API is written in, for example, SAPAdapter.jar written in Java or ClarifyAdapter.dll written in C/C++.

A toolkit is available for the creation of custom WebSphere Business Integration and JCA adapters. The framework for WebSphere Business Integration Adapters is proprietary but common among all WebSphere Business Integration Adapters.

Adapters are lightweight, distributed, metadata-driven components that are ready to run. They require no code modification, only configuration and transaction metadata in the form of business object definitions. One adapter can handle an object with multiple operations and multiple objects. Furthermore, business rules, transformation, and routing are delegated to the integration server: they are outside the scope of the adapter, which provides a common means for an integration server to interact with the application. The primary benefits are rapid deployment, ease of maintenance, and flexible distributed deployment.

WebSphere (JCA) Adapters

- WebSphere Adapters accelerate integration projects with a set of Java EE Connector Architecture (JCA) capabilities that service-enable your assets
 - Help in minimizing the need for integration coding and create standard interface points
- Two types of adapters: application and technology
- Application adapters connect to existing packaged applications so you can use data and services specific to the applications
 - All JCA application adapters are included in IBM Integration Designer for development use in the unit test environment (UTE)
- Technology adapters provide connectivity to data through technologies and protocols
 - All WebSphere Technology Adapters (JCA only) included in IBM Integration Designer are for development and production use with licensed copies of IBM Process Server

© Copyright IBM Corporation 2016

Figure 12-7. WebSphere (JCA) Adapters

WB8601.2

Notes:

Available WebSphere (JCA) application adapters

- SAP
 - IBM WebSphere Adapter for SAP Software: Offers bidirectional integration with SAP applications, providing comprehensive support for the broad range of SAP interfaces
- Oracle
 - IBM WebSphere Adapter for Oracle EBS: Offers comprehensive integration with the most commonly used EBS interfaces, including database access and interactions with XML Gateway and Oracle AQ
 - IBM WebSphere Adapter for PeopleSoft: Offers bidirectional, real-time integration to PeopleSoft Enterprise
 - IBM WebSphere Adapter for Siebel: Offers bidirectional integration with Siebel Business Applications
 - IBM WebSphere Adapter for JD Edwards: Offers bidirectional integration to the JD Edwards EnterpriseOne servers

© Copyright IBM Corporation 2016

Figure 12-8. Available WebSphere (JCA) application adapters

WB8601.2

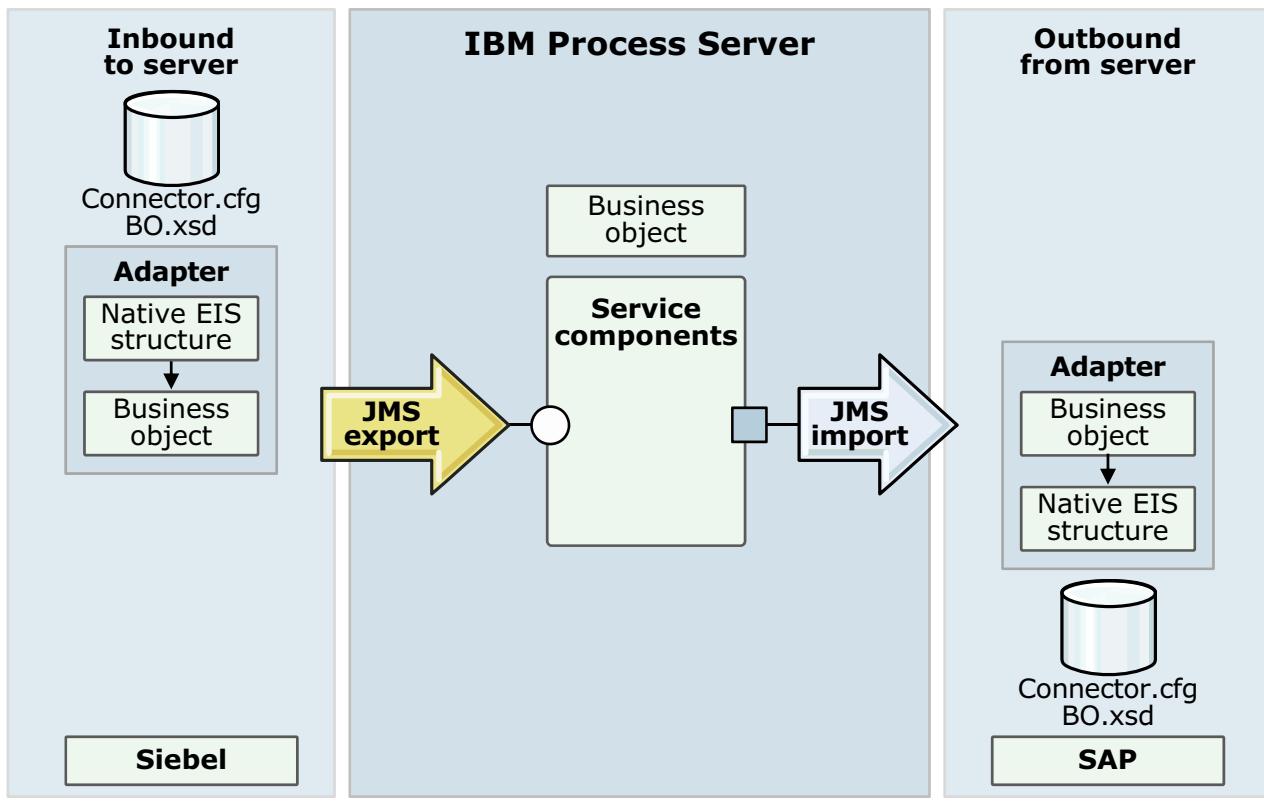
Notes:

IBM WebSphere Adapters accelerate integration projects with a set of Java Connector Architecture (JCA) capabilities that service-enable your assets. They help you minimize the need for integration coding and create standard interface points. WebSphere Adapters help extend service-oriented architecture (SOA) applications beyond organizational walls to customers, partners, and suppliers. They can reduce maintenance and development costs while optimizing and renewing the value of your enterprise assets.

For more information on application adapters, go to:

<http://www.ibm.com/software/products/en/adapters>

Architecture



© Copyright IBM Corporation 2016

Figure 12-9. Architecture

WB8601.2

Notes:

Available WebSphere (JCA) technology adapters

- IBM WebSphere Adapter for Flat Files
 - Bidirectional integration through the exchange of local flat files
- IBM WebSphere Adapter for JDBC
 - Bidirectional JDBC data source integration of tables, views, and stored procedures
- IBM WebSphere Adapter for FTP
 - Bidirectional exchange of remote flat files with support for FTP over SSL
- IBM WebSphere Adapter for Email
 - Bidirectional integration through exchange of email by using mail systems
 - Inbound support for IMAP and POP3, and outbound support for SMTP
 - Support for transformation of content or attachments, and the local mail archive
- IBM WebSphere Adapter for IBM System i
 - Provides bidirectional integration to native i5/O applications, including RPG programs on the IBM i platform
- IBM WebSphere Adapter for Lotus Domino
 - Bidirectional exchange of business data (calendar entry, task list item, or note) between SCA applications and Domino servers)

© Copyright IBM Corporation 2016

Figure 12-10. Available WebSphere (JCA) technology adapters

WB8601.2

Notes:

IBM WebSphere Adapter for Technology Standard delivers connectivity by using technology standards, protocols, and database. It uses the Java Connector Architecture (JCA) and enterprise metadata discovery specifications to provide integration with graphical discovery tools without writing code.

For more information on technology adapters, go to:

<http://www.ibm.com/software/products/en/websphere-adapter-for-technology>

Advantages of Java EE Connector Architecture (1 of 2)

- JCA provides a Java open standard for EIS connectivity
- JCA provides added qualities of service
- Connection management
 - Application server JCA container maintains a reusable connection pool
 - Provides a scalable application environment that can support many EIS clients
- Transaction management for outbound transactions
 - Distributed transactions (through XA)
 - Local transactions
 - Support is dependent upon EIS
- Inbound transactions: Assured event delivery
 - Ensures “once and only once” delivery of events (requires acknowledgment)
 - Works even if EIS does not support transactions (through event staging)
- Security management

© Copyright IBM Corporation 2016

Figure 12-11. Advantages of Java EE Connector Architecture (1 of 2)

WB8601.2

Notes:

The JCA V1.5 specification has several advantages. First, it is an industry standard (which different vendors widely accept). Second, by moving into the Java EE environment, more quality of service qualifiers are provided through the WebSphere Resource Adapter container.

JCA provides a non-proprietary framework for creating a Java EE adapter component. JCA is a Java EE standard that was created to normalize the way Java applications access enterprise applications. Customers can use the JCA resource adapter to integrate business functions and to provide a centralized source for the application resource adapter development. A resource adapter is the component of the JCA that provides EIS connectivity.

The following list contains more characteristics of the JCA specification:

- **Connection pooling:** JCA container maintains a pool of connections
- **Security:** End-to-end Java EE security model
- **Transaction support** (the following are supported):
 - Outbound:
 - XA (distributed transactions)

- Local transactions
- Inbound (assured event delivery):
 - Ensures “once and only once” delivery of events
 - Works even if EIS does not support transactions (through an optional event staging database)

Advantages of Java EE Connector Architecture (2 of 2)

- Open standard provides faster time-to-market and lower maintenance
 - JCA designed for adoption by independent software vendors
- JCA adapters are multithreaded and support two kinds of event delivery
- Ordered delivery:
 - Single thread delivers one event at a time
 - Events arrive in the order they were created
 - Event ordering takes precedence over throughput
- Unordered delivery:
 - Multiple threads deliver events on several channels simultaneously
 - Event order is not kept
 - Higher throughput takes precedence over event order

© Copyright IBM Corporation 2016

Figure 12-12. Advantages of Java EE Connector Architecture (2 of 2)

WB8601.2

Notes:

If the delivery type is set to “ORDERED,” an instance of the adapter runs in a single thread that delivers the events in the order that they were created. If the delivery type is set to “UNORDERED,” events get delivered simultaneously and the delivery order is not guaranteed. The benefit of “UNORDERED” is higher throughput. The default is “ORDERED.”

12.2.Using WebSphere (JCA) Adapters

Using WebSphere (JCA) Adapters



© Copyright IBM Corporation 2016

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

10.1

Figure 12-13. Using WebSphere (JCA) Adapters

WB8601.2

Notes:

Using WebSphere Adapters

- The External Service wizard is used to discover connectivity information about an application or technology
- WebSphere Adapters use Enterprise Metadata Discovery (EMD) inside the External Service wizard to introspect an EIS
 - EMD defines common API adapter uses to expose EIS services and business objects to application development tools
 - Single tool supports multiple EIS systems and multiple adapters
 - EMD is provided as an IBM Integration Designer plug-in
- WebSphere Adapters use the Data Exchange Service Provider Interface (DESPI) to transfer data between adapter components and the IBM Process Server runtime
 - DESPI is an IBM extension to JCA
 - It is used to transfer data between adapters and various runtimes
 - Provides runtime data format independence

© Copyright IBM Corporation 2016

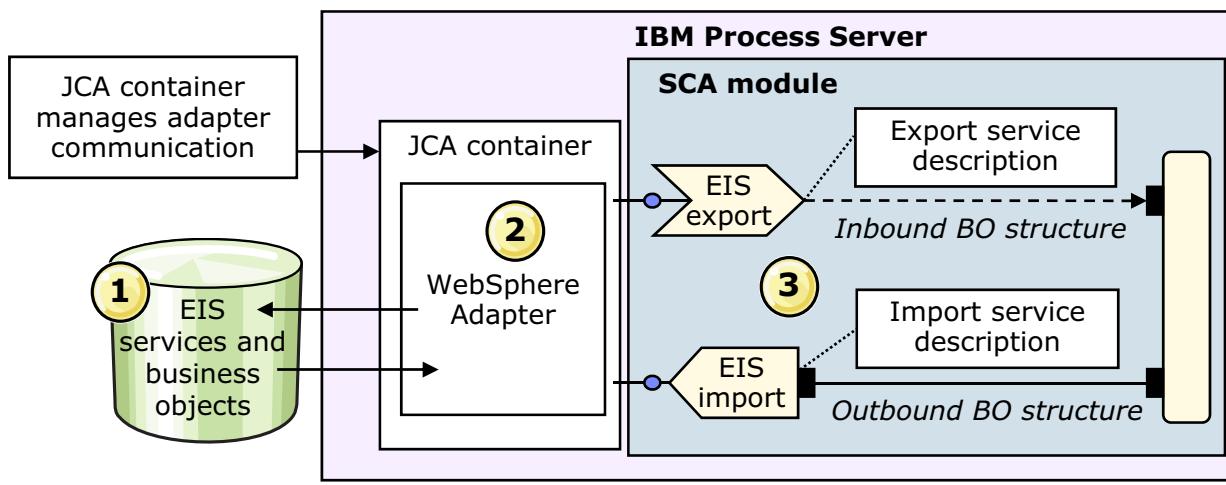
Figure 12-14. Using WebSphere Adapters

WB8601.2

Notes:

Enterprise Metadata Discovery

- When the External Service wizard is run:
 - EMD introspects the EIS
 - User selects entities of interest (interfaces or business objects) for use with the adapter project
 - EIS bound imports and exports, WSDL interfaces, and business object definitions are generated and packaged for deployment in the SCA module



© Copyright IBM Corporation 2016

Figure 12-15. Enterprise Metadata Discovery

WB8601.2

Notes:

Enterprise Metadata Discovery provides a rich environment for building and managing enterprise class applications and solutions. With the rise of SOA, more enterprises are deploying SOA solutions in a Java EE environment, but find it necessary to service-enable their existing IT assets to realize the full benefits of business automation. IBM and BEA collaborated on specifications that provide more productive development and deployment of this “last stage” of integration. The “Enterprise Metadata Discovery Specification for J2CA” is a cross-industry initiative started by IBM and BEA, and supported by adapter vendors and application vendor partners (it is not part of the original JCA specification). This new specification provides a standard way of managing this “last mile” with a seamless design-time experience (by using a top-down approach), thus allowing customers significant productivity improvements in service-enabling their existing IT assets.

The Enterprise Metadata Discovery specification intends to solve the problem of standard rich tool interfaces for adapters. Thus it unlocks the potential of the integration industry to deliver higher-quality adapters to more end systems at lower costs. Adapter vendors can focus on building more and better adapters, without worrying about interoperability of proprietary extensions. Infrastructure vendors can focus on building better platforms and better development experiences. Application vendors can build their own “last mile” components, confident that a fully standard implementation addresses both runtime and design-time interoperability requirements. As a result,

any adapter product can plug seamlessly into any Java EE SOA implementation, delivering both superior end-to-end runtime behavior and developer productivity.

The result to customers is greater adapter availability and quality at a lower cost. Existing IT assets can be a service that is enabled more readily, thus reducing the total cost of ownership and improving time-to-value of an SOA implementation. Moreover, each component in the end-to-end solution is built to agreed-upon specifications as opposed to proprietary extensions, thus reducing vendor lock-in.

For more information about the EMD specification, see:

<http://www.ibm.com/developerworks/webservices/library/ws-soa-eisjca/#EMD>

The EMD tool (through the external service wizard in IBM Integration Designer) does the following tasks:

1. The adapter metadata discovery service introspects the EIS and examines its data structure.
2. Based on what is discovered, the EMD tool generates definitions for business objects, imports and exports, and WSDL interfaces.

When an adapter is used, the module that contains the adapter has an export or import component that is bound to the adapter. The WebSphere Application Server JCA container manages the communication between the adapter and the enterprise application or technology.

Binding is done at three levels:

- **Interface binding:** Describes the mapping between the EIS service connection information and the service interface
- **Method binding:** Expresses the relationship between interface operations and service interaction or event information
- **Data binding:** Defines how the service input or output data is mapped to the native data as understood by the EIS service

For SCA clients, the function of the adapter is exposed through a WSDL interface. An SCA client invokes the adapter service through the `EISImportBinding`, which EMD produces. The inbound service description is an `EISExportBinding`, which EMD also produces. EMD generates business objects, which both SCA components and the adapter use. Using SCA components and business objects to interact with adapters fits the goals and the vision of SOA solutions.



Discovered interactions

- The EMD tool discovers the interaction styles of the EIS based on your input in the External Service wizard
- Two types of interaction styles:
 - Outbound: Client initiated
 - Inbound: EIS initiated
- Two modes of interaction for each style: Request/response or one-way
- Request/response interaction
 - Request/response interaction takes a request and returns a response
 - Generated method includes both an input and an output
- One-way interaction
 - Takes a request, but does not return a response
 - Generated method includes only an input

© Copyright IBM Corporation 2016

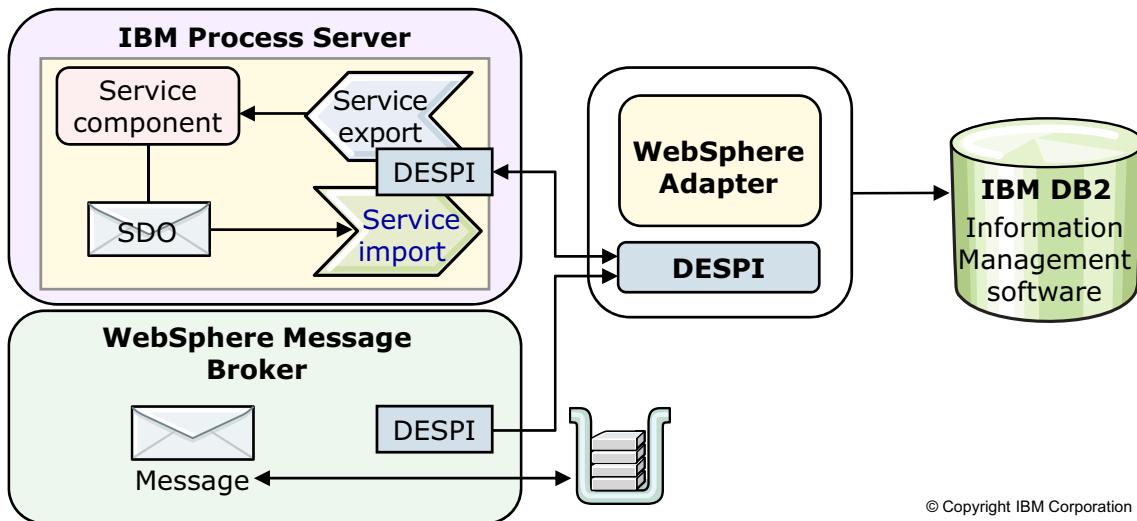
Figure 12-16. Discovered interactions

WB8601.2

Notes:

Data Exchange Service Provider Interface

- Adapters handle data internally, in a format-independent manner, by using DESPI
 - An export or import data binding produces and consumes business objects and communicates with the adapter by using the Data Exchange SPI
 - The DESPI interface decouples the adapter from data representations that the EIS and the business object data model understand



© Copyright IBM Corporation 2016

Figure 12-17. Data Exchange Service Provider Interface

WB8601.2

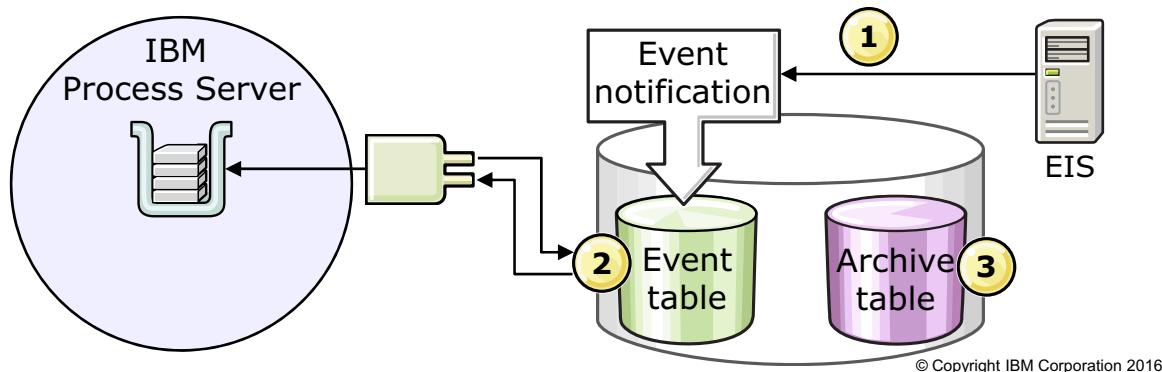
Notes:

DESPI is the interface that resource adapters and runtime components use to exchange business object data. The Data Exchange SPI architecture, which is based on the concept of cursors and accessors, abstracts the data type so that an adapter can be written only once. It can then work on runtime environments that support different data types, such as data objects and JavaBeans. A DESPI architectural diagram is included in the appendices.

Application adapters and event polling

Most application adapters require event polling in the EIS

1. A database trigger or script creates an event in the event table
2. The adapter polls the event table for new items
 - A business object that is the subject of the event is retrieved
 - The business object is placed in a queue for processing
3. When processed, the adapter moves the event from the event table to the archive table (for auditing)
 - Archive table cleanup is done manually by using provided scripts



© Copyright IBM Corporation 2016

Figure 12-18. Application adapters and event polling

WB8601.2

Notes:

JCA application adapters often require modification of the EIS so that the application generates an “adapter-identifiable” event. Modification includes:

- Installation of scripts
- Creation of database triggers and tables
- Addition of objects and tools

IBM adapters have components and scripts to build the schema and tables that the adapter needs for event notification on a particular application. If an adapter is publishing events, the corresponding application needs the following items:

- A storage location (an event table) in which to cache key event data (a database table or directory); event table polling speed is configurable
- A mechanism to capture changes in the application and create an event (triggering mechanism: database trigger, script, or workflow)
- An archive location to note events that are successfully sent to the integration tool (database or directory)



Unit summary

- Describe the purpose and business value of using adapters in applications
- Describe the capabilities of WebSphere (JCA) adapters
- List the advantages of using the JCA architecture for WebSphere Adapters
- Describe how to use the External Service wizard for WebSphere Adapters

© Copyright IBM Corporation 2016

Figure 12-19. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. True or False: Adapters provide integration with enterprise information systems (EIS) that have clearly defined service interfaces.
2. True or False: The External Service wizard discovers and generates the necessary SCA artifacts, including business objects, interfaces, and import and export components.
3. True or False: Technology adapters connect to existing packaged applications so you can use data and services specific to the applications.

© Copyright IBM Corporation 2016

Figure 12-20. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.

Checkpoint answers

1. False. One of the key advantages of adapters is that they provide integration with enterprise information systems (EIS) that do not provide service interfaces.
2. True.
3. False. Application adapters connect to existing packaged applications so you can use data and services specific to the applications

© Copyright IBM Corporation 2016

Figure 12-21. Checkpoint answers

WB8601.2

Notes:

Exercise 10



Implementing WebSphere (JCA) adapters

© Copyright IBM Corporation 2016

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

10.1

Figure 12-22. Exercise 10

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Configure the WebSphere Adapter for Flat Files
- Use the external service tool to generate artifacts that are used in an application
- Incorporate adapter-related SCA artifacts in an assembly diagram
- Test an adapter in the IBM Integration Designer test environment

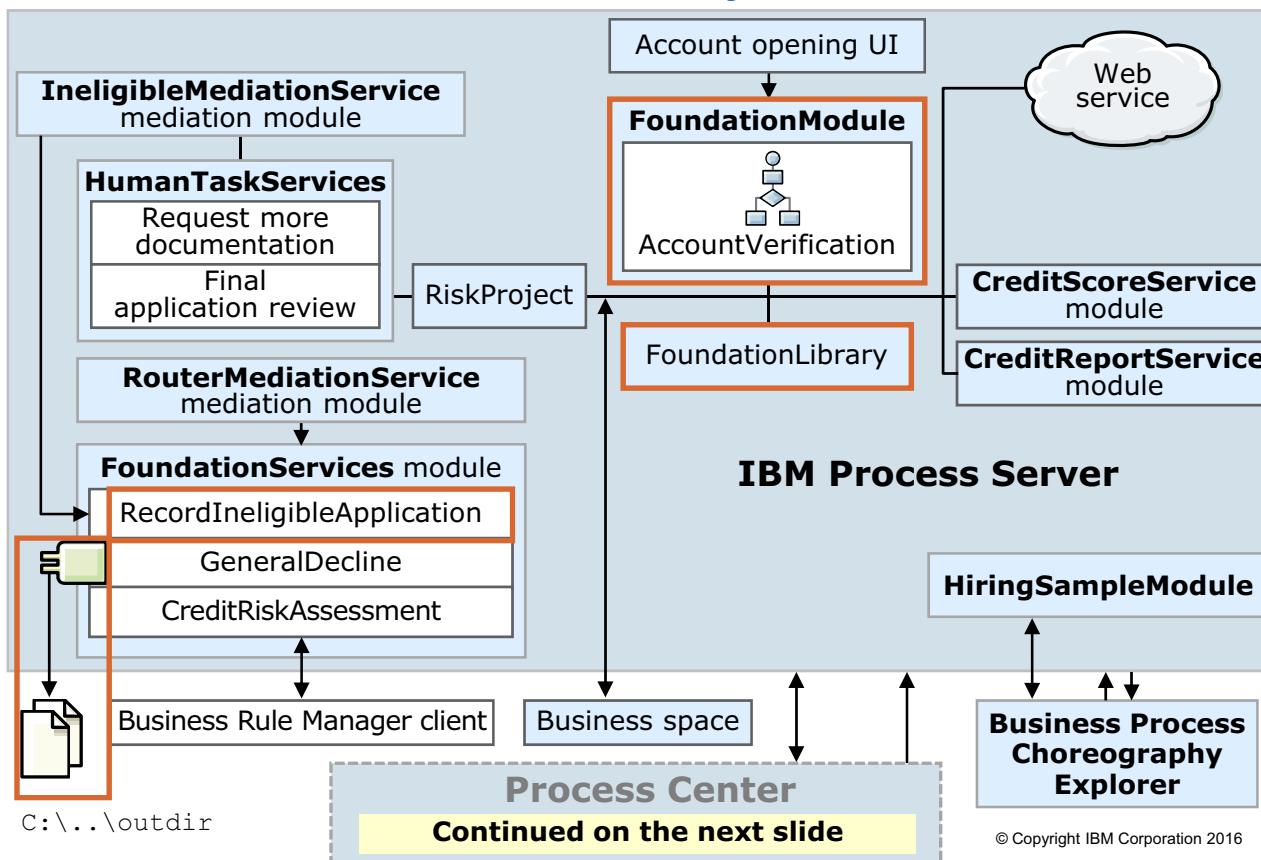
© Copyright IBM Corporation 2016

Figure 12-23. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 10



© Copyright IBM Corporation 2016

Figure 12-24. Course business scenario: What you work with in Exercise 10

WB8601.2

Notes:

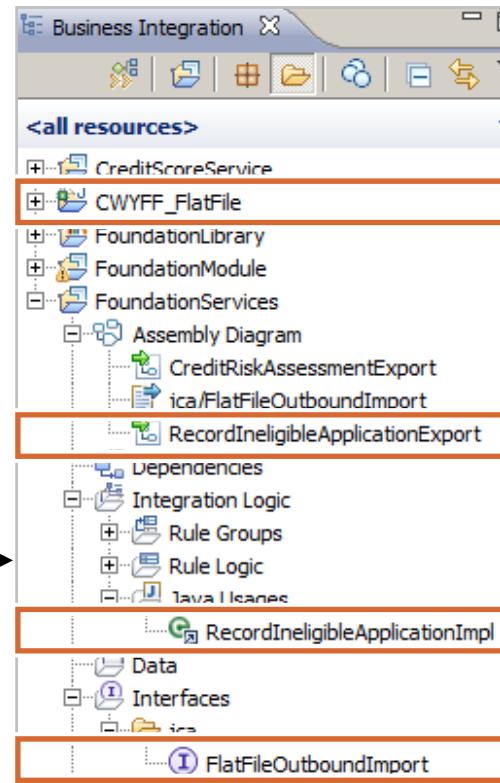
Components that are required for Exercise 10 (1 of 2)

Prebuilt components that are imported in the lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. FoundationServices** that you built in Exercise 9

New components that you create in this lab:

- 1. CWYFF_FlatFile** connector project
- 2. FlatFileOutboundImport** import component
- 3. RecordIneligibleApplicationExport** export component
- 4. RecordIneligibleApplicationImpl.java** component



© Copyright IBM Corporation 2016

Figure 12-25. Components that are required for Exercise 10 (1 of 2)

WB8601.2

Notes:

IBM WebSphere Adapters make it possible for Java Platform, Enterprise Edition (Java EE) components, such as applications, to communicate with enterprise information system (EIS) resources. An EIS is the information infrastructure for an enterprise (for example, an enterprise resource planning system). A WebSphere adapter acts as an intermediary between the Java EE component and the EIS. This way, it is not necessary for the Java EE component to understand the low-level API or data structures of the EIS.

The IBM WebSphere Adapter for Flat Files facilitates the exchange of business data in the form of delimited records between file systems and Java EE applications. The adapter supports inbound and outbound operations and the use of business objects, business components, and business services.

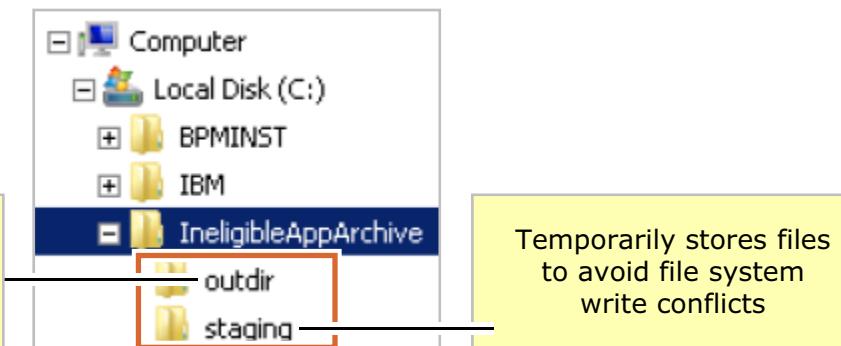
In this exercise, you implement a service to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files. The AccountVerification process invokes the service when the eligibleApplication attribute is set to false.

See the process application model that was created previously. In this exercise, you implement the Record Ineligible Application activity. The purpose of this activity is to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files.

Components that are required for Exercise 10 (2 of 2)

Two new directories that you create in this lab:

1. The `C:\IneligibleAppArchive\staging` directory
2. The `C:\IneligibleAppArchive\outdir` output directory
 - The `IneligibleApplication` business object is written to this output directory



© Copyright IBM Corporation 2016

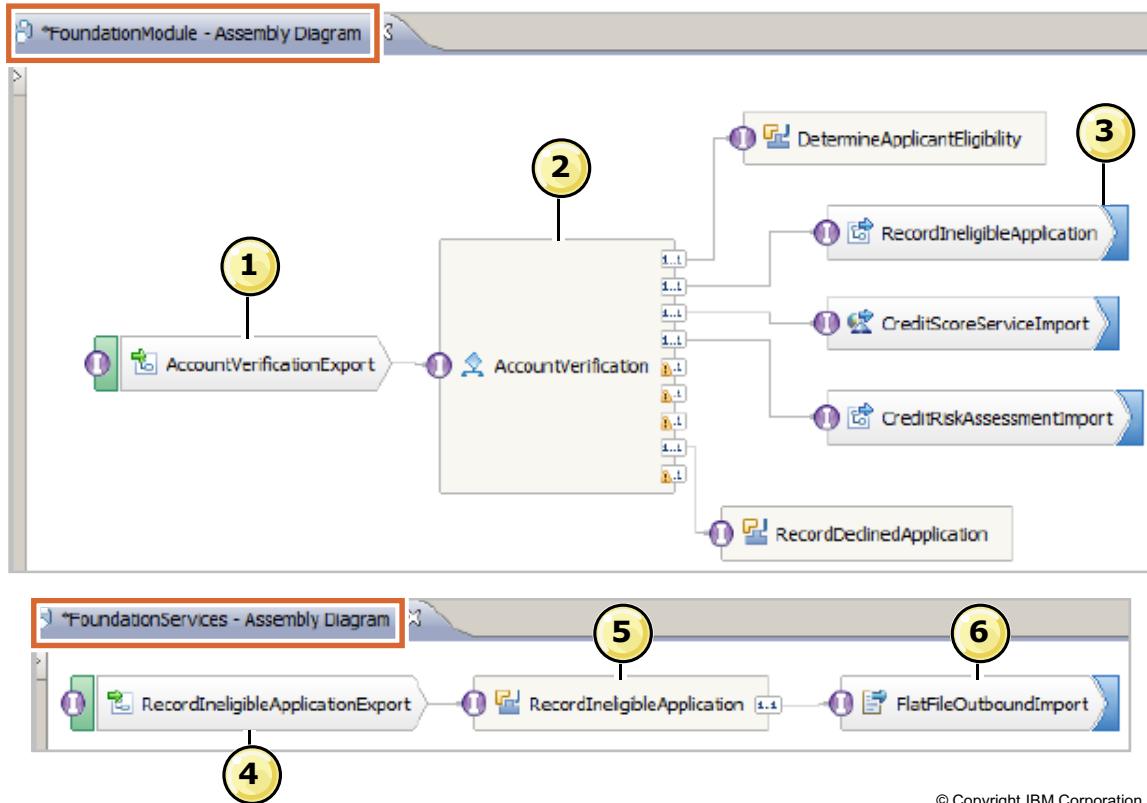
Figure 12-26. Components that are required for Exercise 10 (2 of 2)

WB8601.2

Notes:

You run the External Service wizard to configure the WebSphere Adapter for Flat Files. The adapter uses the directories that you examined previously. The adapter code is deployed inside the FoundationServices module, and a `FlatFileOutboundImport` component is created on the FoundationServices assembly diagram. The `FlatFileOutboundImport` component invokes the `recordIneligibleApplication` operation to write an `IneligibleApplication.txt` file to `C:\IneligibleAppArchive\outdir`.

Assemble SCA components in Exercise 10



© Copyright IBM Corporation 2016

Figure 12-27. Assemble SCA components in Exercise 10

WB8601.2

Notes:

Finally, after creating the flat file adapter component, you assemble the SCA components in IBM Integration Designer and then test the application.

The following steps are illustrated in the diagram:

1. The AccountVerificationExport component exposes the AccountVerification business process.
2. When the application is ineligible, the AccountVerification process needs to record the ineligible application in the database and terminate the process. It calls the RecordIneligibleApplication import component.
3. The RecordIneligibleApplication import component is used to call the application or service outside FoundationModule. In this scenario, it calls the RecordIneligibleExport component in the FoundationServices module.
4. The RecordIneligibleExport component calls the RecordIneligibleApplication Java component.
5. The RecordIneligibleApplication component is a Java component that creates the output message.

"Account verification recorded this application as ineligible for the customer <company name>" is going to be recorded to the system. In this scenario, it calls the FlatFileOutboundImport component.

6. The FlatFileOutboundImport component writes the output message to the file system. In this scenario, it writes to a text file in the `C:\IneligibleAppArchive\outdir` output directory.



References

- WebSphere Business Integration Adapter Information Center:
 - <http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>
- Integrate WebSphere Business Integration Adapters with WebSphere Process Server Version 6: Application Event Notification (AgentDelivery) scenario:
 - http://www.ibm.com/developerworks/websphere/library/techarticles/0601_reddy/0601_reddy.html
- Java Connector Architecture:
 - <http://java.sun.com/j2ee/connector/index.jsp>
- IBM Redbooks for WebSphere Adapter Development:
 - <http://www.redbooks.ibm.com/abstracts/sg246387.html>
- WebSphere Adapters home page:
 - <http://www.ibm.com/software/integration/wbiadapters>

© Copyright IBM Corporation 2016

Figure 12-28. References

WB8601.2

Notes:

Unit 13. Developing mediation services

What this unit is about

This unit covers mediation services, mediation modules, and service message objects (SMOs) in IBM Process Server.

What you should be able to do

After completing this unit, you should be able to:

- Describe the role of mediation services in IBM Process Server
- Define the concept of mediation modules
- Describe how to create mediation flows in IBM Integration Designer
- Describe the role of SMOs in mediations
- Explain the structure of SMOs

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Describe the role of mediation services in IBM Process Server
- Define the concept of mediation modules
- Describe how to create mediation flows in IBM Integration Designer
- Describe the role of SMOs in mediations
- Explain the structure of SMOs

© Copyright IBM Corporation 2016

Figure 13-1. Unit objectives

WB8601.2

Notes:



Topics

- Mediation services
- Service message objects

© Copyright IBM Corporation 2016

Figure 13-2. Topics

WB8601.2

Notes:

13.1.Mediation services

Mediation services



© Copyright IBM Corporation 2016

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

10.1

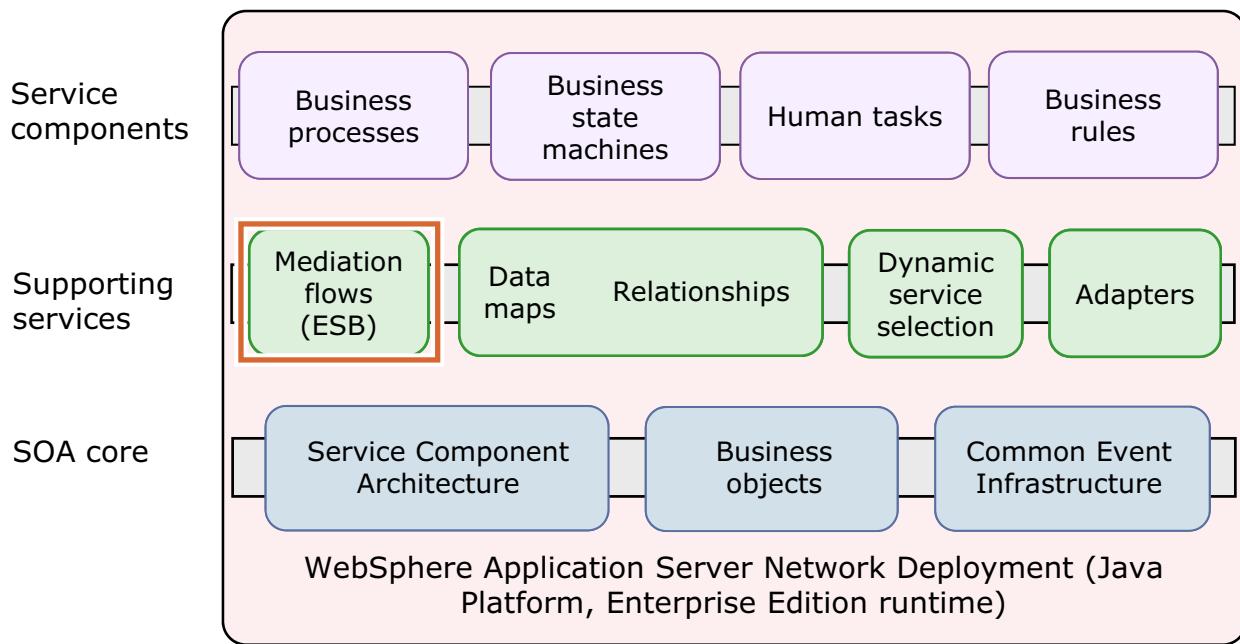
Figure 13-3. Mediation services

WB8601.2

Notes:

Mediation flows are supporting services

- Mediation modules can be deployed to IBM Process Server
- No equivalent support in IBM Process Designer



© Copyright IBM Corporation 2016

Figure 13-4. Mediation flows are supporting services

WB8601.2

Notes:

Above the SOA core, IBM built a set of added services that provide higher-level abstractions and ease-of-use capabilities. These functions are mediation flows, data maps, relationships, dynamic service selection components (selectors, mediation primitives), and adapters. Except for selectors, these components can be used in both IBM Process Server business integration modules and mediation modules.

What does an enterprise service bus do? (1 of 2)

- An ESB supplies a communication layer to support service interactions
 - It should support communication through various protocols

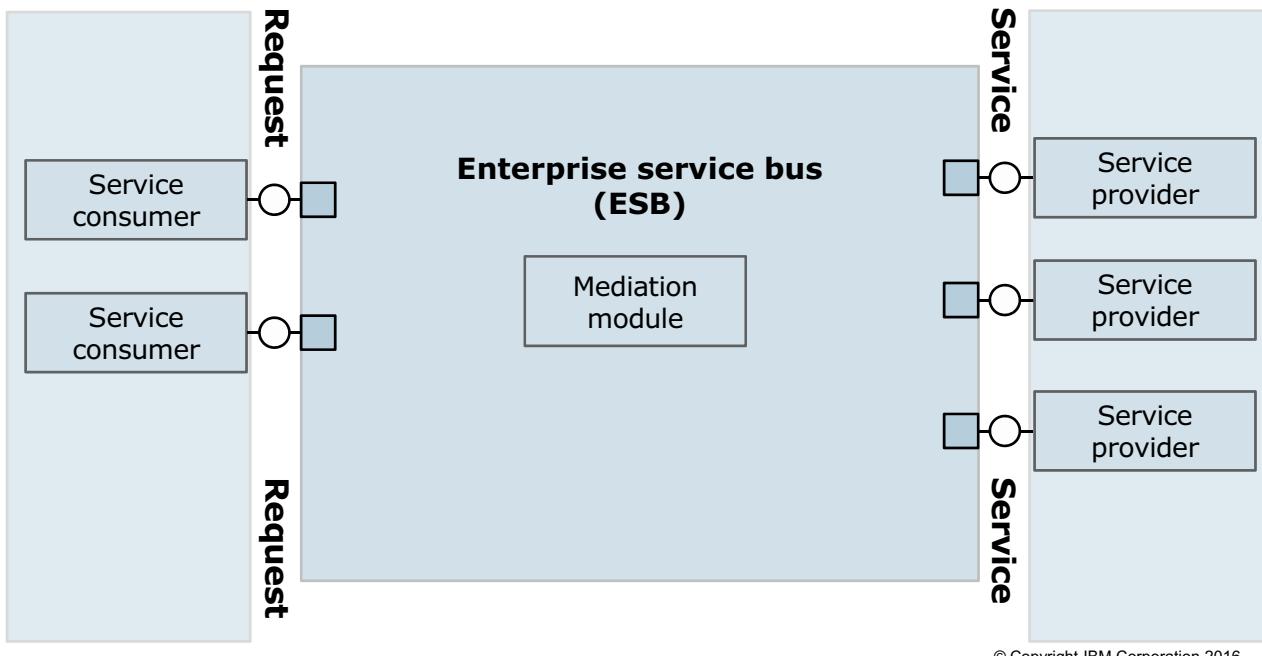


Figure 13-5. What does an enterprise service bus do? (1 of 2)

WB8601.2

Notes:

In the SOA realm, services do not communicate directly with each other. The enterprise service bus receives the messages from the service requester; it does the required routing, converting, and transforming of the message, and then passes it to the service provider.

IBM Business Process Manager Advanced provides one of the keys to helping you achieve the goals of SOA. It provides a flexible connectivity infrastructure for integrating applications and services, enabling composite applications to be built as a loose coupling of independent services. It reduces the number, size, and complexity of interfaces and connections that must be defined and maintained.

What does an enterprise service bus do? (2 of 2)

- ESB reduces the tight coupling between the service consumer and provider
- The service consumer is **not** affected if:
 - The service provider location changes
 - The service provider interface changes
 - A different service provider is used
- An ESB does the following actions between the service consumer and service provider:
 - **Route** messages between services
 - **Convert** transport protocols between consumer and provider
 - **Transform** message formats between consumer and provider
 - **Handle** business events from disparate sources

© Copyright IBM Corporation 2016

Figure 13-6. What does an enterprise service bus do? (2 of 2)

WB8601.2

Notes:

An enterprise service bus provides four primary functions:

- **Routing** messages: The requester sends the request to the ESB, and the ESB is responsible for calling the appropriate service provider.
- **Converting** transport protocols: The ESB allows the service requester to use one transport protocol while the service provider uses another.
- **Transforming** message formats: ESB eliminates the direct call from the service requester to the service provider. In this way, the ESB can modify the message, so the interfaces that the requester and provider use do not have to be identical.
- **Handling** business events: Events can be handled from disparate sources.

Key concepts

- Mediation module
 - Special type of SCA module
 - Mediate messages that flow between service requesters and providers
- Mediation flow component
 - Contains the mediation flow logic
 - Unique flow logic for every interface operation
 - Modules can contain zero or multiple mediation flow components
- Mediation primitives
 - Are used to construct the logic of a mediation flow
 - Each primitive does a specific part of the flow logic
 - An encapsulated unit of logic that manipulates the message as it passes through the enterprise service bus
- Service message object (SMO)
 - Internal representation of message body and headers
 - Mediation primitives act upon the SMO within the mediation flow

© Copyright IBM Corporation 2016

Figure 13-7. Key concepts

WB8601.2

Notes:

Mediation modules are a special type of SCA modules that can change the format, content, or target of service requests. They use SCA exports and imports to communicate with service requesters and service providers, which provide the key to handling protocol conversions within the bus.

The mediation module also contains a mediation flow component. The mediation flow component contains the logic for the mediation. For every operation defined on an input interface, a unique mediation flow logic is defined for the request and response of the operation. The mediation flow logic transforms and dynamically routes messages.

The flow logic is defined in mediation flow components by using mediation primitives. Each mediation primitive provides some specific portion of the logic and is wired to other primitives into a logical flow. A service message object represents the data that mediation primitives manipulate.

Example: Mediation module and mediation flow

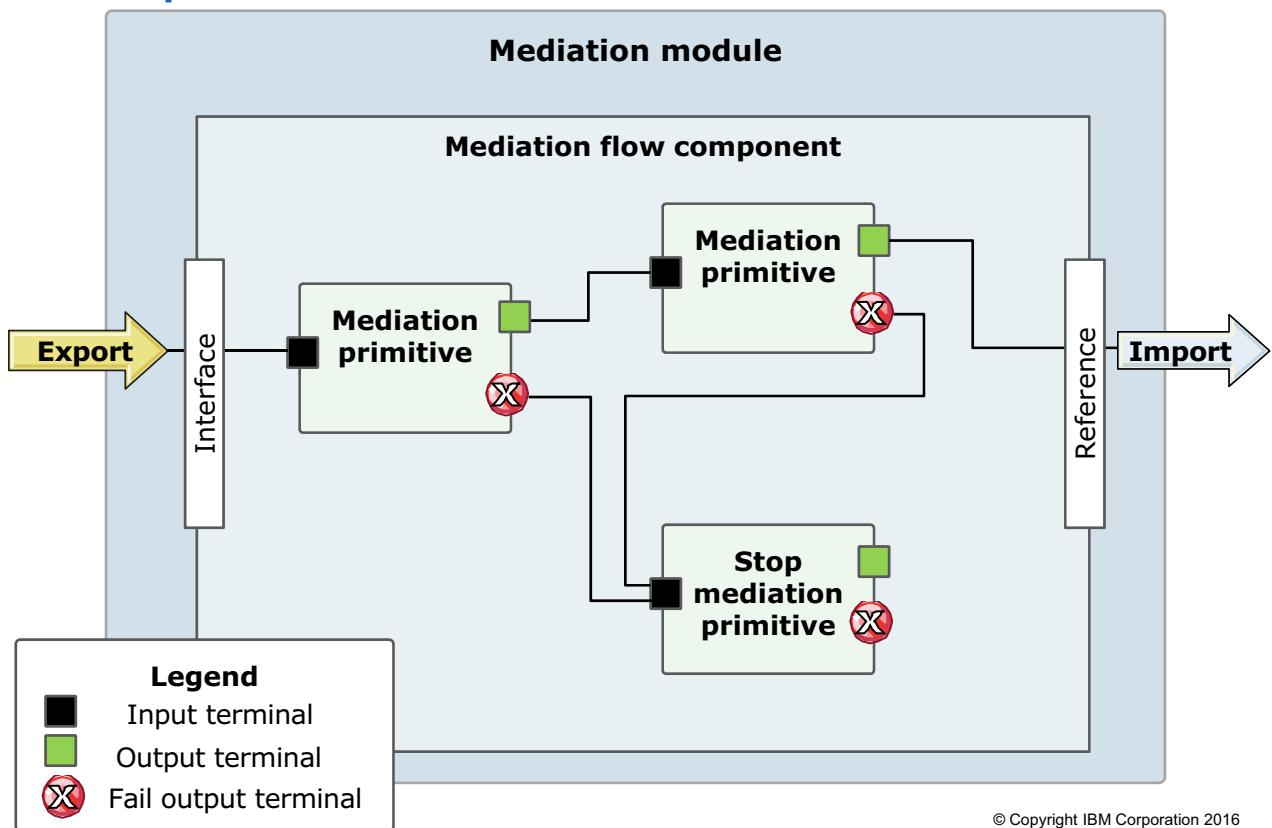


Figure 13-8. Example: Mediation module and mediation flow

WB8601.2

Notes:

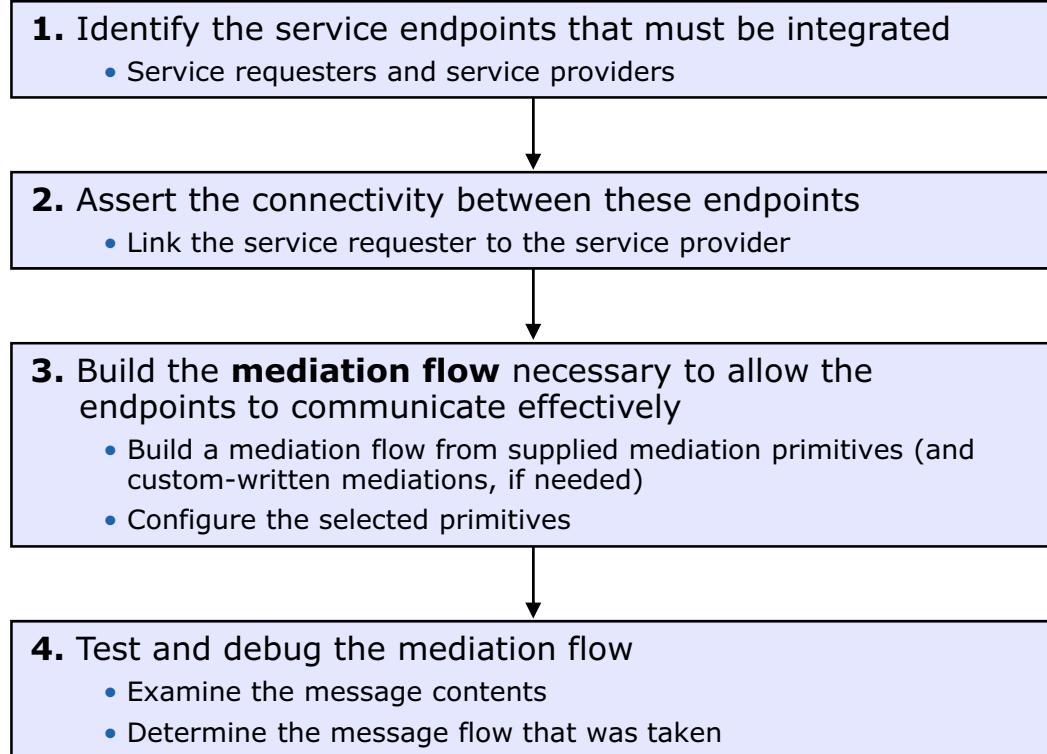
The diagram shows a simplified example of a mediation module. The mediation module contains one mediation flow component, which contains mediation primitives.

Usually, mediation modules contain a specific type of SCA component that is called a *mediation flow component*. Mediation flow components define mediation flows.

A mediation flow component can contain none, one, or a number of mediation primitives. IBM Business Process Manager supports a supplied set of mediation primitives that provide functions for message routing and transformation. For more mediation primitive flexibility, use the Custom Mediation primitive to call custom logic.

The purpose of a mediation module that does not contain a mediation flow component is to transform service requests from one protocol to another. For example, a service request might be made by using SOAP/JMS but might need transforming to SOAP/HTTP before sending on. You can view and make certain changes to mediation modules from IBM Business Process Manager. However, you cannot view or change the SCA components inside a module from IBM Business Process Manager. Use Integration Designer to customize SCA components.

Typical task to implement a mediation flow



© Copyright IBM Corporation 2016

Figure 13-9. Typical task to implement a mediation flow

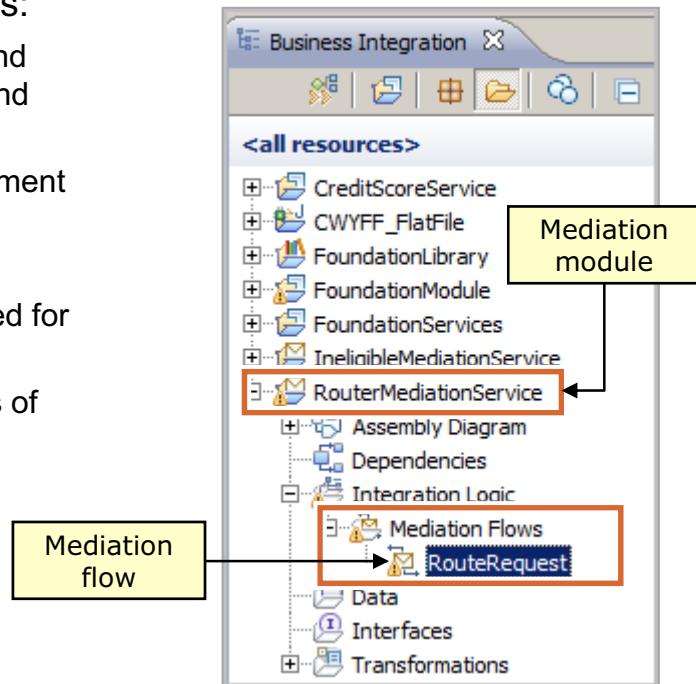
WB8601.2

Notes:

This visual describes the typical tasks that an integration designer does when developing an integration solution by using a mediation flow.

Mediation module and mediation flow: Tools (1 of 2)

- IBM Integration Designer tools:
 - Easy-to-use tools for defining and managing mediation modules and mediation flows
 - Eclipse-based tools for development
 - Focused on the more technical developer role
 - Business Integration view is used for interaction with mediation flow
 - You can change certain aspects of mediation modules from the administrative console without having to redeploy the module



© Copyright IBM Corporation 2016

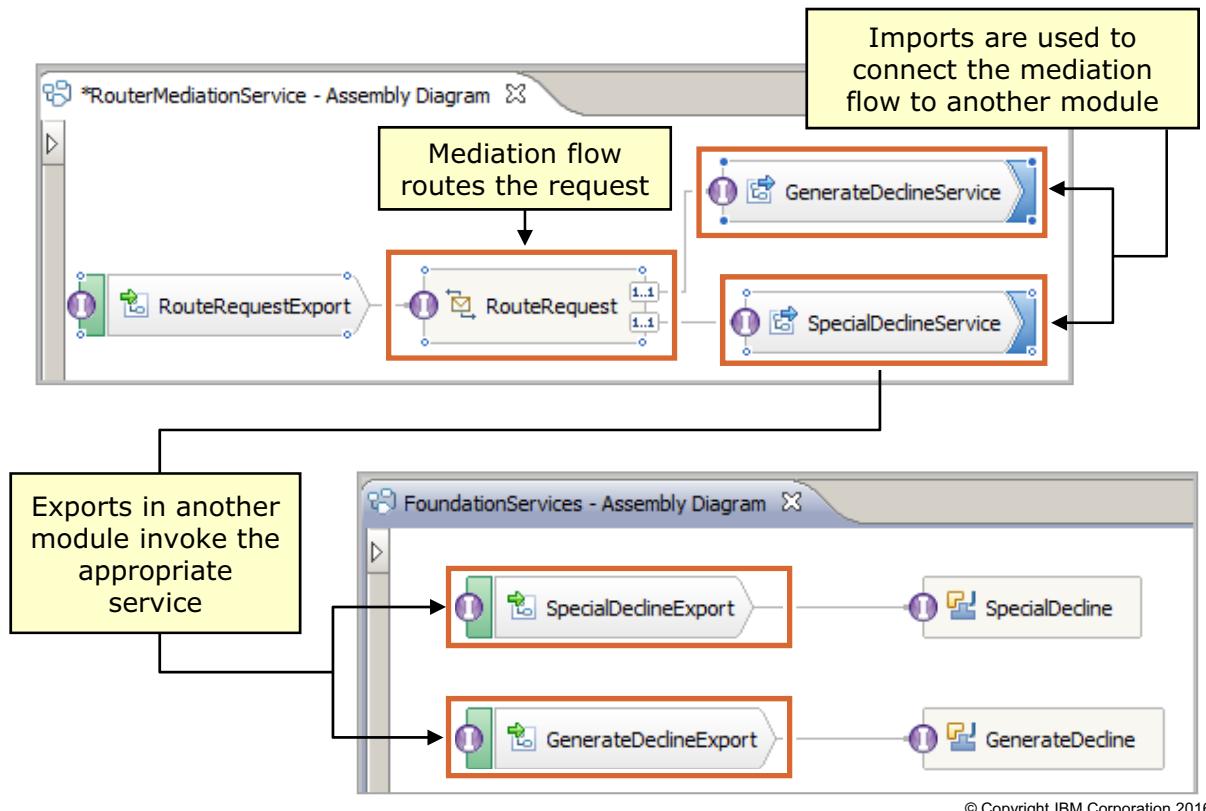
Figure 13-10. Mediation module and mediation flow: Tools (1 of 2)

WB8601.2

Notes:

For mediation primitive properties to be visible from the IBM Business Process Manager administrative console, the integration developer must promote the properties. Certain properties lend themselves to being administratively configured, and Integration Designer describes these properties as promotable properties because they can be promoted from the integration cycle to the administrative cycle. Other properties are not suitable for administrative configuration because modifying them can affect the mediation flow in such a way that the mediation module needs to be redeployed. Integration Designer lists the properties that you can choose to promote under the promoted properties of a mediation primitive.

Mediation module and mediation flow: Tools (2 of 2)



© Copyright IBM Corporation 2016

Figure 13-11. Mediation module and mediation flow: Tools (2 of 2)

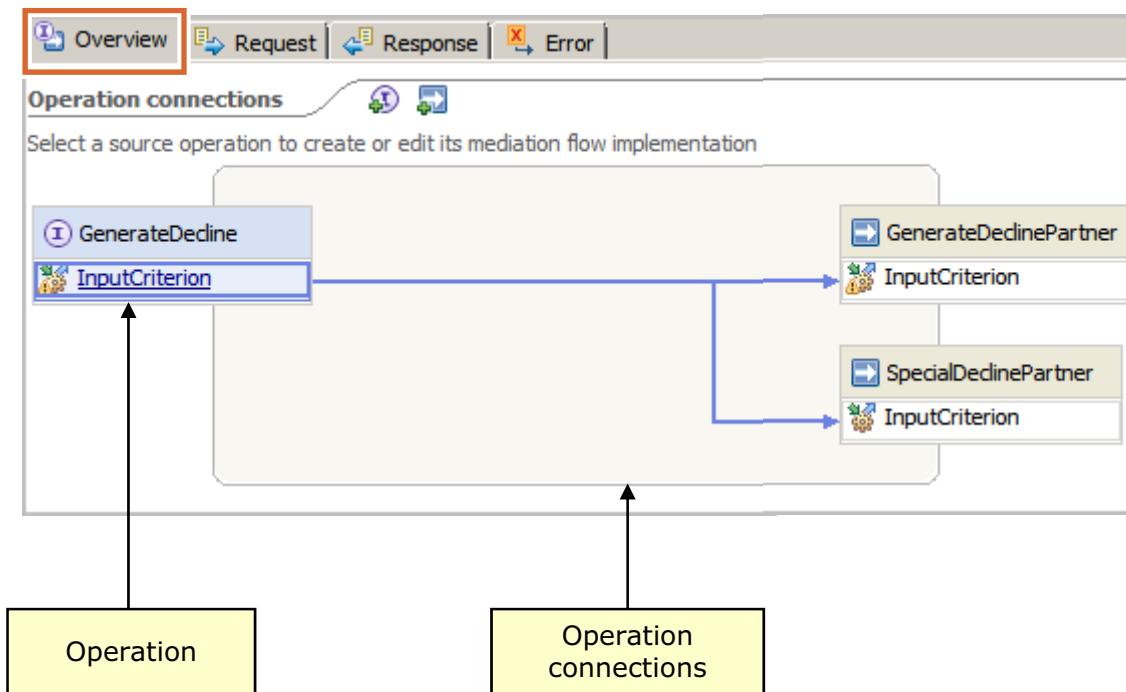
WB8601.2

Notes:

Assembly editor is used to connect a mediation flow component to calling SCA components.



Mediation flow editor: Overview tab



© Copyright IBM Corporation 2016

Figure 13-12. Mediation flow editor: Overview tab

WB8601.2

Notes:

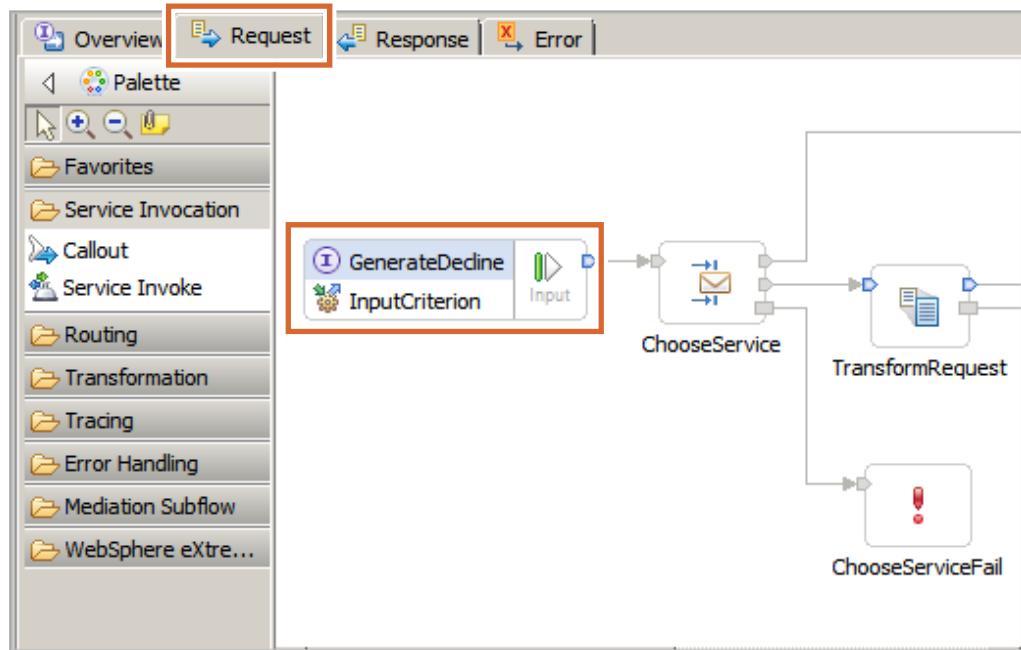
Mediation flow editor has two main sections:

- Operation connections section (the **Overview** tab): Mappings between the corresponding operations of interfaces and references are displayed.
- Request mediation flow: If the mediation is a two-way operation, the request and response flows are shown.



Mediation flow editor: Request flow view

Mediation primitives from the palette are wired together in the drawing canvas: Request view



© Copyright IBM Corporation 2016

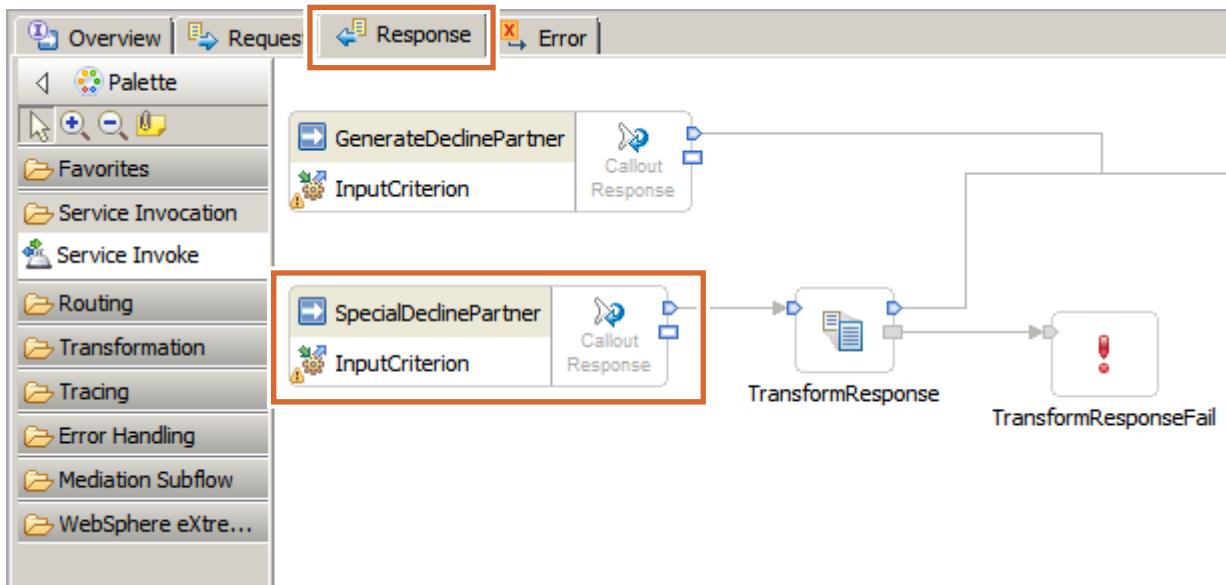
Figure 13-13. Mediation flow editor: Request flow view

WB8601.2

Notes:

Mediation flow editor: Response flow view

Mediation primitives from the palette are wired together in the drawing canvas: Response view



© Copyright IBM Corporation 2016

Figure 13-14. Mediation flow editor: Response flow view

WB8601.2

Notes:

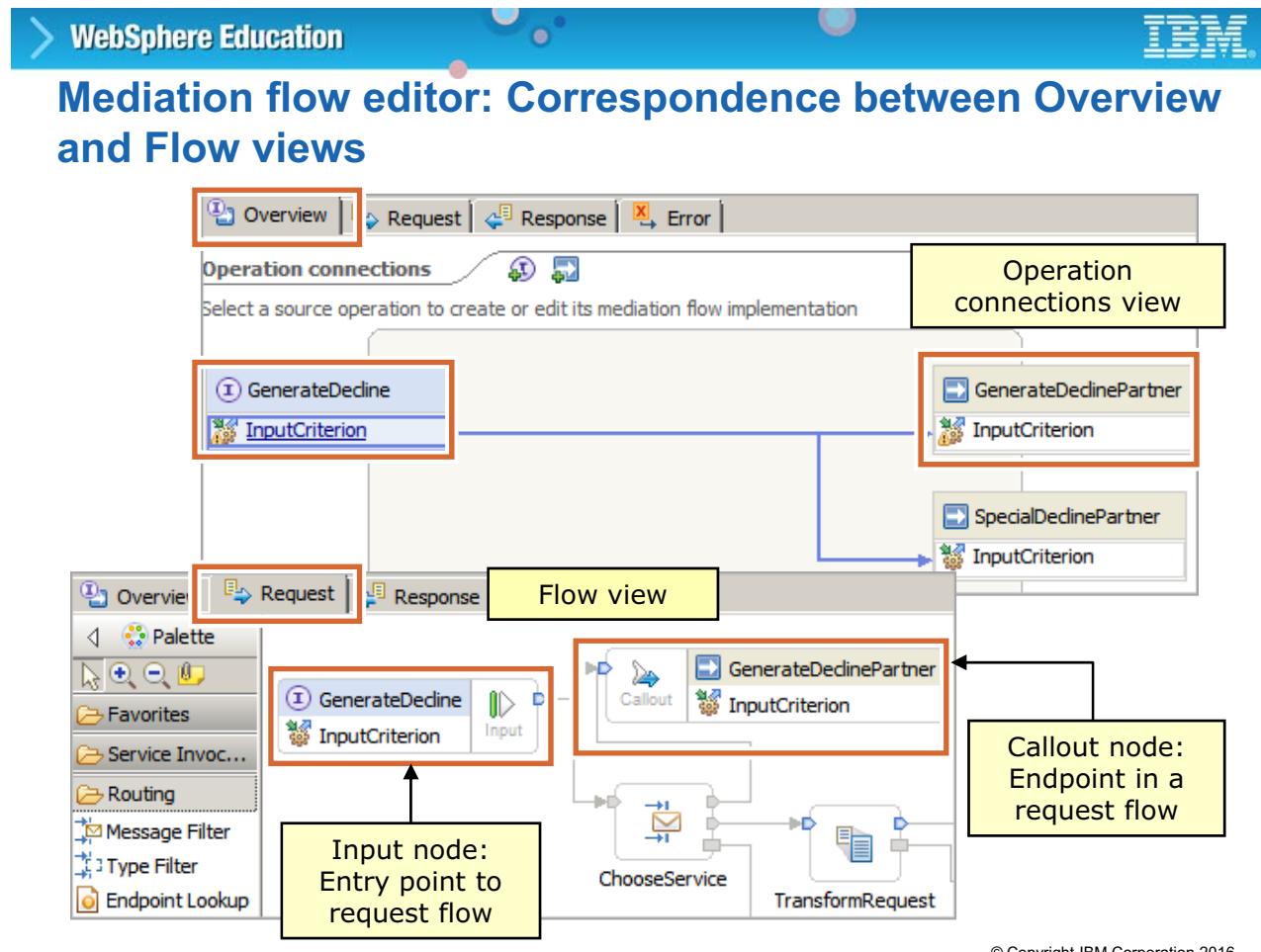


Figure 13-15. Mediation flow editor: Correspondence between Overview and Flow views

WB8601.2

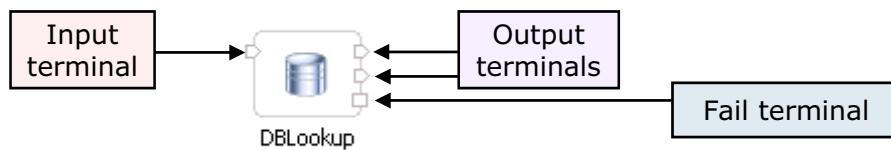
Notes:

The operation connection in the Overview perspective is mapped from the **service requester** (or **export**) to the **service provider** (or **import**). The Operation connections view shows connections to GenerateDeclinePartner and SpecialDeclinePartner.

The Flow view shows the detailed functional steps that take place across this connection after it enters the input node of GenerateDecline.

Terminals in the mediation flow editor

Terminal representation



Hovering over the terminal display name or type

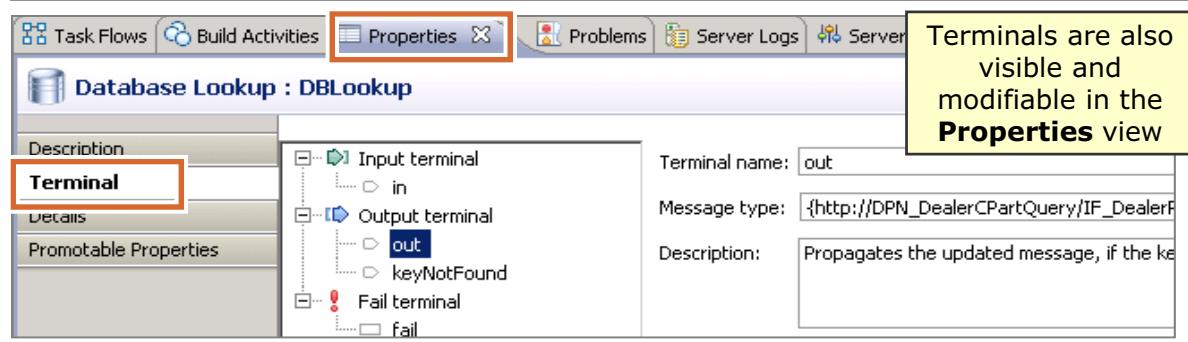
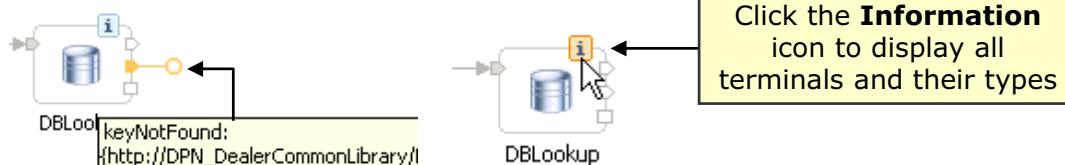


Figure 13-16. Terminals in the mediation flow editor

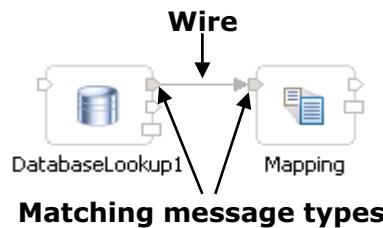
WB8601.2

Notes:

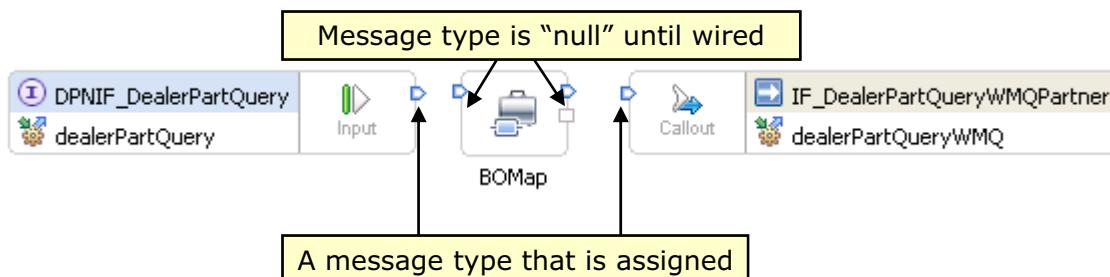
Terminals have a message type that is associated with them. These types are based on the WSDL message types.

Wiring of mediation primitive terminals

- Connections between terminals are represented as wires
- A connection must have matching terminal message types



- The editor dynamically manages the terminal message types



© Copyright IBM Corporation 2016

Figure 13-17. Wiring of mediation primitive terminals

WB8601.2

Notes:

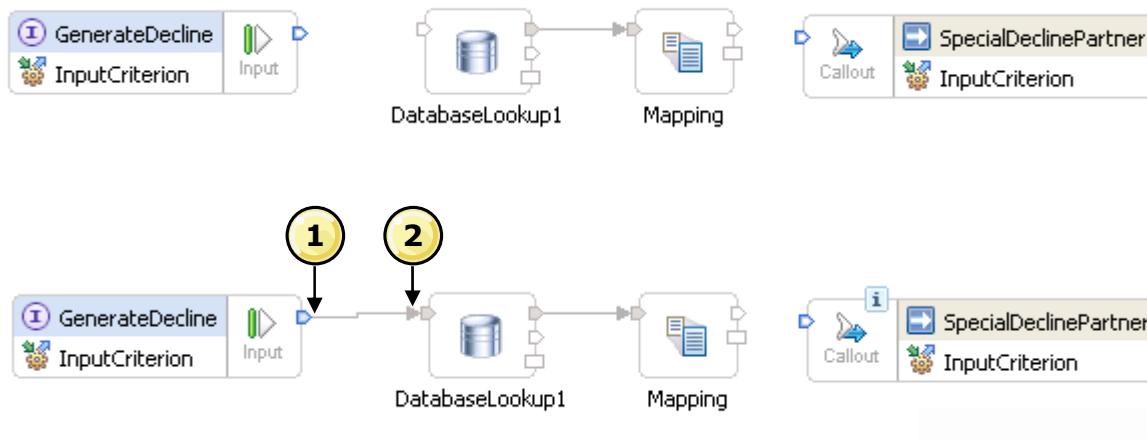
The following notes apply to the wiring of primitive terminals:

- Connections between terminals are represented as wires.
- A connection must have matching terminal message types.
- The editor dynamically manages terminal message types.
 - Input and callout nodes have terminal message types that are fixed.
 - Primitives have terminal message types that are dynamically configured.

Example of wiring of mediation primitive terminals

- Example of wiring and message type

Wiring from a “null” to a “null” type keeps the type as “null”



© Copyright IBM Corporation 2016

Figure 13-18. Example of wiring of mediation primitive terminals

WB8601.2

Notes:

To wire two primitives together, the type of the message that flows from one terminal to the other must match, or the terminal must accept a message of “anyType” (with some exceptions).

If you attempt to wire terminals that do not have the same message type, IBM Integration Designer warns you. It provides options to help you resolve the type mismatch, such as inserting a primitive to convert the message type.

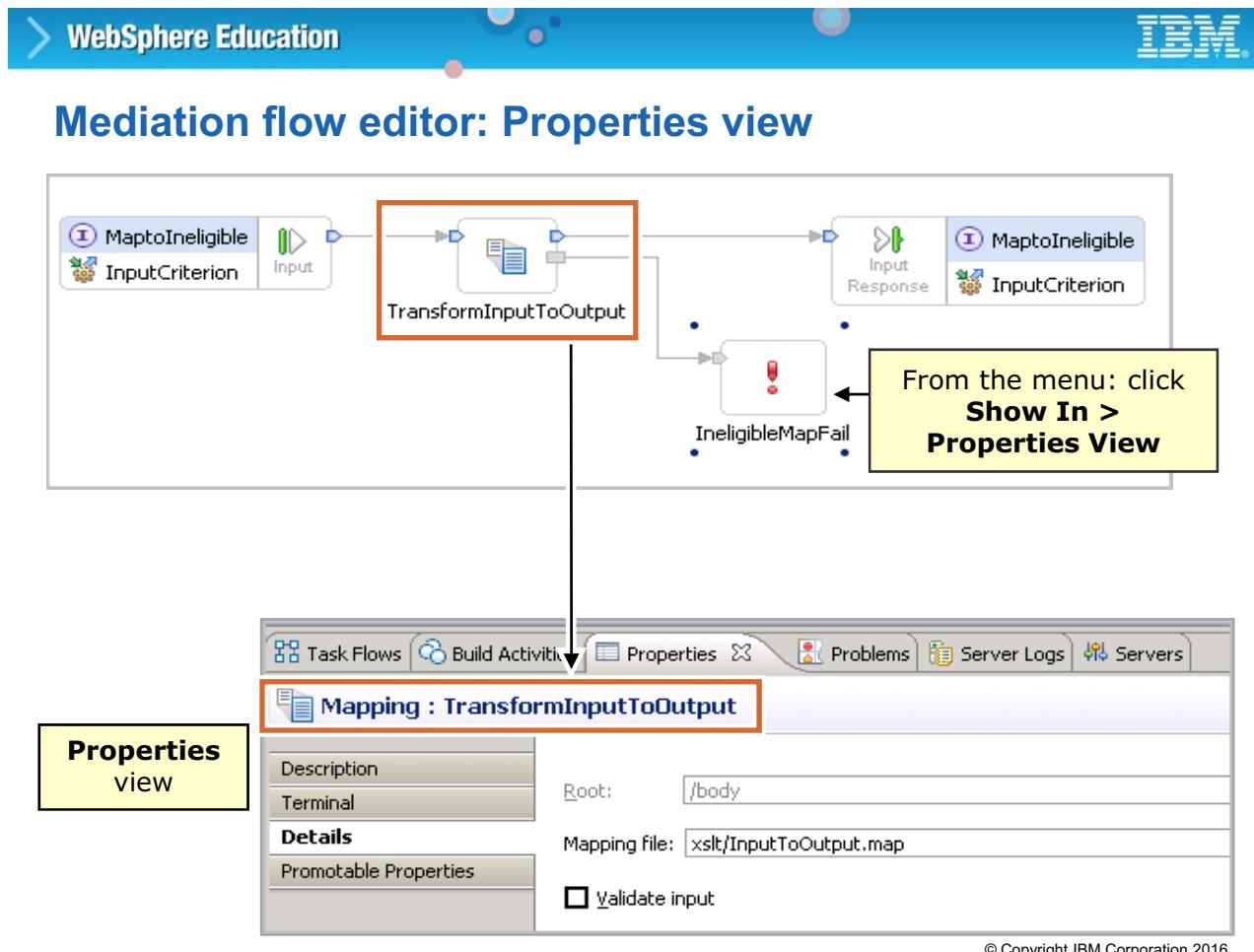


Figure 13-19. Mediation flow editor: Properties view

WB8601.2

Notes:

The properties of a primitive control how the primitive operates.

You can view the properties for any artifact in the mediation flow, such as a primitive or line. Switch to the Properties perspective and then select the artifact, or right-click the artifact and click **Show In > Properties View** from the menu. You use this view frequently when you are developing a mediation flow.

Mediations subflow: Overview

- Facilitates the reuse of mediation logic without rewriting
- Composed of Mediation primitives that are wired together
 - Start with an **in** primitive and end with an **out** primitive
 - The primitives act like input and callout nodes of a mediation flow
- Used by a parent 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 (a library increases reusability)



© Copyright IBM Corporation 2016

Figure 13-20. Mediations subflow: Overview

WB8601.2

Notes:

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 a good practice to store subflows in libraries so that they can be easily shared between mediation flows in different modules.

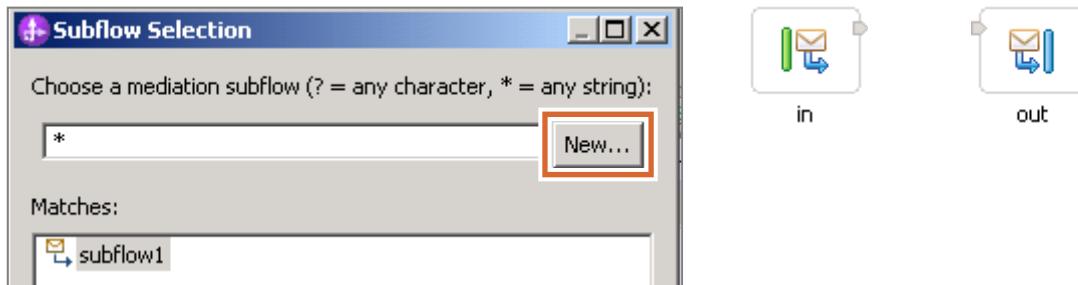
A mediation subflow has one or more **in** nodes, followed by one or more mediation primitives, followed by one or more **out** nodes. Unlike mediation flows, the inputs and outputs of a subflow are not tied to specific interfaces. Instead, the **in** and **out** nodes represent the start and end points 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. When it is defined, it is displayed as 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 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 that invokes a subflow, the promoted properties for the subflow are visible with the promoted properties of the parent flow.

Mediations subflow: Implementation

- To create a subflow, select the subflow primitive from the palette and place it on the canvas
- The Subflow Selection dialog box is displayed so you can choose an existing subflow or create one
- When you are creating a subflow, the subflow editor opens and places in and out nodes on the canvas
- You add the remaining nodes, and then configure and wire them



© Copyright IBM Corporation 2016

Figure 13-21. Mediations subflow: Implementation

WB8601.2

Notes:

The subflow editor provides the starting nodes (in and out), and then you code whatever primitives are needed to implement the required logic. When you save the subflow, it is available for reuse whenever you select the subflow primitive from the palette.



- “Promoting” the properties of a mediation primitive allows them to be changed at run time

XSL Transformation : DealerETransform					
Description	Filter	Property	<Type in the filter string>		
Terminal					
Details					
Promotable Properties					
Root	<input type="checkbox"/>				
Mapping file	<input type="checkbox"/>				
Validate input	<input type="checkbox"/>				

- Properties are changed at run time by using the administrative console; no server restart or redeployment of the mediation is required
- Not all primitive properties can be promoted

© Copyright IBM Corporation 2016

Figure 13-22. Promoted properties: Overview

WB8601.2

Notes:

By using promoted properties, you can add flexibility by being able to change how primitives function without having to change them in IBM Integration Designer.

Promoted properties: Administrative management

The screenshot shows the 'Properties' tab for the 'RouterMediationService' module. The table lists four promoted properties:

Name	Type	Value
Callout1.retryDelay1	INTEGER	0
Callout1.retryOn	STRING	0
Callout1.retryCount1	INTEGER	0
Callout1.retryCount	INTEGER	0

Administrator manages promoted properties from the Integrated Solutions Console

© Copyright IBM Corporation 2016

Figure 13-23. Promoted properties: Administrative management

WB8601.2

Notes:

The Integrated Solutions Console is also called the “administrative console” because that is where runtime administrative tasks are done.

Every promoted property has an alias name, which is the name visible to the administrator. Every promoted property belongs to a group. You can create collections of promoted properties for a mediation module. Properties that have the same alias name in the same group are administered together.

In most cases, you want to change the group name and alias name from the default values that are supplied by IBM Integration Designer. This change helps you to identify properties at run time. If you use the default alias names, you are not able to distinguish between the promoted properties of primitives of the same type, if these primitives are assigned to the same group. Promoted properties that have the same alias name and group use the same values.

Promoted properties that are changed through the administrative console affect all messages that are processed within the mediation flow. Dynamic properties allow promoted properties to be overridden on a message-by-message basis. This overriding capability requires the use of mediation policies that are stored in a registry. It also requires the use of a special mediation

primitive, the Policy Resolution primitive, which is wired in before the primitive whose properties change.

13.2. Service message objects

Service message objects



© Copyright IBM Corporation 2016

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

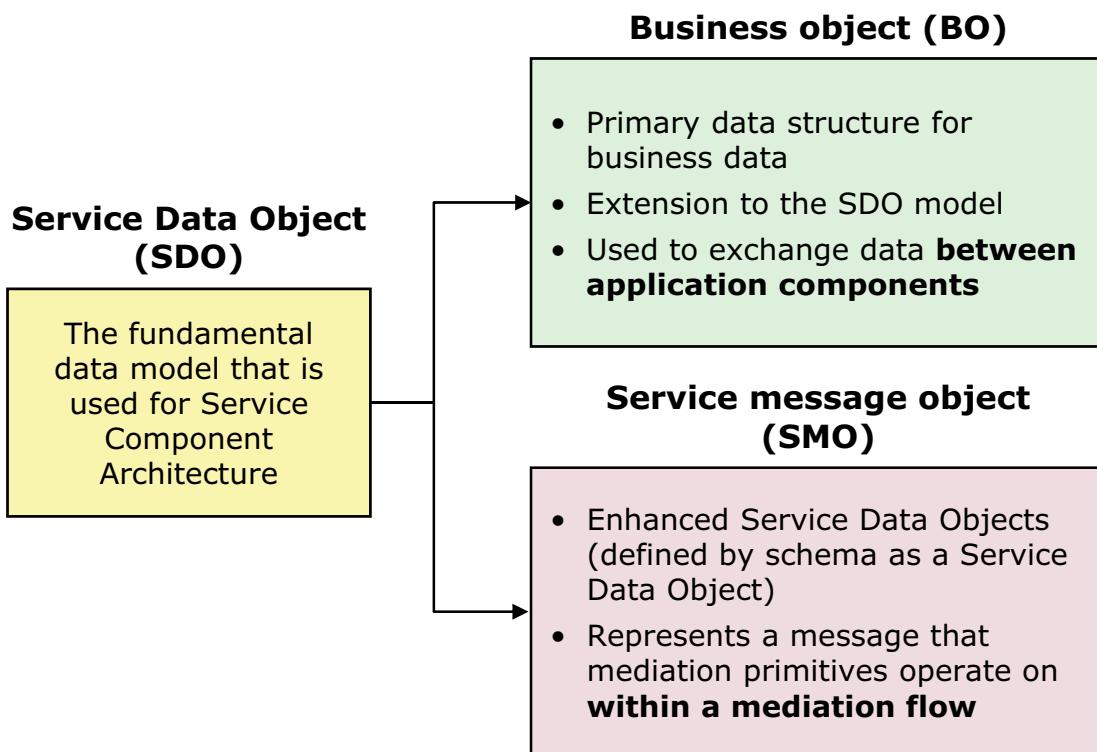
10.1

Figure 13-24. Service message objects

WB8601.2

Notes:

Overview: SDO, SMO, business objects



© Copyright IBM Corporation 2016

Figure 13-25. Overview: SDO, SMO, business objects

WB8601.2

Notes:

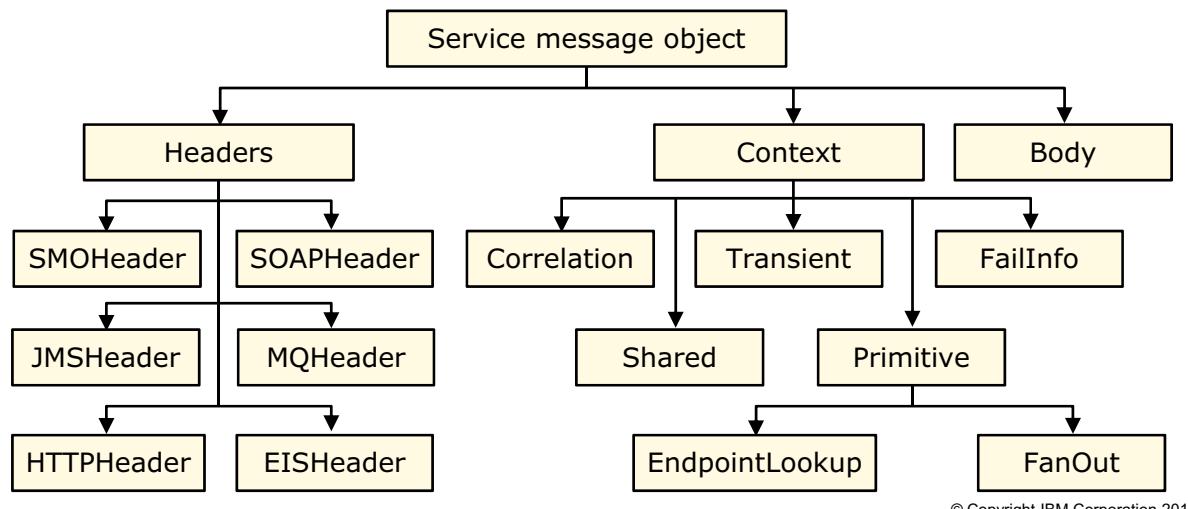
Mediations operate primarily on **service message objects**. A service message object consists of three major parts:

- **Headers** contain information relevant to the transport protocols used to bring the message onto and off the enterprise service bus.
- **Context** contains data about the message as it is propagated from one mediation primitive to the next within a mediation flow.
- **Body** is the data that is sent from the invoking application to the application that is being invoked. Body is encapsulated in a *business object*. Body is often referred to as the *payload* of the message. A service message object exists only within a mediation flow.

A **business object** is a structure; it is often a complex data type that contains the actual business data that is being transmitted through or manipulated by the mediation flow. **Service Data Object** is the standard conceptual data model that is used in Service Component Architecture. The concepts of SDO are implemented physically as business objects and service message objects. The service message object is a physical realization or implementation of Service Data Object principles.

Service message object (SMO)

- At the top level, an SMO is composed of headers, context, and body
 - Header is information about the transport protocol that is used to send a message
 - Context is other data specific to the logic of the flow as data passes between primitives (such as failure information)
 - Body (optional) is the application data (payload) of the message that contains the input or output values of the operation



© Copyright IBM Corporation 2016

Figure 13-26. Service message object (SMO)

WB8601.2

Notes:

The service message object contains the transport headers, context information that is used during the lifetime of the SMO when it flows through a mediation, and the message data itself, the body (payload). The payload is usually the business object. The body is not required; it is possible for an SMO to contain only headers and context information.

SMO structure: Headers

The headers might include:

- **SMOHeader:** Information about the message (for example, the message ID and the SMO version)
 - SMO header is **always** present
- **JMSHeader:** Contains a JMS header
 - Used when a JMS import or export binding exists
- **SOAPHeader:** Contains SOAP header information
 - Used when a web service import or export binding exists
- **SOAPFaultInfo:** Contains information about SOAP faults
- **properties[]:** Arbitrary list of name-value pairs (for example, JMS user properties)
- **MQHeader:** Contains md (MQMD), control (format information of the message body), and other headers (like RFH and RFH2)
- **HTTPHeader:** Contains HTTP headers when an HTTP import or export binding exists
- **EISHeader:** Present when the email, flat file, or FTP adapters are used
 - Different headers exist for each adapter
- **WSAHeader:** Used when an export is configured with JAX-WS binding to provide WS-Addressing support

headers
SMOHeader
JMSHeader
SOAPHeader
SOAPFaultInfo
properties
MQHeader
HTTPHeader
EISHeader
WSAHeader

© Copyright IBM Corporation 2016

Figure 13-27. SMO structure: Headers

WB8601.2

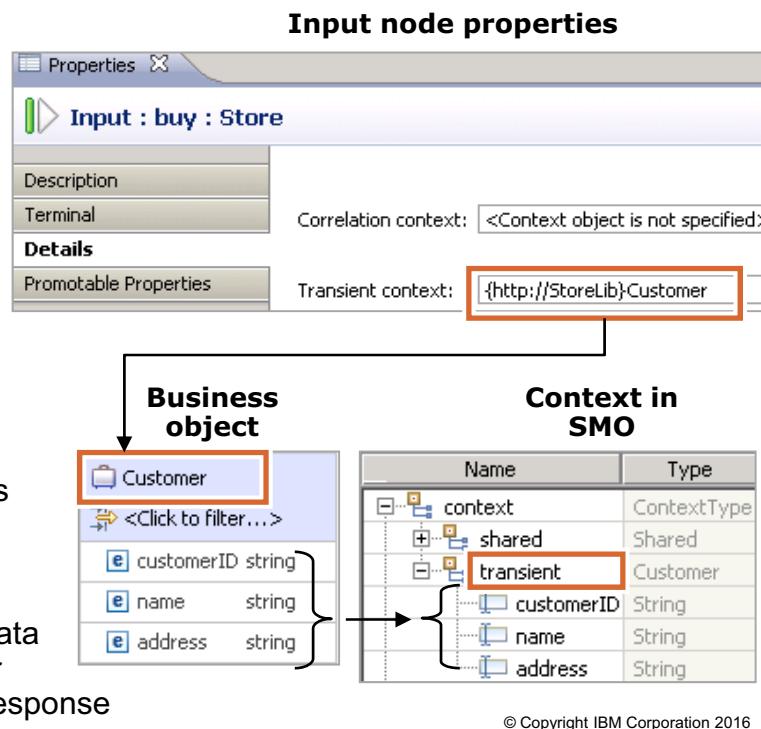
Notes:

The SMO headers contain information that originates from a specific import or export binding (a binding specifies the message format and transport protocol details). Messages can come from a number of sources, so the SMO can carry different kinds of message headers.

SMO structure: Correlation and transient contexts

The context includes the **correlation** and **transient** contexts

- Both of them are:
 - Used to pass application data between mediation primitives
 - XSD-defined data objects
 - Specified on the input node properties of the mediation flow
- Correlation** context maintains data across a request/response flow
- Transient** context:
 - Exists only while the flow is running
 - Maintains data only during one direction (request or response), but the same data object definition is used for both the request and the response



© Copyright IBM Corporation 2016

Figure 13-28. SMO structure: Correlation and transient contexts

WB8601.2

Notes:

Using the service message object **context**, mediation primitives can pass data that is not part of the message payload, between themselves. Several context types are available.

SMO structure: FailInfo context

- The **context** also includes the `failInfo` type
 - Contains failure information
 - Added to an SMO when it flows across the fail terminal of a primitive
- The information that is provided includes:

<code>failureString</code>	Describes the failure
<code>origin</code>	Mediation primitive in which failure occurred
<code>invocationPath</code>	The flow that is taken through the mediation
<code>predecessor</code>	Previous failure

⊖ [e] failInfo	[0..1]	FailInfoType
⊕ lang	[0..1]	<Anonymous>
⊖ [e] failureString	[1..1]	string
⊖ [e] origin	[1..1]	string
⊖ [e] invocationPath	[1..1]	<Anonymous>
⊖ [e] primitive	[1..*]	PrimitiveType
⊕ lang	[1..1]	string
⊕ name	[1..1]	string
⊕ lang	[0..1]	string
⊖ [e] predecessor	[0..1]	FailInfoType
⊕ lang	[0..1]	<Anonymous>
⊖ [e] failureString	[1..1]	string
⊖ [e] origin	[1..1]	string
⊕ [e] invocationPath	[1..1]	<Anonymous>
⊕ [e] predecessor	[0..1]	FailInfoType

© Copyright IBM Corporation 2016

Figure 13-29. SMO structure: FailInfo context

WB8601.2

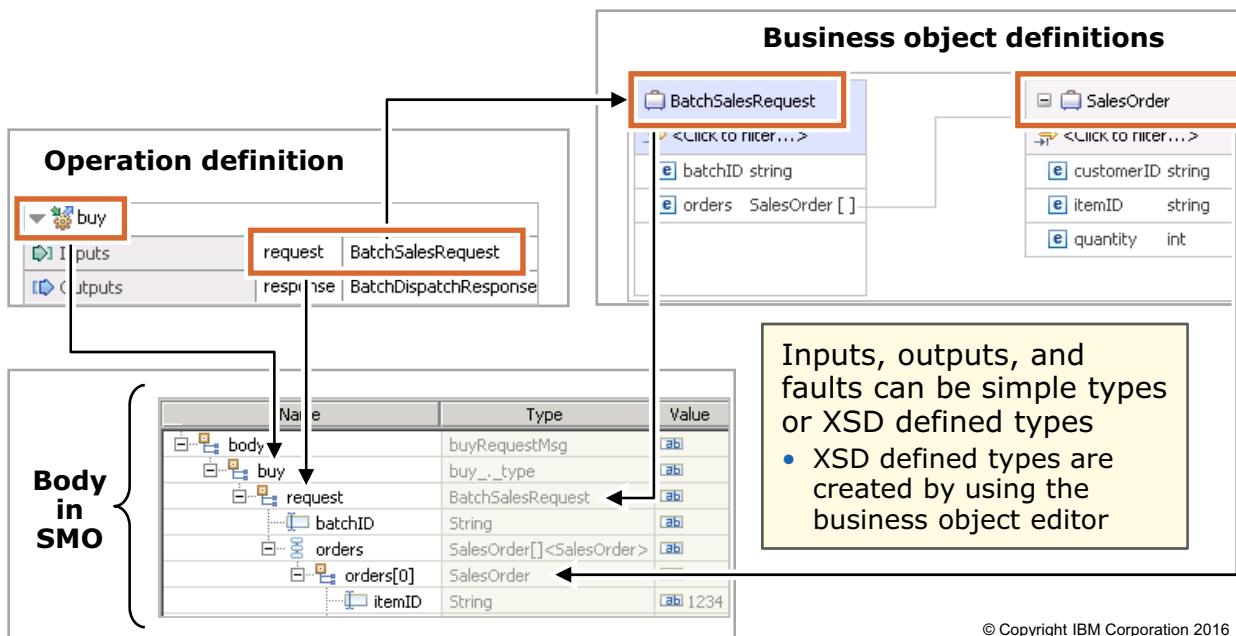
Notes:

SMO structure: Body

The body contains the payload of the message

- Payload is the application data that flows in the message
- It identifies the operation and either its inputs, outputs, or faults

Operation is defined in WSDL by using the interface editor



© Copyright IBM Corporation 2016

Figure 13-30. SMO structure: Body

WB8601.2

Notes:

The body of the SMO message contains the actual payload of the message. The body of the message identifies the operation on the data, along with its input message type, output message type, or fault message type, depending on which operation is specified.

SMO in a mediation flow

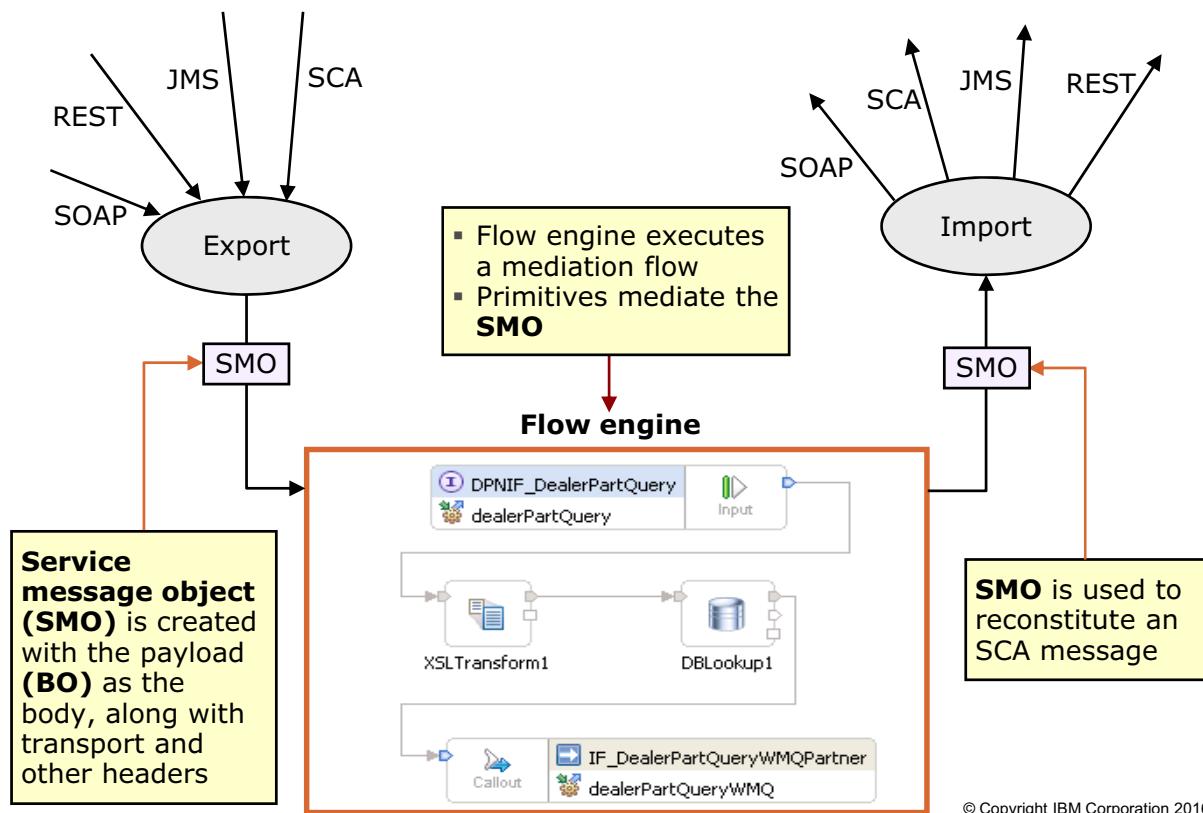


Figure 13-31. SMO in a mediation flow

WB8601.2

Notes:

This visual shows the various forms that data takes through a mediation flow. In this case, the message enters as a JMS message and leaves as a SOAP message.

- The export receives the incoming message, along with the information about the transport protocol by which it is received. The artifacts of the export, including the bindings, interface, and business object, are used to construct the service message object (SMO). Business objects represent the data that flows between each service. Whether the interface that is associated with a particular component is defined as a Java interface or a WSDL port type, the input and output parameters are represented by using business objects.
- When the message is passed into the mediation flow, it is converted to a service message object. The service message object is, again, composed of the transport (and other) headers, plus the actual data ("payload") that is sent from the invoking application. That data is stored in the service message object as a business object.
- As the message (as an SMO) is propagated through the mediation flow, the various mediation primitives access and modify the elements of the SMO. Some primitives examine the header structure, while others might modify the payload in the business object. Still others update the

transport headers in preparation for the message that is being sent to another service that uses a different transport protocol from the one on which the message was received.

- When the message reaches the end of the mediation flow (for example, when the mediation invokes another service), the artifacts of the import are used to reconstitute the SMO into the appropriate native message type. The data in the outgoing message is derived from the payload of the SMO. The transport protocol uses the headers in the SMO to know how to deliver the message to its next destination.

Manipulating SMOs (1 of 2)

- Three ways to access and manipulate SMOs: XPath, XSL, and Java
- XPath expressions
 - Primary mechanism for accessing the SMO
 - Used in some form by all of the mediation primitives
 - Identify elements to read or update the conditional expressions
- XSL stylesheets
 - Used by the Mapping mediation primitive
 - Normally used to modify an SMO type within a flow
 - Can also be used to manipulate the SMO content without changing the type
- Java code
 - Used by the Custom Mediation primitive
 - SMO is accessed through generic Service Data Object APIs (loosely typed), or the SMO APIs (“strongly typed”)
 - Can access and update content
 - Can also modify the SMO type

© Copyright IBM Corporation 2016

Figure 13-32. Manipulating SMOs (1 of 2)

WB8601.2

Notes:

The ability to access or modify the contents of the SMO is a fundamental notion in mediation programming. SMOs can be manipulated in several ways in IBM Process Server.

Manipulating SMOs (2 of 2)

- A set of Java services is available to allow business objects to be created and manipulated
- A **shallow** copy uses object references and does not copy primitive data types
 - In a **deep** copy, the objects themselves get copied
- These services are part of the `com.ibm.websphere.bo` package
- Classes include:
 - **BOFactory**: Creates instances of business objects
 - **BOXMLSerializer**: Converts to a business object from a stream, or writes the content of a business object in XML format to a stream
 - **BOCopy**: Methods that make copies of business objects (“deep” and “shallow” semantics)
 - **BODataObject**: Allows access to the data object aspects of a business object, such as the change summary, the business graph, and the event summary
 - **BOXMLDocument**: Allows the manipulation of the business object as an XML document
 - **BOChangeSummary** and **BOEventSummary**: Access and manipulate the change summary and event summary portion of a business object
 - **BOEquality**: Determines whether two business objects contain the same information, both “deep” and “shallow” equality

© Copyright IBM Corporation 2016

Figure 13-33. Manipulating SMOs (2 of 2)

WB8601.2

Notes:

Classes include:

- **BOFactory**: Creates instances of business objects
- **BOXMLSerializer**: Converts to a business object from a stream, or writes the content of a business object in XML format to a stream
- **BOCopy**: Methods that make copies of business objects (“deep” and “shallow” semantics)
- **BODataObject**: Allows access to the data object aspects of a business object, such as the change summary, the business graph, and the event summary
- **BOXMLDocument**: Allows the manipulation of the business object as an XML document
- **BOChangeSummary** and **BOEventSummary**: Access and manipulate the change summary and event summary portion of a business object
- **BOEquality**: You can use BOEquality to determine whether two business objects contain the same information, both “deep” and “shallow” equality
- **BOType** and **BOTypeMetaData**: Materialize instances of the `commonj.sdo` type; you can manipulate the associated metadata

- **BOInstanceValidator:** Validates the data in a business object to determine whether it conforms to the XSD



Unit summary

- Describe the role of mediation services in IBM Process Server
- Define the concept of mediation modules
- Describe how to create mediation flows in IBM Integration Designer
- Describe the role of SMOs in mediations
- Explain the structure of SMOs

© Copyright IBM Corporation 2016

Figure 13-34. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. True or False: A mediation primitive contains the logic to transform the content of a message.
2. True or False: The operation connections in the Overview perspective of the mediation flow editor display the mappings between the interfaces and operations of a mediation.
3. True or False: Mediation primitive terminals are associated with the message type that the primitive is processing, in most cases.
4. True or False: Input and output messages to the mediation primitives are represented as service message objects.
5. True or False: XPath expressions can be used to manipulate an SMO.

© Copyright IBM Corporation 2016

Figure 13-35. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.



Checkpoint answers

1. True.
2. False: They show the mapping between interfaces and references.
3. True.
4. True.
5. True.

© Copyright IBM Corporation 2016

Figure 13-36. Checkpoint answers

WB8601.2

Notes:

Exercise 11



Creating mediation services, part I

© Copyright IBM Corporation 2016

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

10.1

Figure 13-37. Exercise 11

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create a mediation module that contains a Mapping primitive
- Define an XML data map
- Test a mediation module that contains a Mapping primitive

© Copyright IBM Corporation 2016

Figure 13-38. Exercise objectives

WB8601.2

Notes:

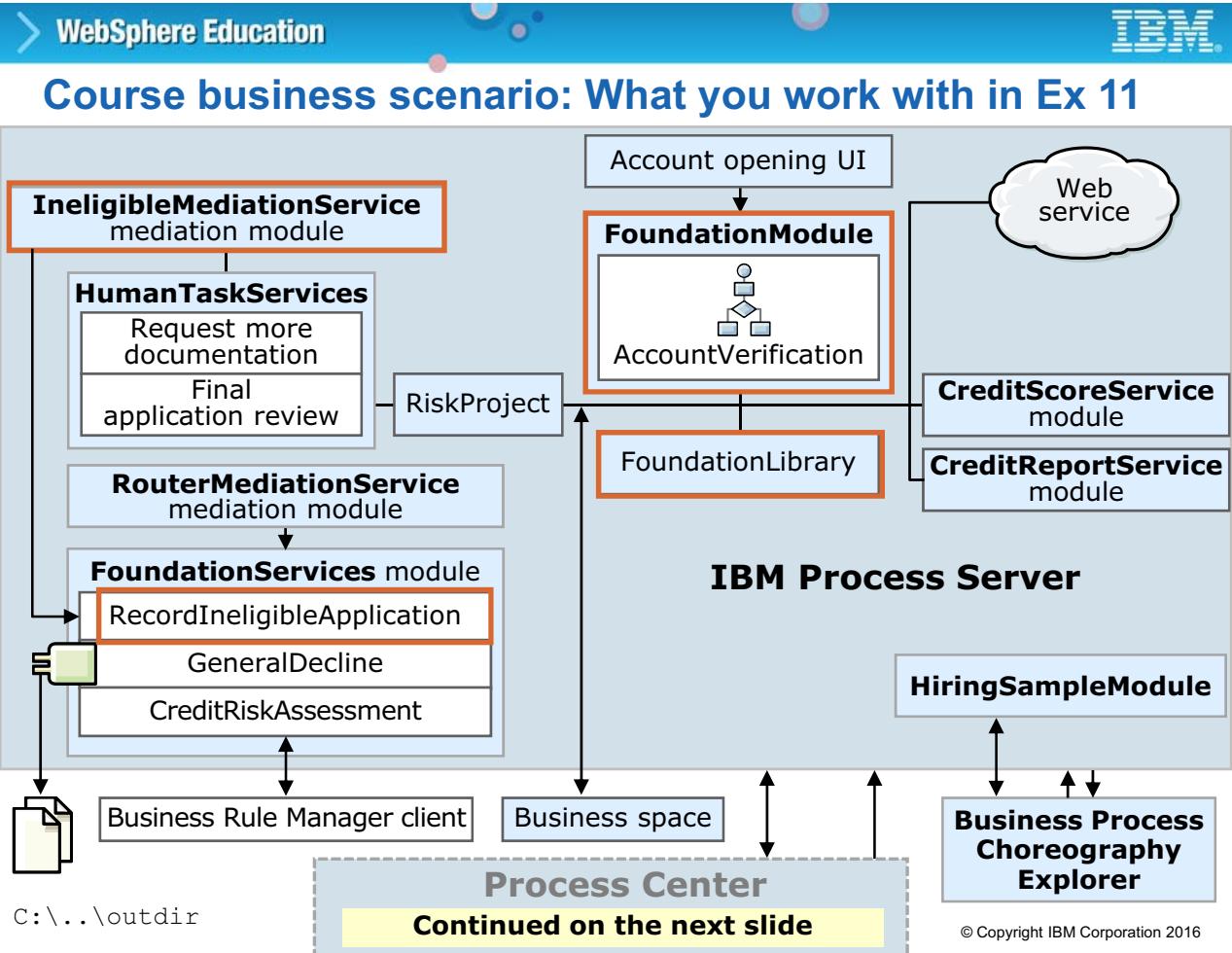


Figure 13-39. Course business scenario: What you work with in Exercise 11

WB8601.2

Notes:

WebSphere Education 

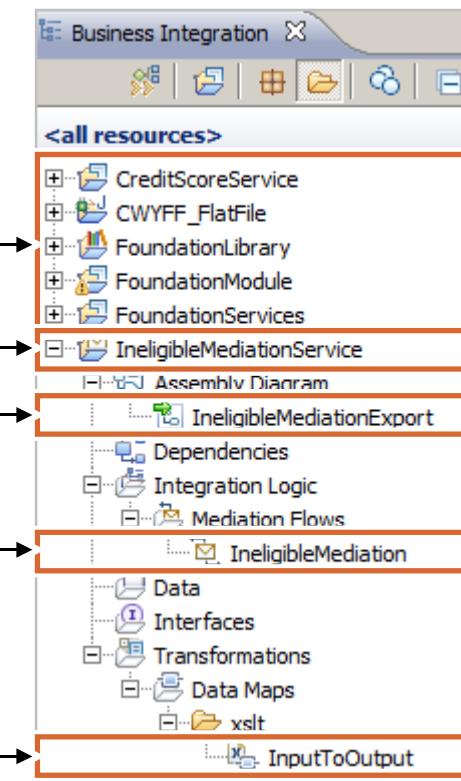
Components that are required for Exercise 11

Prebuilt components that are imported in this lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **FoundationServices**
5. **CWYFF_FlatFile**

New components that you create in this lab:

1. **IneligibleMediationService** module
2. **IneligibleMediationExport** export component
3. **IneligibleMediation** mediation flow
4. **InputToOutput** map



© Copyright IBM Corporation 2016

Figure 13-40. Components that are required for Exercise 11

WB8601.2

Notes:

In this exercise, you implement a mediation module that is named **IneligibleMediationService**, which transforms messages between the **AccountVerification** process and the **RecordIneligibleApplication** service (which archives ineligible applications). Suppose the data input for the **RecordIneligibleApplication** service is different from the **CustomerApplication** input that the **AccountVerification** process uses. Under this scenario, the **RecordIneligibleApplication** component does not read the data and no output message is sent. To fix this problem, you must transform the messages that are passed between the services. This problem is a common challenge that your application must be prepared to resolve, and IBM Integration Designer solves it by creating a transformation between the two messages.

You implement a Mapping primitive inside a mediation flow to transform the messages by using a data map. After implementing the transformation in the mediation flow, you assemble the application by wiring the mediation service to the **AccountVerification** process.

Finally, after creating the flat file adapter component, you assemble the SCA components in IBM Integration Designer and then test the application.

Assemble SCA components in Exercise 11

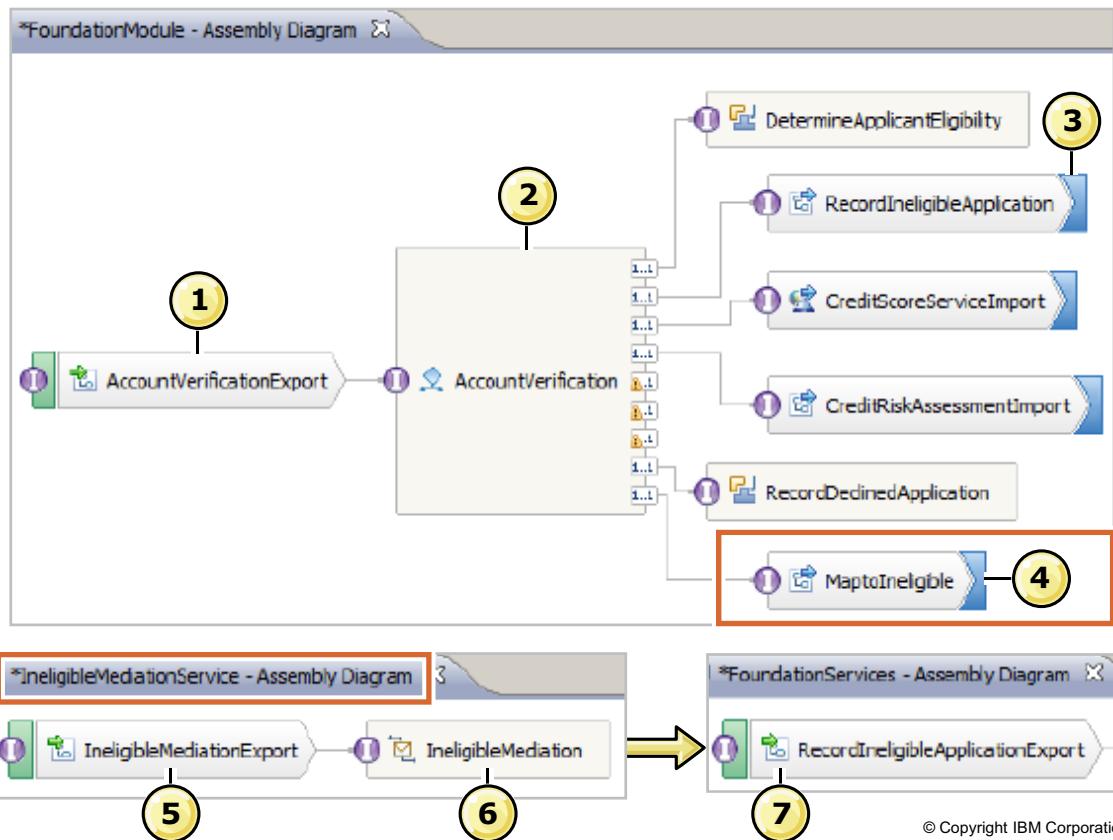


Figure 13-41. Assemble SCA components in Exercise 11

WB8601.2

Notes:

The following steps are illustrated in the diagram:

1. The AccountVerificationExport component exposes the AccountVerification business process.
2. When the application is ineligible, the AccountVerification process needs to record the ineligible application in the database and terminate the process. It calls the RecordIneligibleApplication import component.
3. The RecordIneligibleApplication import component is used to call the application or service outside FoundationModule. In this scenario, it calls the RecordIneligibleExport component in the FoundationServices module.
4. The MaptoIneligible import invokes the IneligibleMediationExport component in the IneligibleMediationService mediation module.
5. The IneligibleMediationExport exposes the mediation flow by calling the IneligibleMediation component.
6. The IneligibleMediation component transforms a CustomerApplication into an IneligibleApplication before invoking the RecordIneligibleApplicationExport component to archive the application.

7. The RecordIneligibleExport component calls the RecordIneligibleApplication Java component. The RecordIneligibleApplication component is a Java component that creates the output message.

"Account verification recorded this application as ineligible for the customer <company name>" is going to be recorded to the system. In this scenario, it calls the FlatFileOutboundImport component. The FlatFileOutboundImport component writes the output message to the file system. In this scenario, it writes to a text file in the C:\IneligibleAppArchive\outdir output directory.

Unit 14. Mediation primitives

What this unit is about

This unit explores the various mediation primitives that are used in mediation flows.

What you should be able to do

After completing this unit, you should be able to:

- Describe the role of mediation primitives in mediation flows
- Describe the prebuilt mediation primitives that are available for mediation flows

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the role of mediation primitives in mediation flows
- Describe the prebuilt mediation primitives that are available for mediation flows

© Copyright IBM Corporation 2016

Figure 14-1. Unit objectives

WB8601.2

Notes:



Topics

- Message transformation and enrichment primitives
- Flow control primitives
- Faults, tracing, and error handling primitives

© Copyright IBM Corporation 2016

Figure 14-2. Topics

WB8601.2

Notes:

14.1.Message transformation and enrichment primitives

Message transformation and enrichment primitives



© Copyright IBM Corporation 2016

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

10.1

Figure 14-3. Message transformation and enrichment primitives

WB8601.2

Notes:

What is a mediation primitive?

- An encapsulated unit of logic that manipulates the message as it passes through the enterprise service bus
 - Mediation primitives accept and process messages to and change the format, the content, or the target service provider
 - Contained within a mediation flow
- IBM Process Server provides variety of predefined mediation primitives
 - You implement the mediation logic
 - In most cases, without the requirement to write code
- Custom primitives can be constructed
 - They are also available from third-party sources

© Copyright IBM Corporation 2016

Figure 14-4. What is a mediation primitive?

WB8601.2

Notes:

Mediation primitives are the core building blocks that are used to process the request and response messages in a flow. They are used to update, add to, and transform the SMO to control the flow for iteration or to make routing decisions, and do logging and event generation.

Built-in primitives do some predefined functions that are configurable by using properties. Custom Mediation primitives allow implementation of functions in Java.

Each mediation primitive presents a structure similar to this example. It has an input terminal, along with a set of output terminals (zero or more) and a fail terminal. Each terminal has a “type” defined with a WSDL message, and that type determines what you can “wire” the terminal to. Terminals must have compatible message types, or they cannot be connected to one another. The input and output terminals are the typical connection points for wiring mediation primitives into mediation flows.

The fail terminal also has a type; it has the same type as the input terminal. If an exception occurs during the execution of the primitive, the fail terminal is invoked, for example, if a database access exception occurs. In this case, the fail terminal propagates the original message, together with information about any exception that occurred, to the primitive to which it is connected. If no

primitive is connected to the fail terminal, the flow fails (the processing of the message ends) with one of the following exceptions:

- **MediationConfigurationException:** The primitive can tell that a configuration problem or a transient problem exists, such as a resource availability issue.
- **MediationBusinessException:** A business error (for example, a key that is expected to be in a message is not found).
- **MediationRuntimeException:** Problems with setting up or establishing the flow in the runtime.

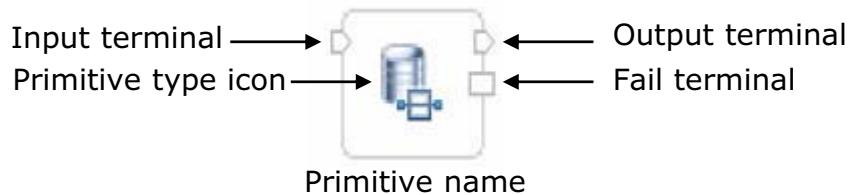
Failed flows also cause the current transaction to roll back.

Mediation primitives operate on service message objects. XPath expressions can be used to access SMOs. Many configuration properties are XPath expressions. The XPath expression builder is used to set these properties. Expressions that are built with the expression builder are called custom XPath expressions. Several primitives have a configuration property that is called “root,” which represents the portion of the service message object that is being used.

Values for root include:

- `/`: The entire service message object
- `/body`: The body of the message (operation and parameter values)
- `/context`: The message context (transient context, correlation context, and `failInfo`)
- `/headers`: Protocol headers

Common features of mediation primitives



- Most mediation primitives share common attributes:
 - A default name, which is assigned when the primitive is moved to the drawing canvas
 - An input terminal, where the message is received
 - An output terminal, where the message is sent after the primitive successfully completes its actions
 - A fail terminal, where the original message and error information are sent when an exception occurs while the primitive is processing
- Some primitives have fewer or more terminals
- Some primitives have configurable properties that are associated

© Copyright IBM Corporation 2016

Figure 14-5. Common features of mediation primitives

WB8601.2

Notes:

Mediation primitives: Transformation (1 of 2)

Mediation primitive	Description
Business Object Map*	<ul style="list-style-type: none"> ▪ Allows graphical creation of message transformations by reusable business object maps ▪ Use it to change the message content or the message type
Custom Mediation	Use Java code to implement custom mediation logic
Data Handler*	Converts an element of a message between a physical format and a logical structure
Database Lookup*	Modifies the content of a message by reading data from a user-supplied database

*See appendixes for details

© Copyright IBM Corporation 2016

Figure 14-6. Mediation primitives: Transformation (1 of 2)

WB8601.2

Notes:

IBM Integration Designer provides a number of built-in mediation primitives that you can use to construct mediation flows. For the purposes of this course, the primitives are broken into three major groups: transformation, flow control, and tracing or error handling. Some primitives are included in the appendixes as reference material. Time constraints prevent coverage of all the primitives in this unit. For more information about mediation primitives, see course WB858 or a later version of the second week of developer training for IBM Business Process Manager Advanced.

Mediation primitives: Transformation (2 of 2)

Mediation primitive	Description
Message Element Setter	Modifies the content (but not the type) of a message by adding, deleting, or changing message elements
Message header setters (four types)	<ul style="list-style-type: none"> ▪ Creates, modifies, copies, or deletes message transport headers ▪ Specific header setter primitives exist for HTTP, JMS, WebSphere MQ, and SOAP messages
Set Message Type*	Overlays message fields with more detailed structures to allow easier manipulation of message content; treat weakly typed fields as strongly typed
Mapping	Allows modification of the message content by using Extensible Stylesheet Language (XSL) transformations or business object map transformations

*See appendixes for details

© Copyright IBM Corporation 2016

Figure 14-7. Mediation primitives: Transformation (2 of 2)

WB8601.2

Notes:

Mapping mediation primitive

- Allows manipulation of messages by using an XSLT transformation or business object maps
- Provides override support for mapping complex structures
- XSLT V2.0 and XPath V2.0 support for an XSL transformation or a business object map transformation
- Can use a graphical editor to change the headers, context, or body of the SMO by mapping between the input and output message
- The XSL transformations operate on an XML serialization of the message, whereas the business object map transformation operates on the Service Data Objects (SDO)

© Copyright IBM Corporation 2016

Figure 14-8. Mapping mediation primitive

WB8601.2

Notes:

You can use the Mapping mediation primitive to transform messages by using XSL transformations or business object maps. When you are integrating services, you often must transform data into a format that the receiving service can process. You can use the Mapping mediation primitive to transform one message type into a different message type.

You can use the Mapping mediation primitive to do the following tasks:

- Transform an input message type to a different output message type; for example, the mediation flow starts with one operation but ends with another operation, and the second operation has a different argument type
- Alter the content of a message without changing the message type
- Apply an existing XML map, or stylesheet, to transform a message

The Mapping mediation primitive can be useful if you want to:

- Manipulate data before or after the “Database Lookup” mediation primitive is invoked
- Copy the response from the Service Invoke mediation primitive into the shared context
- Use data in the shared context to create a message body after the Fan In mediation primitive

You can use either the Mapping mediation primitive or the Business Object Map mediation primitive to transform messages. The key difference is that the Business Object Map mediation primitive uses Service Data Objects (SDO) to do transformations on business objects. The Mapping mediation primitive can do transformations in XML, by using a stylesheet, business objects, and Service Data Objects.

If you have existing XML maps, XSL stylesheets, or business object maps, you might be able to reuse them with the Mapping mediation primitive. Additionally, if you have existing business object maps, you might be able to reuse them with the Business Object Map mediation primitive. Some kinds of transformation are easier to do in XSL, and others by using a business object map.

Message Element Setter primitive

Modifies the body of an SMO by specifying:

- The element that must be modified (specified as an XPath expression)
- The operation that you want to perform on the target:
 - **Set**: Assigns a constant value to the target element
 - **Copy**: Copies a source element to a target element
 - **Append**: Copies from a source element to a new element instance, by appending to a repeating element in the output
 - **Delete**: Deletes an element instance
- The value that is going to be used, if a **copy** or **append** operation is selected (ensure that types are compatible, or runtime exception occurs)
- **Validate input** property: If set to “true,” and the input message is invalid (does not match its schema), a runtime exception occurs



Message Element Setter

Action:	<input type="text" value="Set"/>	<input type="button" value="▼"/>
Target:	<input type="text"/>	<input type="button" value="Browse..."/>
Type:	<input type="text"/>	<input type="button" value="Browse..."/>
Value:	<input type="text"/>	<input type="button" value="Browse..."/>

You cannot use the Message Element Setter to change the **type** of a message, only the **body** of the message

© Copyright IBM Corporation 2016

Figure 14-9. Message Element Setter primitive

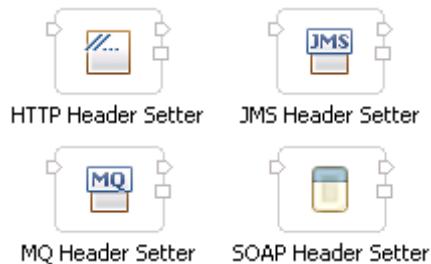
WB8601.2

Notes:

The Message Element Setter is used to modify the body of the SMO at the individual element level.

Message header setter primitives

- Four unique message header setter primitives:
 - HTTP Header Setter
 - JMS Header Setter
 - WebSphere MQ Header Setter
 - SOAP Header Setter
- Modifies the transport protocol header of an SMO message by specifying:
 - Type (“mode”) of operation you want to do (create, copy, modify, or delete)
 - The name of the header to be modified
 - The value that is used if a create, copy, or modify operation is requested
 - A flag that indicates whether the value is an XPath expression or a literal
- Validate input** property: If set to “true,” and the input message is invalid (does not match its schema), a runtime exception occurs



Mode:	Create
Header Name::*	<Select the name of a standard Header or enter your own>
Type:	String
<input type="checkbox"/> Set Value using XPath	
Value:	

You cannot change the **body** of a message that uses the message header setter primitives, only the specified transport protocol headers

© Copyright IBM Corporation 2016

Figure 14-10. Message header setter primitives

WB8601.2

Notes:

Four message header setter primitives are available, one for each of the transport protocols that are most often used in the IBM Process Server environment: JMS, HTTP, WebSphere MQ, and SOAP. This group of primitives is used to modify the transport protocol header, for example, changing the message destination in the JMS or WebSphere MQ headers.

WebSphere Education



Custom Mediation primitive

- Allows implementation of custom mediation logic in mediation flows by using Java snippets or visual snippets
- Useful when no built-in primitive provides the needed function
- Can define more input and output terminals and properties on the primitive at development time for increased flexibility
- Must explicitly send the message through the output terminal by using the `out.<terminal name>` notation
- Must explicitly define the message types on the terminals
- When a Custom Mediation primitive is created on the canvas, the initial “skeleton” code is created for you



Custom Mediation

Custom Mediation : Custom Mediation

Description	Implementation: <input type="radio"/> Visual <input checked="" type="radio"/> Java
Terminal	
User Properties	
Details	
Java Imports	<pre>/*** * GENERATED COMMENT - DO NOT MODIFY * Variables: for output terminals - ou * for user properties - <No * Inputs: inputTerminal (com.ibm.ws * Exceptions: com.ibm.wsspi.sibx.mediat */ out.fire(smo); // propagate the service</pre>
Promotable Properties	

© Copyright IBM Corporation 2016

Figure 14-11. Custom Mediation primitive

WB8601.2

Notes:

The Custom Mediation primitive allows writing your own Java code to manipulate the SMO.

14.2.Flow control primitives

Flow control primitives



© Copyright IBM Corporation 2016

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

10.1

Figure 14-12. Flow control primitives

WB8601.2

Notes:

Mediation primitives: Routing (1 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 based on the message content
Service Invoke	Calls a service from within a mediation flow (instead of waiting until the end of the mediation flow and then implementing the callout mechanism)
Type Filter	Routes messages within a mediation flow based on the message type

© Copyright IBM Corporation 2016

Figure 14-13. Mediation primitives: Routing (1 of 2)

WB8601.2

Notes:

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 invoked. As in the previous topic, some primitives are included in the appendixes as reference material. Time constraints prevent coverage of all the primitives in this unit.

Mediation primitives: Routing (2 of 2)

- These primitives control routing within the mediation flow

Mediation primitive	Description
Fan In*	Aggregates multiple messages (created by a Fan Out primitive) based on a decision point, and 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 the branches of a flow are run

*See appendixes for details

© Copyright IBM Corporation 2016

Figure 14-14. Mediation primitives: Routing (2 of 2)

WB8601.2

Notes:

Message Filter primitive

- Routes a message within a mediation flow based on the message content
- Acts like an “if . . goto” statement to send an incoming message through one or more paths if the filter criteria are met
- Any number of output terminals can be defined
- Filter criteria are XPath expressions in a table
- If no criteria match, then the message is propagated through the **default** terminal
- Usage:
 - Verify that the contents of a message meet conditions, such as the required fields are present
 - If the conditions are not met, invoke the error handler subflow or Fail primitive to stop processing
 - Route the message based on its contents (the value in the field indicates that service provider A is invoked; other values indicate that service provider B is invoked)
 - Conditionally bypass steps in a mediation based on the message content



Message Filter

© Copyright IBM Corporation 2016

Figure 14-15. Message Filter primitive

WB8601.2

Notes:

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. This attention is especially important with the **default** terminal, which receives a message that does not match any of the XPath expressions in the table.



Message Filter primitive: Configuration

Expression:
`/headers/MQHeader/md/MsgType = '8'`

Select a field:
<type to filter>

...	E md : MQMD
	E ReportUnit
	E MsgType : int
	E Expiry : int
	E Feedback : int
	E Priority : int
	E Persistence : int

Add an optional filter

Add an optional condition

Operator	Value
=	8

... becomes a row in the filter table

An XPath expression that is constructed by selecting fields and adding filters and conditions . . .

Enabled

Distribution mode: First

Filters:

Pattern	Terminal name
<code>/headers/MQHeader/md/MsgType = '8'</code>	match1

© Copyright IBM Corporation 2016

Figure 14-16. Message Filter primitive: Configuration

WB8601.2

Notes:

To configure the Message Filter primitive:

1. You 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. The XPath expression builder helps you construct the expression.
2. If that condition is met, you create a terminal (or use an existing one) to receive the incoming message.
3. Repeat those steps for all conditions that 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. (This distribution mode implies that the order in which expressions are listed 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 property can also be promoted, so it can be changed administratively without altering the mediation flow.

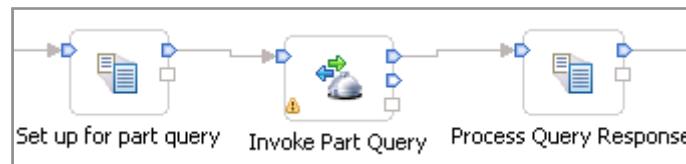
At run time, the table of expressions is scanned in sequential order. If an XPath expression evaluates to true, the incoming message is sent out through the terminal that is listed in the table for that expression. Depending on the value of the distribution mode property, the table scan either stops or continues.

Service Invoke primitive: Overview

- Similar to the callout node, but it can be used anywhere in a request flow
- Sends a message to (invokes) a requested service and operation
- Has **in** and **out** terminals for inbound and outbound messages
- Has a **timeout** terminal that is fired when the timeout threshold is exceeded, like a callout node
- Has a **<fault message name>** terminal for each modeled fault
 - When a modeled fault occurs, it propagates the modeled message that is returned from the invocation



Service Invoke



A sample flow that invokes a service

© Copyright IBM Corporation 2016

Figure 14-17. Service Invoke primitive: Overview

WB8601.2

Notes:

By using the Service Invoke primitive, it is not necessary to wait until the end of the mediation flow before invoking a service. In every other aspect, it behaves identically to the callout node.

Service Invoke primitive compared to callout node

- Service Invoke is embedded in a mediation flow such as callout, callout response, and callout fault
- Service Invoke mediation primitive versus callout:
 - Service Invoke mediation primitive does not switch from the request flow to the response flow; callout does
 - Service Invoke mediation primitive does not modify either the transient context or the correlation context; callout does
- When to use which:
 - To mediate a message (without calling an intermediate service) and call a service provider, use a callout
 - Use Service Invoke to call a service and return a result in a request flow without invoking the response flow; multiple Service Invoke primitives can exist in the same flow
 - To call an intermediate service, use Service Invoke and callout to invoke the final service
 - To invoke multiple intermediate services, use multiple Service Invoke primitives; you might not need the callout node
 - Interface and reference must not be wired when you use a Service Invoke to call a service and immediately return the response to the user
 - No response flow is needed in this case

© Copyright IBM Corporation 2016

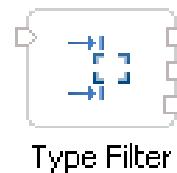
Figure 14-18. Service Invoke primitive compared to callout node

WB8601.2

Notes:

Type Filter primitive

- Routes a message within a mediation flow based on the message type
- Acts like an “if . . goto” statement to send a message through 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



© Copyright IBM Corporation 2016

Figure 14-19. Type Filter primitive

WB8601.2

Notes:

The Type Filter is similar to the Message Filter, except that 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. There is no means to match multiple rows in the expression table. Thus, the order in which the expressions are listed can be important.

14.3.Faults, tracing, and error handling primitives

Faults, tracing, and error handling primitives



© Copyright IBM Corporation 2016

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

10.1

Figure 14-20. Faults, tracing, and error handling primitives

WB8601.2

Notes:

Faults

- Three types of WSDL operations: input, output, and fault (can be multiple faults)
- For each fault that is defined in WSDL, a corresponding terminal is created on the input fault node
 - If you define the fault operation in WSDL (the interface editor) for a request flow, an **input fault** node is created
 - Corresponding nodes are created on the response flow for handling faults

Name	Type
query	IBM_AutoDataInterchangeFormatBO
response	IBM_AutoDataInterchangeFormatBO
Fault	FailureBusinessObject

© Copyright IBM Corporation 2016

Figure 14-21. Faults

WB8601.2

Notes:

You can define three types of WSDL operations on an interface. You see the interface editor here, displaying those operations. Whether you define a fault operation determines whether more nodes are added to the mediation to support WSDL error handling.

WSDL faults: Modeled

- Two types of WSDL faults: modeled and unmodeled
- **Request flow:** Modeled faults
 - Begins with an input node and (normally) ends with a callout node
 - If a fault is defined in WSDL for a request flow, an **input fault** node is created
 - If an error occurs during request processing, wire a branch of the request flow to the input fault node to return an error message to the caller
 - Do this wiring instead of invoking a callout
 - For each fault defined in WSDL, a corresponding terminal on the input fault node is created
- **Response flow:** Modeled faults
 - Begins with the callout response node and (normally) ends with the input response node
 - If a fault is defined in WSDL for a response flow, the **callout fault** node is created to handle errors that the called service generates
 - For each fault defined in WSDL, the corresponding terminal on the callout fault node is created
 - Can wire the callout fault node to the input fault node to send errors back to the client that the called service detects

© Copyright IBM Corporation 2016

Figure 14-22. WSDL faults: Modeled

WB8601.2

Notes:

The two types of WSDL faults are modeled and unmodeled. A modeled fault is one for which a fault is defined in the WSDL interface so that the mediation can have a defined reaction if that fault arises during the service invocation.

WSDL faults: Unmodeled

- Unmodeled faults:
 - Errors that a WSDL operation (a service invocation) returns, but are not defined in WSDL as a fault
 - No input fault node or callout fault node is created in the mediation flow
 - If an unmodeled fault occurs, the message is propagated through the fail terminal of the callout response node, and failure information is written to the `failInfo` element of the SMO context
 - If the response flow receives the unmodeled fault and the fail terminal of the callout response node is not wired, a runtime exception occurs

© Copyright IBM Corporation 2016

Figure 14-23. WSDL faults: Unmodeled

WB8601.2

Notes:

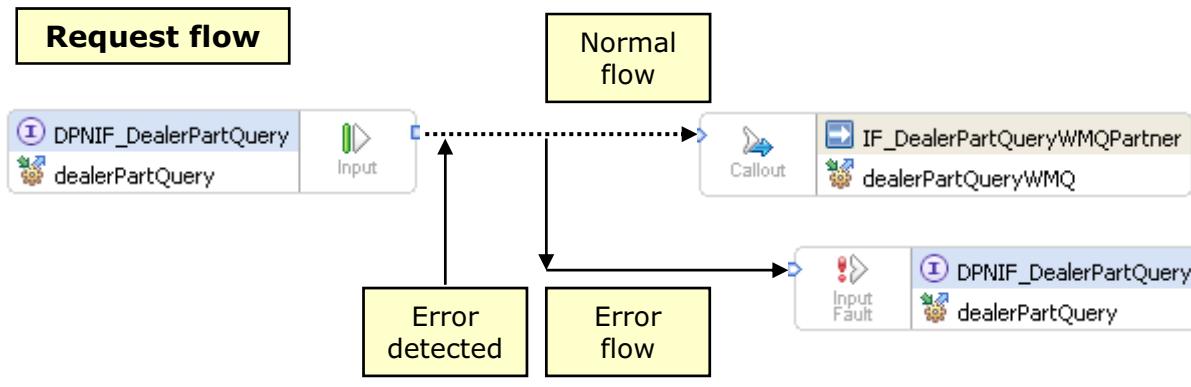
An unmodeled WSDL fault is a fault that is returned from a service invocation, but for which there is no means provided in the mediation flow to handle it. There is no means to handle it because the fault was not defined in the WSDL interface. How IBM Process Server responds to unmodeled faults depends on how the mediation is wired.



Handling WSDL faults: Request flow

Operations

		Name	Type			
dealerPartQueryWMQ						
	Inputs	query	IBM_AutoDataInterchangeFormatBO			
	Outputs	response	IBM_AutoDataInterchangeFormatBO			
	Fault	fault	FailureBusinessObject			



© Copyright IBM Corporation 2016

Figure 14-24. Handling WSDL faults: Request flow

WB8601.2

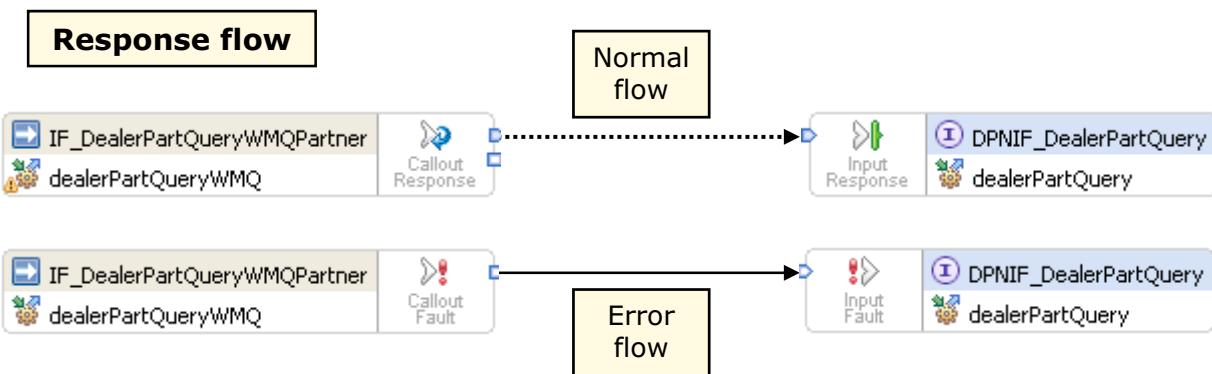
Notes:

In this example of a request flow, the WSDL interface is defined to include a fault operation. During the execution of the request flow, a determination is made that indicates that the message data has a problem. Rather than send the invalid data to the callout node, a primitive (such as the Message Filter) instead routes the message to the input fault node. The input fault node returns the message to the caller without doing the callout operation at the end of the request flow.

Handling WSDL faults: Response flow

Operations

Operations and their parameters	
Name	Type
dealerPartQueryWMQ	
Inputs	query
Outputs	response
Fault	fault



© Copyright IBM Corporation 2016

Figure 14-25. Handling WSDL faults: Response flow

WB8601.2

Notes:

In the normal situation in a response flow that follows a Service Invoke or callout operation, the response message flows to the input response node. However, in the situation where the service invocation returns a fault, the message travels from the callout fault node to the input fault node. Again, the assumption is that the fault is a modeled fault; that is, you provide for the fault in the WSDL interface. If an unmodeled fault occurs, the behavior is as described in the “WSDL faults: Unmodeled” slide (two visuals previously), which uses the fail terminal of the Service Invoke, or a runtime exception occurs.

 **WebSphere Education** 

Mediation primitives: Error handling, debugging, and event recording

Mediation primitive	Description
Event Emitter	Sends a monitoring event (in the Common Base Event format) from within a mediation flow to a Common Event Infrastructure server
Fail	Stops processing in a mediation flow and generates an exception
In*	Is used as an entry point in a subflow
Out*	Is used as an exit point in a subflow
Message Logger	Stores messages in a relational database or other medium
Message Validator*	Validates all or part of a message against its schema
Stop	Stops processing in a mediation flow, without generating an exception
Trace	Writes trace messages to a server log or to log files

*See appendixes for details

© Copyright IBM Corporation 2016

Figure 14-26. Mediation primitives: Error handling, debugging, and event recording

WB8601.2

Notes:

These built-in mediation primitives are used for error handling in mediation flows and debugging and recording or logging of events that occur at run time. Some primitives are included in the appendixes as reference material. Time constraints prevent coverage of all the primitives in this unit.

Message Logger primitive

- Stores a message in a relational database table
- Can also write a message to another medium by using the custom log facility
- Saves a selectable amount of the 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



Message Logger

© Copyright IBM Corporation 2016

Figure 14-27. Message Logger primitive

WB8601.2

Notes:

The following list is an overview of the Message Logger properties:

- **Enabled:** Determines whether the logger does its function; it allows logging to be disabled without removing the primitive. The administrator promotes or sets this property.
- **Root:** XPath expression that describes the message to log. The message to log is converted to XML from the point that Root specifies.
- **Transaction mode:** Determines whether Message Logger participates in a mediation transaction or acts independently.
- **Logging type:** The destination type for the logging event: database or custom (uses the custom logging 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 that is named CommonDB.
- **Handler, Formatter, Filter, Literal, Level:** Implementation classes that are used when custom logging is enabled.

By default, messages are written to a data source that is named CommonDB, to a table that is named MSGLOG.

By using the default value for the Literal property, the call to

`MessageFormat.format(<LogRecord>.getMessage(), <LogRecord>.getParameters())` in the default Formatter implementation class means the following values:

- {0} would then be replaced with the Time Stamp value: `logMessageParameters[0]`
- {1} would then be replaced with the Message ID value: `logMessageParameters[1]`
- {2} would then be replaced with the Mediation Name value: `logMessageParameters[2]`
- {3} would then be replaced with the Module Name value: `logMessageParameters[3]`
- {4} would then be replaced with the Message value: `logMessageParameters[4]`
- {5} would then be replaced with the version value: `logMessageParameters[5]`

Trace primitive

- Writes a message to the server log, user trace log file, or another file
- Can be used for debugging or tracing a mediation flow
- Saves a selectable amount of the message content plus identifying information (time stamp, message identifier, primitive and module names, and other information)
- Original inbound message passes through without alteration



© Copyright IBM Corporation 2016

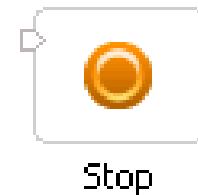
Figure 14-28. Trace primitive

WB8601.2

Notes:

Stop primitive

- Stops the execution of a mediation flow
- Consumes the incoming message silently and then stops the flow
- Contains only an **in** terminal, no **out** or **fail** terminals
- Has no configurable properties
- If wired to a normal output terminal, the behavior is the same as if the output terminal were unwired (the message is lost, silently)
- If wired to a fail terminal, the exception from the fail terminal is consumed silently, rather than propagated



© Copyright IBM Corporation 2016

Figure 14-29. Stop primitive

WB8601.2

Notes:

Fail primitive

- Stops the execution of a mediation flow
- Consumes the incoming message, raises a `FailFlowException`, and then stops the flow
- You can specify an error message, and define part or all of the SMO to include with the exception (the default SMO content that is listed is `/context/failInfo`)
- Uses the same message format (with substitution variables) as the Message Logger primitive



© Copyright IBM Corporation 2016

Figure 14-30. Fail primitive

WB8601.2

Notes:

The Fail primitive uses the same message format and content as the Message Logger primitive, including substitution variables `{0}` through `{5}`:

- `{0}` would then be replaced with the Time Stamp value: `logMessageParameters[0]`
- `{1}` would then be replaced with the Message ID value: `logMessageParameters[1]`
- `{2}` would then be replaced with the Mediation Name value: `logMessageParameters[2]`
- `{3}` would then be replaced with the Module Name value: `logMessageParameters[3]`
- `{4}` would then be replaced with the Message value: `logMessageParameters[4]`
- `{5}` would then be replaced with the version value: `logMessageParameters[5]`

If you supply the optional error message in the Properties view of the Fail primitive, it is substituted for message `{4}`.



Unit summary

- Describe the role of mediation primitives in mediation flows
- Describe the prebuilt mediation primitives that are available for mediation flows

© Copyright IBM Corporation 2016

Figure 14-31. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. True or False: A Service Invoke mediation primitive allows a service to be invoked from within a request or response flow, rather than waiting to reach a callout node.
2. True or False: It is possible for more than one condition in a Message Filter primitive expression table to match.
3. True or False: Using the Event Emitter to log every incoming message for audit purposes would be considered a good practice.
4. True or False: You can configure the Stop primitive to display a user-defined message when it stops the mediation flow.

© Copyright IBM Corporation 2016

Figure 14-32. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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



Checkpoint answers

1. True.
2. True, only if the distribution mode property is set to “all.”
3. False: An Event Emitter should not normally be in the main flow of a mediation because of the message traffic (and related potential effect on performance).
4. False: The Stop primitive has no configurable properties, but you can set the Fail primitive to display a user-defined message.

© Copyright IBM Corporation 2016

Figure 14-33. Checkpoint answers

WB8601.2

Notes:

Exercise 12



Creating mediation services, part II

© Copyright IBM Corporation 2016

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

10.1

Figure 14-34. Exercise 12

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create a mediation module that contains a Message Filter mediation primitive and a Mapping primitive
- Define an XML data map
- Test a mediation module that contains a Message Filter mediation primitive and a Mapping primitive

© Copyright IBM Corporation 2016

Figure 14-35. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 12

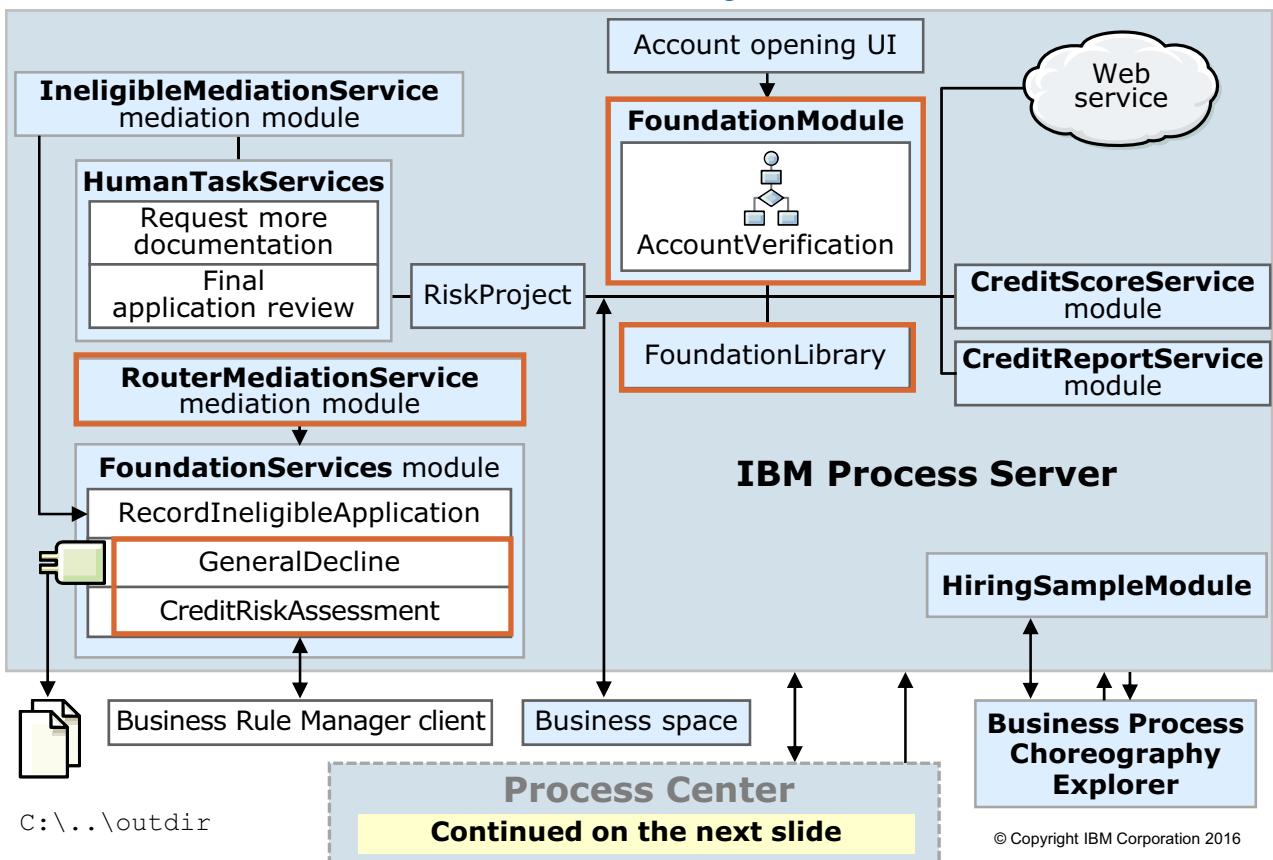


Figure 14-36. Course business scenario: What you work with in Exercise 12

WB8601.2

Notes:

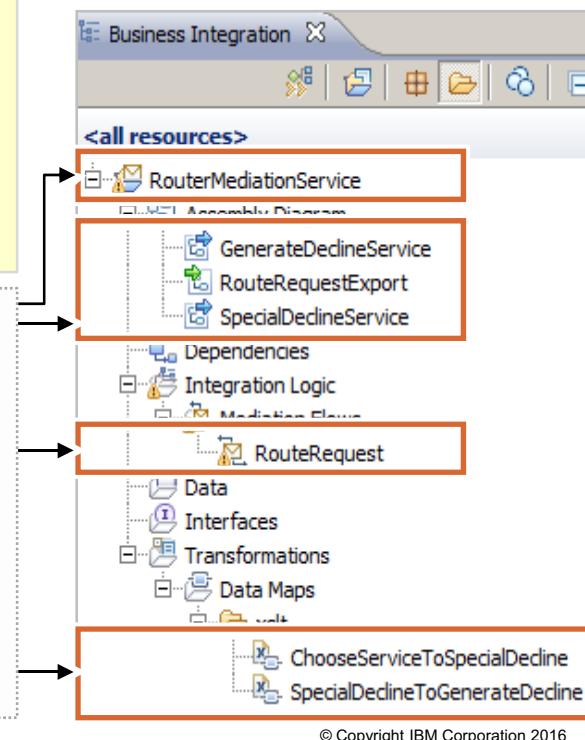
Components that are required for Exercise 12 (1 of 2)

Prebuilt components that are imported in this lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **FoundationServices**
5. **CWYFF_FlatFile**
6. **IneligibleMediationService**

New components that you create in this lab:

1. **RouterMediationService** module
2. **GenerateDeclineService** import
3. **RouteRequestExport** export
4. **SpecialDeclineService** import
5. **RouteRequest** mediation flow
6. **ChooseServiceToSpecialDecline** map
7. **SpecialDeclineToGenerateDecline** map



© Copyright IBM Corporation 2016

Figure 14-37. Components that are required for Exercise 12 (1 of 2)

WB8601.2

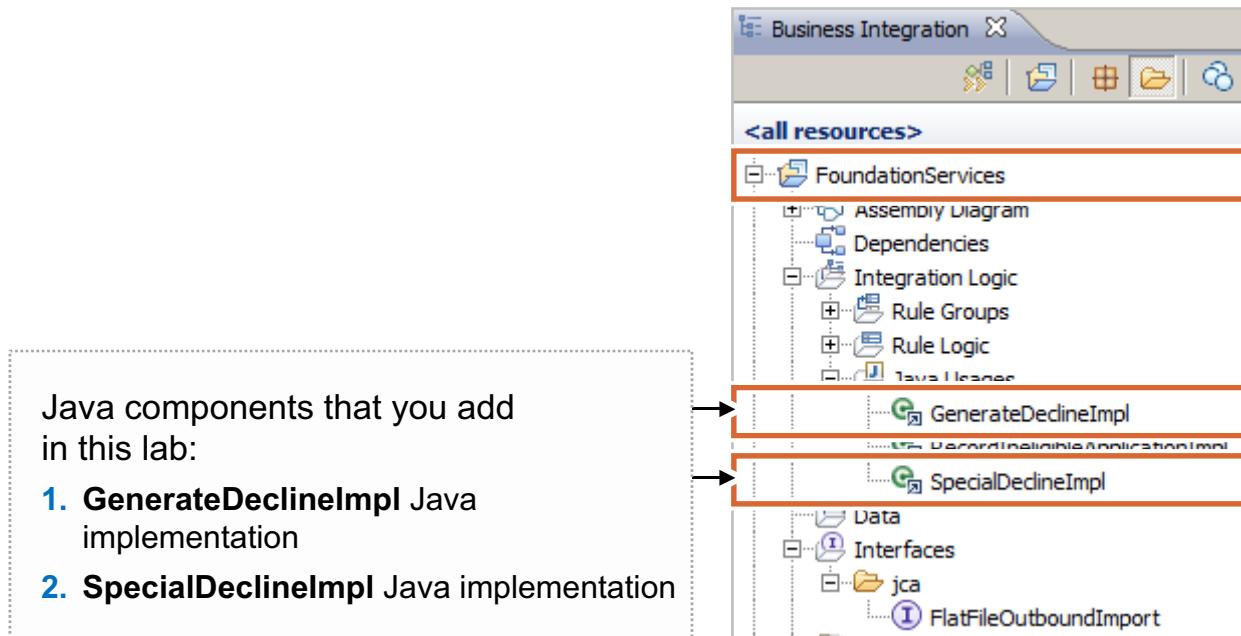
Notes:

If applicationDecision is set to false during FinalApplicationReview (the application is declined) and the customer's creditRisk is HIGH, the application is routed through the "generate decline" component. If applicationDecision is set to false during FinalApplicationReview and the customer's creditRisk is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the RouteRequest component. The RouteRequest flow component contains the mediation logic that routes the application to the appropriate decline service.

The RouteRequest mediation flow consists of both a request flow and a response flow. In the flow, the CustomerApplication is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the AccountVerification process.

Components that are required for Exercise 12 (2 of 2)



© Copyright IBM Corporation 2016

Figure 14-38. Components that are required for Exercise 12 (2 of 2)

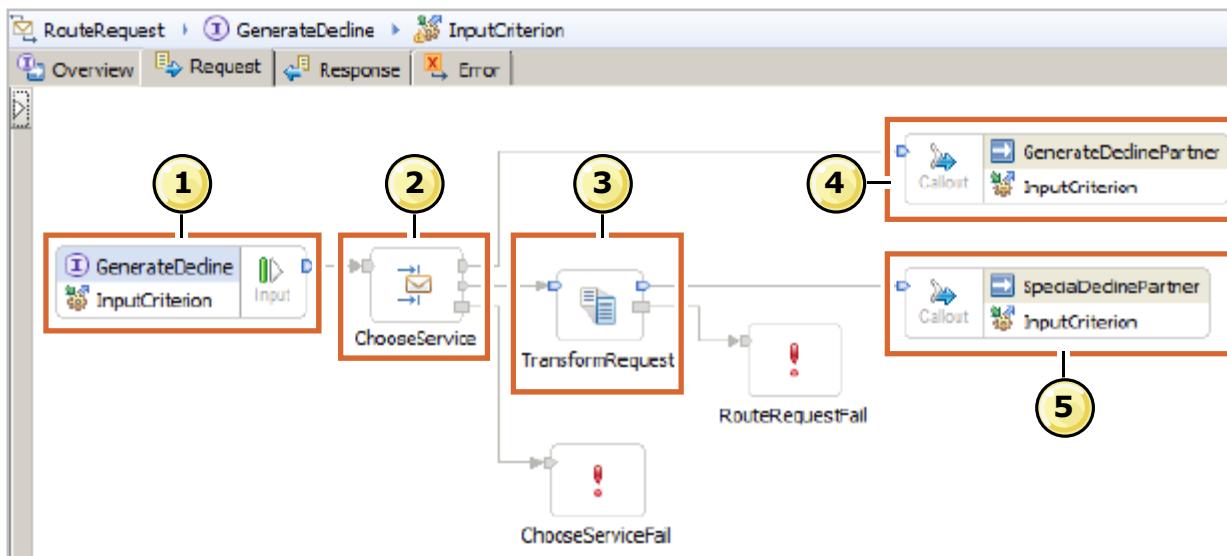
WB8601.2

Notes:

If the employee approves the application, the process completes successfully. However, if the application is declined, it is routed to one of two possible services. If the application is declined and the creditRisk is HIGH, the generate decline service is invoked. It is the **GenerateDeclineImpl** implementation code that you add in this exercise. The code returns a Message business object that contains the message: "Account for customer <company name> was declined and the credit risk was <credit risk>."

If the application is declined and the creditRisk is MED, the special decline service is invoked. It is the **SpecialDeclineImpl** implementation code that you add in this exercise. The "special decline" service is called when the application is declined during FinalApplicationReview, and the creditRisk is MED. The code in SpecialDecline returns a Message business object that contains the message: "Account for customer <company name> was routed through special decline because the credit risk was <credit risk>."

Create the RouteRequest mediation flow in Exercise 12



© Copyright IBM Corporation 2016

Figure 14-39. Create the RouteRequest mediation flow in Exercise 12

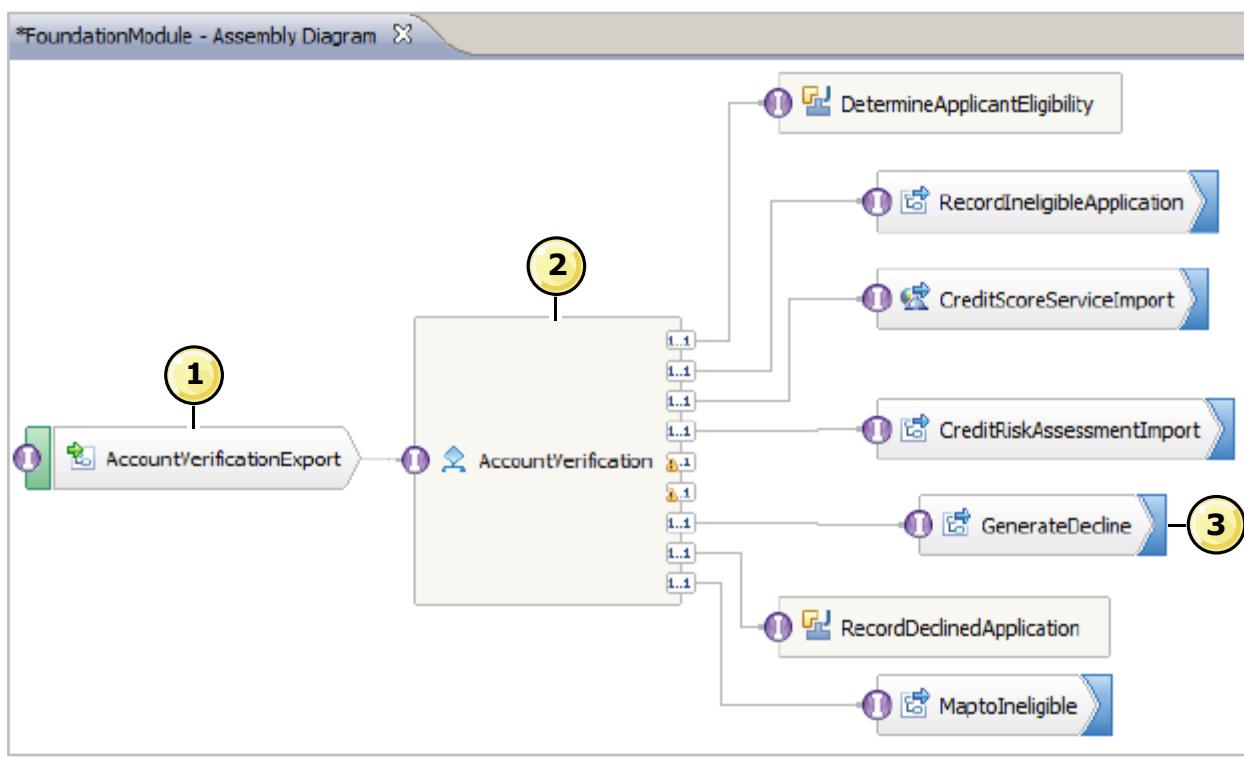
WB8601.2

Notes:

If applicationDecision is set to false during FinalApplicationReview (the application is declined) and the customer's creditRisk is HIGH, the application is routed through the "generate decline" component. If applicationDecision is set to false during FinalApplicationReview and the customer's creditRisk is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the RouteRequest component. The RouteRequest flow component contains the mediation logic that routes the application to the appropriate decline service. The RouteRequest mediation flow consists of both a request flow and a response flow. In the flow, the CustomerApplication is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the AccountVerification process.

Assemble SCA components in Exercise 12 (1 of 2)



© Copyright IBM Corporation 2016

Figure 14-40. Assemble SCA components in Exercise 12 (1 of 2)

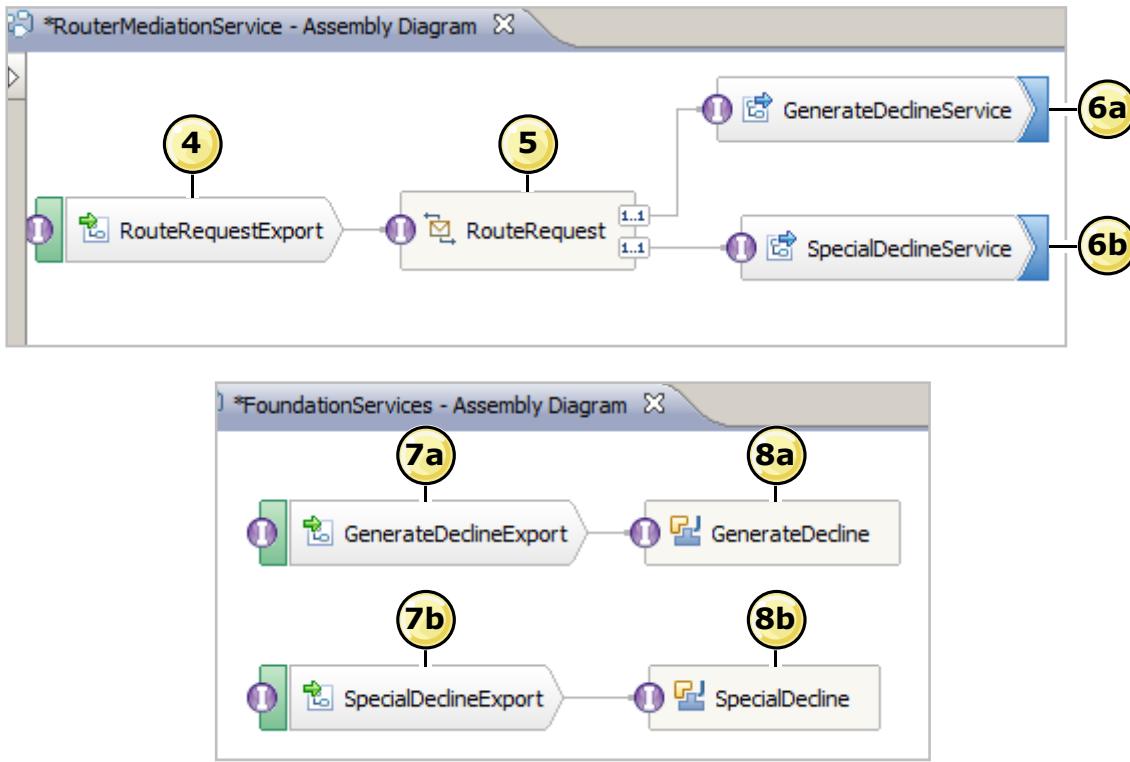
WB8601.2

Notes:

The following steps are illustrated in the diagram:

- Step 1:** The **AccountVerificationExport** component exposes the **AccountVerification** business process.
- Step 2:** If the Application is declined link is followed, the **Generate Decline** invoke activity is processed. The **Generate Decline** activity uses the **GenerateDeclinePartner** reference partner to call the **GenerateDecline** import component on the **FoundationModule** assembly diagram.
- Step 3:** The **GenerateDecline** import component invokes the **RouteRequestExport** component on the **RouterMediationService** assembly diagram.

Assemble SCA components in Exercise 12 (2 of 2)



© Copyright IBM Corporation 2016

Figure 14-41. Assemble SCA components in Exercise 12 (2 of 2)

WB8601.2

Notes:

The following steps are illustrated in the diagram:

- **Step 4:** RouteRequestExport exposes the services of the RouteRequest mediation flow.
- **Step 5:** If the creditRisk is HIGH and applicationDecision is false, the message is routed to the GenerateDeclineService import component. If the creditRisk is MED and applicationDecision is false, the message is routed to the SpecialDeclineService import component.
- **Step 6a:** The GenerateDeclineService import invokes the GenerateDeclineExport component on the FoundationServices assembly diagram.
- **Step 6b:** The SpecialDeclineService import invokes the SpecialDeclineExport component on the FoundationServices assembly diagram.
- **Step 7a:** The GenerateDeclineExport exposes the services of the GenerateDecline Java component.
- **Step 7b:** SpecialDeclineExport exposes the services of the SpecialDecline Java component.
- **Step 8a:** This component sets the message element of the Message business object to: Account for customer TestCo was declined and the credit risk was HIGH.

- **Step 8b:** This component sets the message element of the Message business object to: Account for customer ACME was routed through special decline because the credit risk was MED.

Unit 15. Business Space

What this unit is about

This unit explains the business value of Business Space and describes how you can use it to interact with content from products in the business process management portfolio.

What you should be able to do

After completing this unit, you should be able to:

- Describe the purpose and business value of Business Space
- Describe the types of widgets that are available in Business Space
- Describe the types of templates that are available in Business Space
- Explain how to create and configure a new space in Business Space

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the purpose and business value of Business Space
- Describe the types of widgets that are available in Business Space
- Describe the types of templates that are available in Business Space
- Explain how to create and configure a new space in Business Space

© Copyright IBM Corporation 2016

Figure 15-1. Unit objectives

WB8601.2

Notes:

15.1.Topics



Topics

- Introduction to Business Space
- Business Space terminology
- Configure a new space

© Copyright IBM Corporation 2016

Figure 15-2. Topics

WB8601.2

Notes:

15.2. Introduction to Business Space

Introduction to Business Space



© Copyright IBM Corporation 2016

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

10.1

Figure 15-3. Introduction to Business Space

WB8601.2

Notes:

Problem statement that Business Space addresses

- A need for a common front end for products in the business process management portfolio
- Create a front end for users of IBM Business Process Manager that is not product-centric
- Move away from a product-centric solution to a user solution
- Bring users of different roles to a page to solve a problem
- Make Business Space the entry point into the business process management product portfolio

© Copyright IBM Corporation 2016

Figure 15-4. Problem statement that Business Space addresses

WB8601.2

Notes:

What is Business Space?

- Business Space is a user interface framework for aggregating content in a web browser
- Within the framework, a business space is a collection of web content for a particular business task or role
- The Business Space web application contains multiple spaces, each with one or more pages
 - Each page consists of a collection of widgets
 - Widgets can use events to interact with each other
- Bundled with IBM Business Process Manager Advanced

© Copyright IBM Corporation 2016

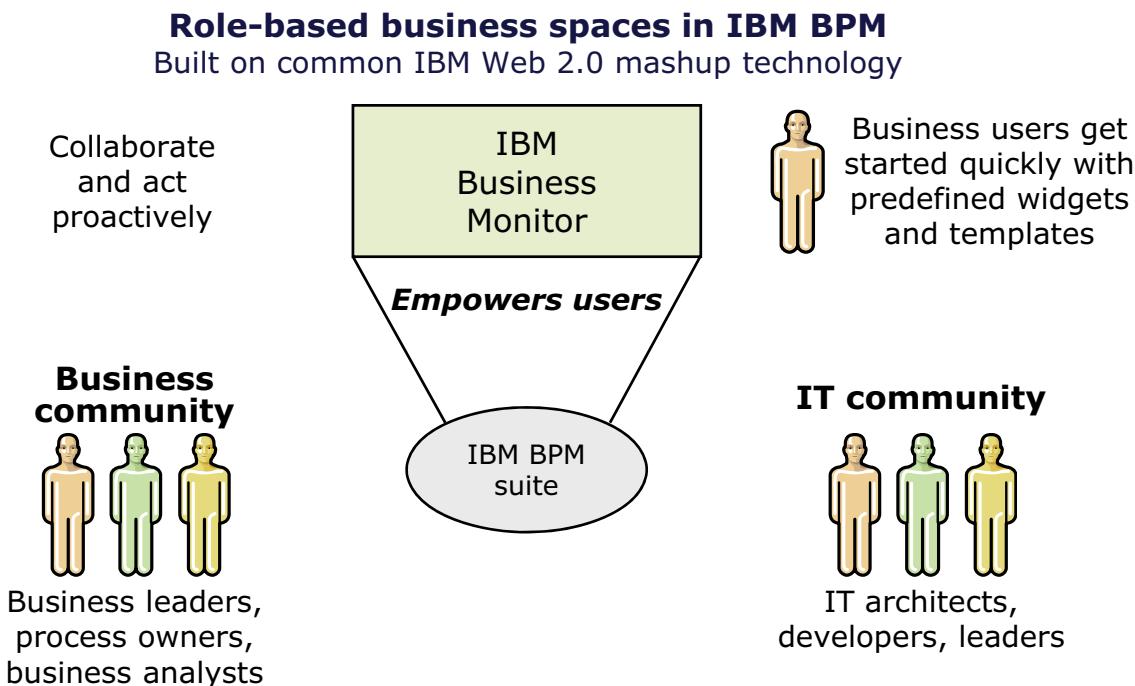
Figure 15-5. What is Business Space?

WB8601.2

Notes:

Business Space is a ready-to-use Web 2.0 IBM BPM client for business users. It is a collection of related web content that provides you with insight into your business and the capability to react to changes within it. Business Space unifies the IBM Business Process Manager user interface space for business users. It allows users to use a single window when working with all of the business activities that take place in different products in the IBM Business Process Manager suite. Business Space is a mashup of IBM Business Process Manager widgets that are targeted for a business user. Business Space is not part of the Express or Standard product but is bundled with IBM Business Process Manager Advanced.

Business Space: A unified environment



© Copyright IBM Corporation 2016

Figure 15-6. Business Space: A unified environment

WB8601.2

Notes:

The business space provides a rich, holistic, and transformative experience to users across the business process management and connectivity product portfolio stack. The business space is a common infrastructure component that comes with all business process management products.

It empowers the business user by enabling key processes (for example, dashboards, point-of-business agility, and composite application policy).

The business space integrates the creation of the user experience with the authoring of the business application itself:

- It allows both business users and IT developers to create and deliver rich content for a broader audience of BPM users across a range of roles.
- It uses the same capabilities for empowering the user experience that is built for the business application.

It also allows collaboration between all participants in the BPM process:

- It provides different levels of collaboration in day-to-day activities across role boundaries: human tasks and coordination, strategy, review, modeling of processes, and organizational navigation.

It transforms IT administration into a business-centric and solution-centric experience:

- It shifts the focus of the user experience away from deployment-centric to business application-centric administration and integrates IT administration into the BPM lifecycle optimization process.

Key roles (participants) in business process management

Business leader ★

Responsible for overall business performance, compliance, and governance

Business user ★

Participates in or is a user of a business process

Business analyst ★

Analyzes and simulates improvements to business process performance

Process owner ★

Owns or serves as a stakeholder in a business process and is empowered to make process changes

IT leader

A business leader responsible for delivering technology solutions that enable the business

IT architect

Uses and extends business assets to implement business processes and applications

IT developer ★★

Implements processes and constructs business process management applications

BPM administrator ★

Ensures smooth operation of one or more business process management applications and processes

KEY

★ Audience of the business space

★★ Uses the business space

© Copyright IBM Corporation 2016

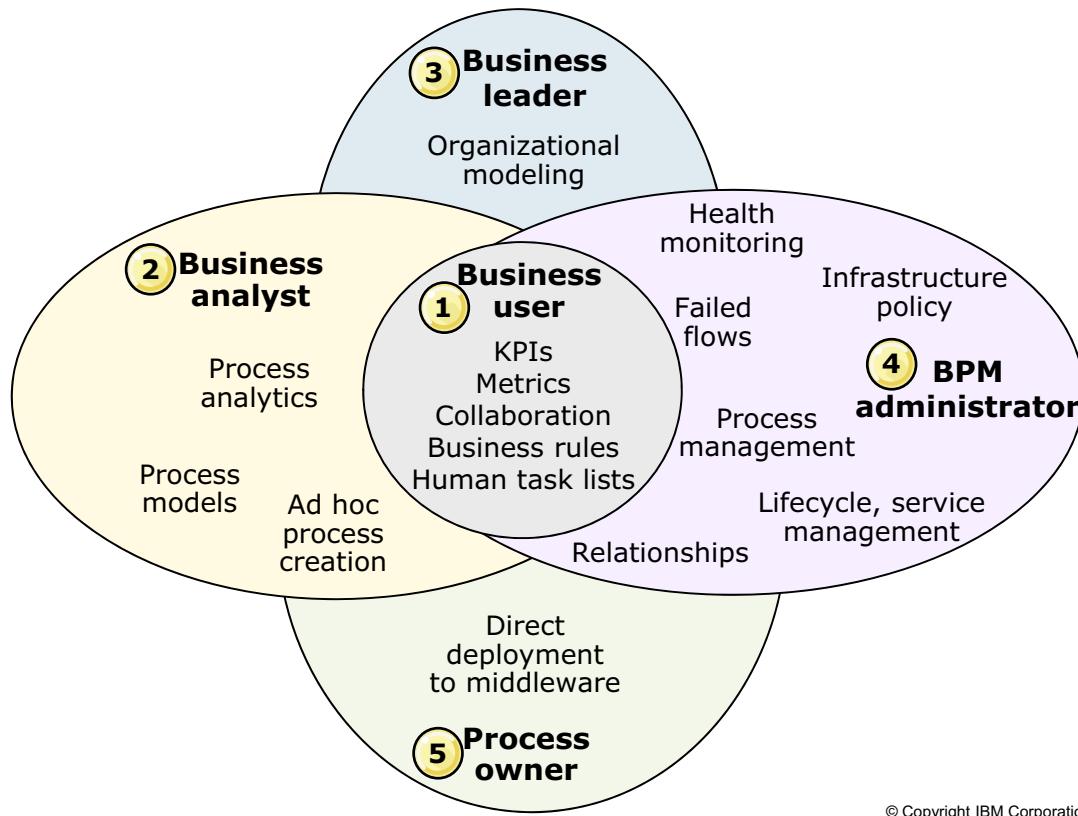
Figure 15-7. Key roles (participants) in business process management

WB8601.2

Notes:

The technology behind the business space provides a Web 2.0 user with a view of IBM BPM data for a business user. The technology provides a holistic view into how business users can handle their day-to-day interaction with IBM BPM data. The overriding concept with the business space is to provide content and solutions from a common front end.

Integration with the BPM portfolio



© Copyright IBM Corporation 2016

Figure 15-8. Integration with the BPM portfolio

WB8601.2

Notes:

The business space is developed to integrate the Business Process Management portfolio around five scenarios:

1. The business space for managing tasks and human workflow: This space helps business users to process tasks efficiently and collaborate to get the job done.
2. The business space for initiating process improvement: This space helps change business performance through analysis, simulation, and refinement of business process decisions.
3. The business space for managing business performance: This space is used to harvest information about the business to make better business decisions and coordinate execution.
4. The business space for solution management: Ensures that business applications keep running optimally and that they support the business.
5. Generating a business space to experiment with the business processes: This scenario provides business agility through automation and direct refinement of business processes.

15.3. Business Space terminology

Business Space terminology



© Copyright IBM Corporation 2016

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

10.1

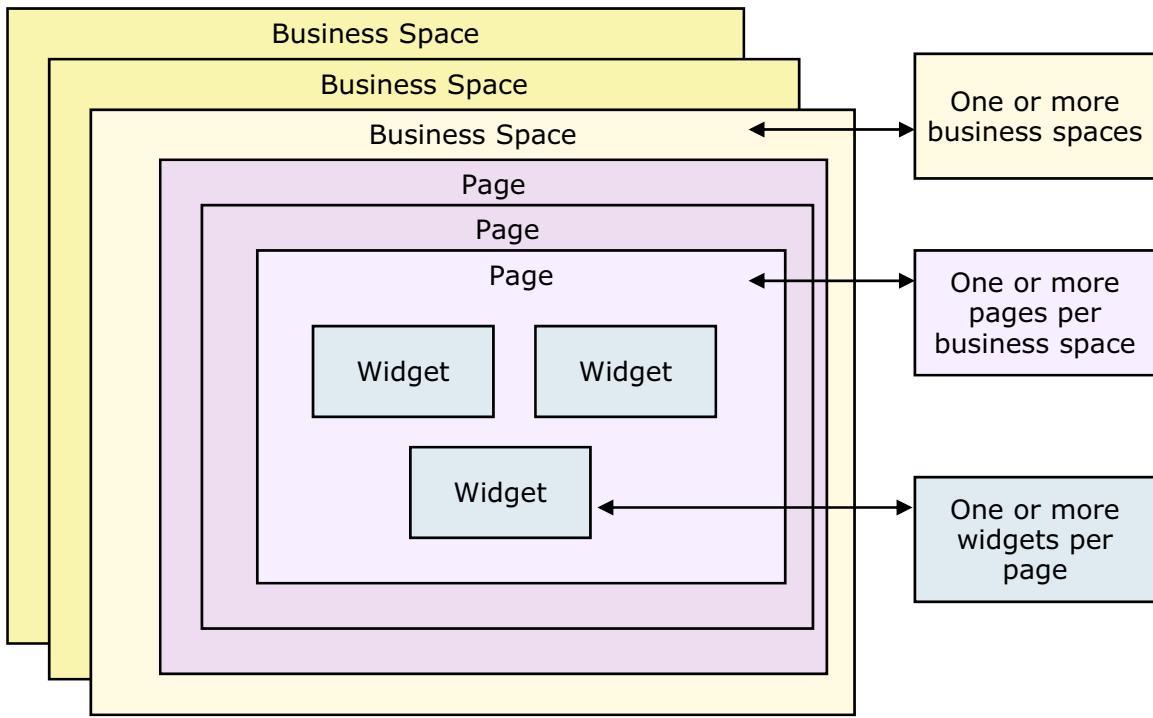
Figure 15-9. Business Space terminology

WB8601.2

Notes:

Business Space hierarchy

- A business space is a web page that has several components:



© Copyright IBM Corporation 2016

Figure 15-10. Business Space hierarchy

WB8601.2

Notes:

The Business Space viewing web page consists of one or more business spaces. Each business space can contain one or more tabbed pages, and each page can contain one or more widgets. As a business user, you control how many business spaces you want on your Business Space web page. Also, for each business space, you control how many pages and widgets you would like to see. Templates are provided for specific business spaces that are based on the specific business functions, which are based on different roles of a business user.

WebSphere Education

Pages (1 of 2)

- Assemble widgets to construct your business space in pages
- If you are the owner or editor of a page, you can create, delete, and modify the page and the widgets that it contains

Create Page

* Page name:
MyPage

Page description:

Create an empty page
 Create a page by duplicating another page

© Copyright IBM Corporation 2016

Figure 15-11. Pages (1 of 2)

WB8601.2

Notes:

Pages are the medium in which you assemble widgets to construct your business space. If you are the owner or editor of a page, you can create, delete, and modify the page and the widgets that it contains.

- Creating pages:** If you can edit a business space, you can create one or more pages in it. You can then place widgets on the pages. You can import a page into a business space that you own or can edit. Importing is a way of creating a page that is already populated with configured widgets. You can export a page from a business space. Exporting a page is useful if you want to give someone a copy of the page without giving them access to it or if you want to re-create the page later.
- Switching pages:** If you are viewing a page in a business space, you can switch to a different page. When you switch pages, Business Space stores which page you last went to in the most recent nine spaces. When you return to one of these spaces, Business Space displays the last open page.
- Editing pages:** You can edit a page in it to add or remove widgets or change where the widgets are displayed in the page.

- **Deleting pages:** If you own a page, you can delete it from its business space. However, you cannot restore deleted pages. If you think you might need the page, export it. The page can then be restored by importing it.



- Layout of widgets, their configuration, and an event flow between widgets
- Pages are also known as a mashup

A screenshot of the IBM WebSphere Business Space interface. The top navigation bar includes links for "Initiate Tasks and Processes", "Work on tasks" (with a dropdown arrow), "Manage Processes", "Manage Tasks", and "Organizations". The main content area features several widgets: a "Tasks" section titled "All - My work" with a table header (Name, Priority, Status, Due date) and a message stating "No tasks were found"; an "Escalations" section titled "All - Escalate tasks" with a similar table header and a message stating "No escalations were found"; a "Task Information" panel with the instruction "Select the task and then select an action"; and a "Process Information" panel with the instruction "Select the process and then select Open". At the bottom right of the interface, there's a copyright notice: "© Copyright IBM Corporation 2016".

Figure 15-12. Pages (2 of 2)

WB8601.2

Notes:

To organize the widgets on a business space page in a different pattern, or to use the screen area more efficiently, you can change the layout of the page.

A page acts as a container for one or more widgets. If more than one widget is placed on a page, it becomes a mashup. By definition, a mashup combines data from more than one source into a single integrated tool. This example shows six widgets that are placed on the page that is named My Work.



Templates (1 of 2)

- Prearranged mashup pages that can quickly be used to create a business space instance that contains pages and widgets
- Prebuilt and configured business spaces
- Expedites creation of business spaces
- Categorized according to function

© Copyright IBM Corporation 2016

Figure 15-13. Templates (1 of 2)

WB8601.2

Notes:

You can configure the business space more easily by using the preconfigured templates that come with a business space. You can access the templates when you create a business space. In addition to the provided templates, you can also create customized templates from business spaces. If you create a business space that you want to save as a template to share with other users, a superuser can convert the business space into a template.

The template can then be shared with other users. If any of these users change any of the pages in the customized template, then the changes are forwarded to other users who share the template. However, because future updates to the business space can overwrite this type of change, the superuser must create a copy of the original template before doing any modifications.



Templates (2 of 2)

- Product templates have widgets from a single product
 - The Create Space window lists only the product templates for the products that are installed
- Cross-product templates have widgets from more than one product
 - The Create Space window lists all of the cross-product templates
 - If a product is not installed, its templates are not available for selection when you are creating a business space
 - Templates must be in the same profile in WebSphere Application Server

© Copyright IBM Corporation 2016

Figure 15-14. Templates (2 of 2)

WB8601.2

Notes:

Cross-product templates are templates that contain widgets from more than one product within the IBM BPM portfolio. Before users can create spaces that are based on these templates, the administrator must import them and make them available.

Templates available in IBM Process Manager Advanced

- Templates for managing processes and tasks
 - Advanced Management of Tasks and Processes template
 - Configure Work Baskets and Business Categories template
 - Integrated Inbox template
 - Interact with the Processes and Tasks template
 - Work on Tasks Continuously template



© Copyright IBM Corporation 2016

Figure 15-15. Templates available in IBM Process Manager Advanced

WB8601.2

Notes:

Several templates are provided for you to create business spaces to manage processes and tasks. These templates support specific usage patterns for working with and managing tasks and processes.

- **Advanced Management of Tasks and Processes template:** Use the Advanced Management of Tasks and Processes template to create a space for business users and team leads who work on tasks and collaborate with other people. You can also use this template to organize your own task lists, check the status of processes you are involved with or responsible for, and check tasks that are related to these processes.
- **Configure Work Baskets and Business Categories template:** Use the Configure Work Baskets and Business Categories template to create a space for users who are responsible for the configuration and management of work baskets and business categories within the organization.
- **Integrated Inbox template:** Use the Integrated Inbox template to create a space for people who must work with both IBM Business Process Manager tasks and IBM Case Manager work items. Users of this space can work on tasks and work items from their inbox.

- **Interact with Processes and Tasks template:** Use the Interact with Processes and Tasks template to create a space for people who use a list of tasks – for example, an inbox – as the basis for their work. Users of this space can also create processes and services, and find detailed status information for specific process instances.
- **Work on Tasks Continuously template:** Use the Work on Tasks Continuously template when you must create a space for users who continuously work on tasks for a task queue. When users complete tasks, they get the next available task in the queue assigned to them automatically.



Business space

- Collection of mashup pages that are organized into tabs
- Typically represents a collaborative user interface or an interface for an IBM BPM application
- Authorization controls determine who can view and edit a space

A screenshot of the MyBusinessSpace interface. At the top, there's a navigation bar with links: Home, Process Portal, Go to Spaces, Manage Spaces, Actions, and a dropdown. Below the navigation is a title bar with the text "MyBusinessSpace". Underneath is a toolbar with tabs: Initiate Tasks and Processes (selected), Work on tasks, Manage Processes, Manage Tasks, and Organize Work. The main area is divided into two panels: "Task Definitions" on the left and "Task Information" on the right. The "Task Definitions" panel has a heading "All" with a "Create" button, a checkbox, and a "Sort By Name" dropdown. The "Task Information" panel contains the message "Select the task and then select an action." At the bottom right of the interface, there's a copyright notice: "© Copyright IBM Corporation 2016".

Figure 15-16. Business space

WB8601.2

Notes:

A business space is a collection of related web content that provides you with insight into your business and the capability to react to changes in it.

In Business Space, you can have many business spaces with each one having a different purpose. For example, a business space with widgets from IBM Business Monitor might be used to monitor key performance indicators in your business and widgets from IBM Process Manager to manage the tasks that people do. Business Space can display the contents of one business space at a time. This space is the open space.

The open space consists of a banner area and a page area. The top of the banner consists of menus and links. The menus contain options that affect the open business space or options for browsing to another space. The links in the top banner go to the help or log you out of Business Space. Below the menus, the banner area displays the name of the open space and tabs for the pages in the space. The page area is below the banner and displays an open page. The tab for the page is highlighted.

If you are the owner of the open page or an editor, an **Edit Page** button is also available. When you click **Edit Page**, the editing toolbar opens. The toolbar contains the palette of widgets and several buttons. By using these buttons, you can see the hidden widgets on the page and save changes.

The toolbar also contains two fields that you can use to filter widgets from the palette. One field removes all widgets that do not belong to a selected category, while the other removes all widgets that do not contain the characters that you specify in their name or description.

The page area is where Business Space displays the visible widgets on the page.

You can create a business space in any of the following ways:

- Using the widget palette to define the tasks
- Using a preconfigured template
- Importing an existing business space



Widget

- Configurable piece of a graphical user interface function, provided as an embeddable component
- Multiple instances can be on the same mashup page, each with a different configuration

The screenshot shows the 'MyBusinessSpace' application interface. At the top, there is a navigation bar with tabs: 'Initiate Tasks and Processes', 'Work on tasks', 'Manage Processes', and 'Manage Tasks'. Below the navigation bar is a search bar labeled 'All Widgets (43)' with a dropdown arrow and a magnifying glass icon. The main area displays a grid of 12 widget icons, each with a label:

Available Tasks	Business Calendars	Business Categories
Business Category Information	Create Tasks	Document
Escalations	Google Gadgets	Human Workflow Diagram
Inbox	Mediation Policy Administration	Module Administration

© Copyright IBM Corporation 2016

Figure 15-17. Widget

WB8601.2

Notes:

A business space comprises one or more pages. Each page contains one or more widgets.

Widgets are the pluggable user interface components that you use to define the functions of your business spaces.

Typically, a single widget has limited or specific capabilities. However, you can combine widgets to interact with each other to do related tasks. Combined widgets are called a *mashup*. For example, you can have a widget that displays news items from an RSS feed and another widget that you can use to create tasks that you assign to someone. If you combine the two widgets onto one page, you create a mashup. In this mashup, you can see a news event and react to it by assigning someone to investigate whether the event might affect your business. Some widgets communicate with other widgets so that the events in one widget affect the contents of a different widget.

You can minimize, maximize, and drag widgets while laying out a page. In addition, each widget has a menu that contains actions that you can take on the widget and a Help option, which links to information about that widget. In addition, you can convert a widget into a hidden widget. A hidden widget is fully functional but is not displayed on a page. Typically, you hide widgets such as the Script Adapter widget, which, instead of displaying server data, transforms business data so that

another widget can use it. Other than not being displayed on a page, a hidden widget is like a visible widget. That is, you can change its settings and wire it to other widgets.

Business Space provides a widget palette that contains categories of widgets that you can use to configure the pages in your business spaces. You can use the categories to filter out the widgets that are not in the category that you select. You can see all of the available widgets by selecting **All Widgets** from the list of categories.



Problem determination widgets

The figure consists of two side-by-side screenshots of the WebSphere Business Space interface.

System Health Widget:

- Title:** System Health
- Section:** Stand-alone Servers
- Details:** Showing 1 of 1, 1 started, 0 stopped.
- Table Headers:** Status, Name, Node
- Data:** server1, xpbaseNode01
- Buttons:** A blue button labeled '1' is highlighted.

Module Health Widget:

- Title:** Module Health
- Section:** Topology
- Details:** Module: FoundationModule, Version:, Cell Identifier:
- Table Headers:** Status, Deployment Environment, Cluster, Server, Type, Node
- Data:** server1, xpbase
- Buttons:** A blue button labeled '1' is highlighted.

© Copyright IBM Corporation 2016

Figure 15-18. Problem determination widgets

WB8601.2

Notes:

Administration widgets offer a way to manage and monitor the individual components of your overall business process management solution, including modules and services.

The administration widgets are grouped on pages in the following templates:

- The Solution Administration template contains the widgets that you need for administering the modules in your solution.
- The Service Administration template contains the widgets that you need for monitoring and working with the services that are used in your solution.
- The Problem Determination template contains the widgets that you need for monitoring the health of your modules and of your overall system.

These widgets are delivered as part of Business Space and function in the same way as other widgets. You can minimize, maximize, and drag widgets while laying out a page. In addition, each widget has a menu that contains actions that you can take on the widget.

The System Health widget and the Module Health widget are commonly used problem determination widgets.

- **System Health** widget: Use the System Health widget to view a snapshot of the overall system health of your business solution. This widget provides a single place from which you can quickly assess the status of application servers, nodes, clusters, deployment environments, messaging engines and their queues, databases, system applications, and failed events.
- **Module Health** widget: Use the Module Health widget to evaluate the health of your module and identify potential problems. The widget provides a central place for health information about module topology, system components, system messaging engines, queues, data sources, and failed events.

Module Health interacts with the Module Browser widget. When you select a module from Module Browser, the Module Health widget refreshes to display health information for the selected module. The Module Health widget displays an overall status for the module. It also groups more status and health information for major areas of your module, organizing it into tabs. If a resource encounters a problem (for example, a queue that reaches capacity, or a node agent that is not running), a warning icon is displayed at the top of each affected tab.



Note

If administrative security is enabled for Business Space, you must also enable application security for Module Browser to work properly.



Human Task Management widget

The screenshot shows two overlapping windows from the IBM WebSphere Business Space interface. The top window is titled 'Task Definitions' and lists various task types: Approval, CreateApplication, Final Loan Review, Inquiry, Review, and To-do. The bottom window is titled 'Process Definitions' and lists process definitions: Account verification for %, Final Loan Review, and Underwriting. Two yellow circles with numbers 1 and 2 point to specific elements: circle 1 points to the 'Create' button in the Task Definitions window, and circle 2 points to the 'View' button in the Process Definitions window.

© Copyright IBM Corporation 2016

Figure 15-19. Human Task Management widget

WB8601.2

Notes:

The Human Task Management widgets provide business users and managers with capabilities to check and explore processes and human tasks. They can easily be combined with other widgets to compose powerful business spaces.

These widgets are delivered as part of Business Space, and they function in the same way as other widgets. You can minimize, maximize, and drag widgets while laying out a page. In addition, each widget has a menu that contains actions that you can take on the widget. The widget menu also has a help icon, which links to information about that widget.

1. Task Definitions widget: Task-related widgets allow business users to work with and manage tasks and escalations. The Task Definitions widget belongs to the group of task-related widgets. Use the Task Definitions widget as the starting point for creating tasks and working on tasks that are associated with specific task definitions.
2. Process Definitions widget: Process-related widgets allow business users to work with and manage processes. The Process Definitions widget belongs to the group of process-related widgets. Use the Process Definitions widget as the starting point for working on processes that are associated with specific process definitions.

The screenshot shows a 'Tasks' widget interface. At the top, there's a navigation bar with a blue arrow pointing right labeled 'WebSphere Education' and the IBM logo. Below the navigation bar, the title 'Tasks widget' is displayed. The main area is a table titled 'Tasks' with columns: Name, Owner, Status, and Actions. There are two rows of data:

Name	Owner	Status	Actions
CreateApplication		In progress	View the task.
Final Application Review		Available	

A yellow circle with the number '1' points to the 'View the task.' button for the 'CreateApplication' row. A mouse cursor is hovering over this button. A callout bubble with the text 'View the task.' is shown below the button. A yellow circle with the number '2' points to the 'Completed' status cell for the 'CreateApplication' row, which is highlighted with a red box. The status cell contains a green checkmark icon and the word 'Completed'.

© Copyright IBM Corporation 2016

Figure 15-20. Tasks widget

WB8601.2

Notes:

Use the Tasks widget to work with different types of tasks; for example, tasks that you own, or tasks that are available for you to work on.

1. You can click the task icon to view the task so you can work on it. Depending on the status, the status might be "In progress" or "Available" when you are ready to work on it.
2. When you are done with the task, the status changes to "Completed."

The screenshot shows the WebSphere Service Registry and Repository interface. At the top, there's a header with the IBM logo and the title "WebSphere Service Registry and Repository widgets". Below the header is the SOA Governance dashboard.

SOA Governance Dashboard:

- My Recent Items - Service Registry Collection:** A table listing recent items:

Name	Type	State	Last Modified
createAccount (1.0)	Service Operation	SLD Subscribable	Nov 3, 2011 6:37:37 PM
http://www.jke.com/Account (2.0) (2.0)	Schema	SLD Subscribable	Nov 3, 2011 6:37:36 PM
AccountCreation (1.0)	Service Interface	SLD Subscribable	Nov 3, 2011 6:37:36 PM
AccountCreationSoapBinding (1.0)	Service Binding	SLD Subscribable	Nov 3, 2011 6:37:27 PM
- Service Registry Navigator:** A graph showing the relationships between a service and its interface and port:


```

graph LR
    Service[AccountCreationService-Staging (1.0) Service] --> Interface[AccountCreation (1.0) Service Interface]
    Service --> Port[AccountCreationStagi... (1.0) Service Port]
      
```
- Service Registry Detail:** A detailed view of the "AccountCreationService-Staging (1.0)" service, including its properties, governance state, and relationships.

Figure 15-21. WebSphere Service Registry and Repository widgets

WB8601.2

Notes:

The slide depicts several widgets in WebSphere Service Registry and Repository. Before you can create such a space with widgets, WebSphere Service Registry and Repository must be installed and configured in the environment.



IBM Business Monitor widgets

KPIs

Average Order Fulfillment

0 m	2 m	4 m	6 m	8 m	10 m
0 s					

Percentage of Shipped Orders

0	20	40	60	80	100
0					

Alerts

	Subject	Date and Time
<input type="checkbox"/>	• Late Order Shipment	February 14, 2010 8:29:12 PM
<input type="checkbox"/>	• Late Order Shipment	February 14, 2010 8:29:12 PM
<input type="checkbox"/>	• Late Order Shipment	February 14, 2010 8:29:12 PM
<input type="checkbox"/>	• Late Order Shipment	February 14, 2010 8:29:12 PM
<input type="checkbox"/>	• MyNewAlert	February 14, 2010 8:29:11 PM
<input type="checkbox"/>	• Late Order Shipment	February 14, 2010 8:28:12 PM

© Copyright IBM Corporation 2016

Figure 15-22. IBM Business Monitor widgets

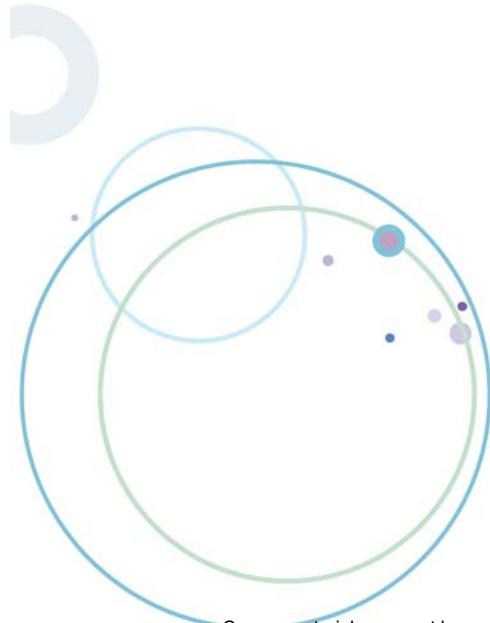
WB8601.2

Notes:

The slide depicts several widgets in IBM Business Monitor. Before you can create such a space with widgets, you must install and configure IBM Business Monitor in the environment.

15.4. Configure a new space

Configure a new space



© Copyright IBM Corporation 2016

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

10.1

Figure 15-23. Configure a new space

WB8601.2

Notes:

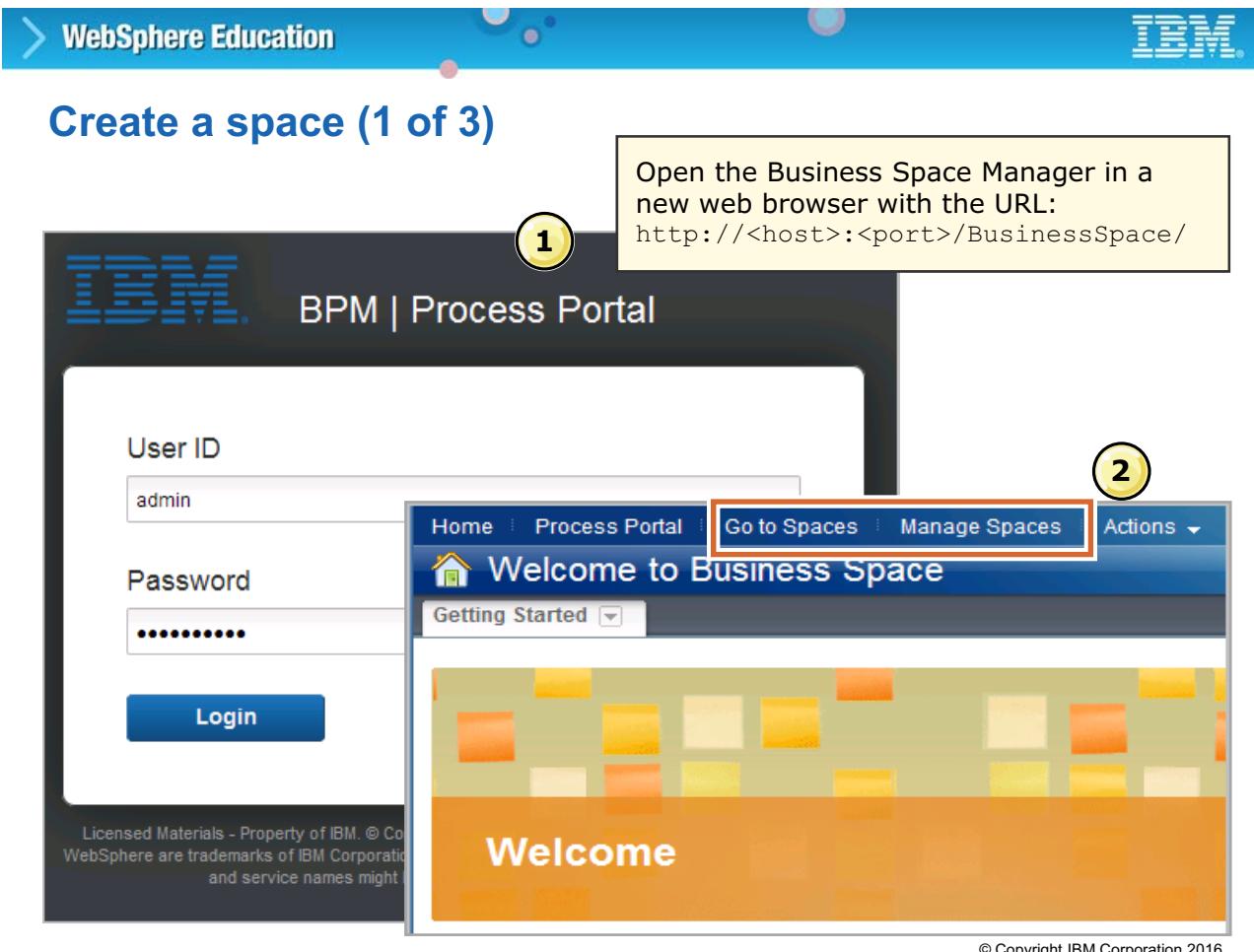


Figure 15-24. Create a space (1 of 3)

WB8601.2

Notes:

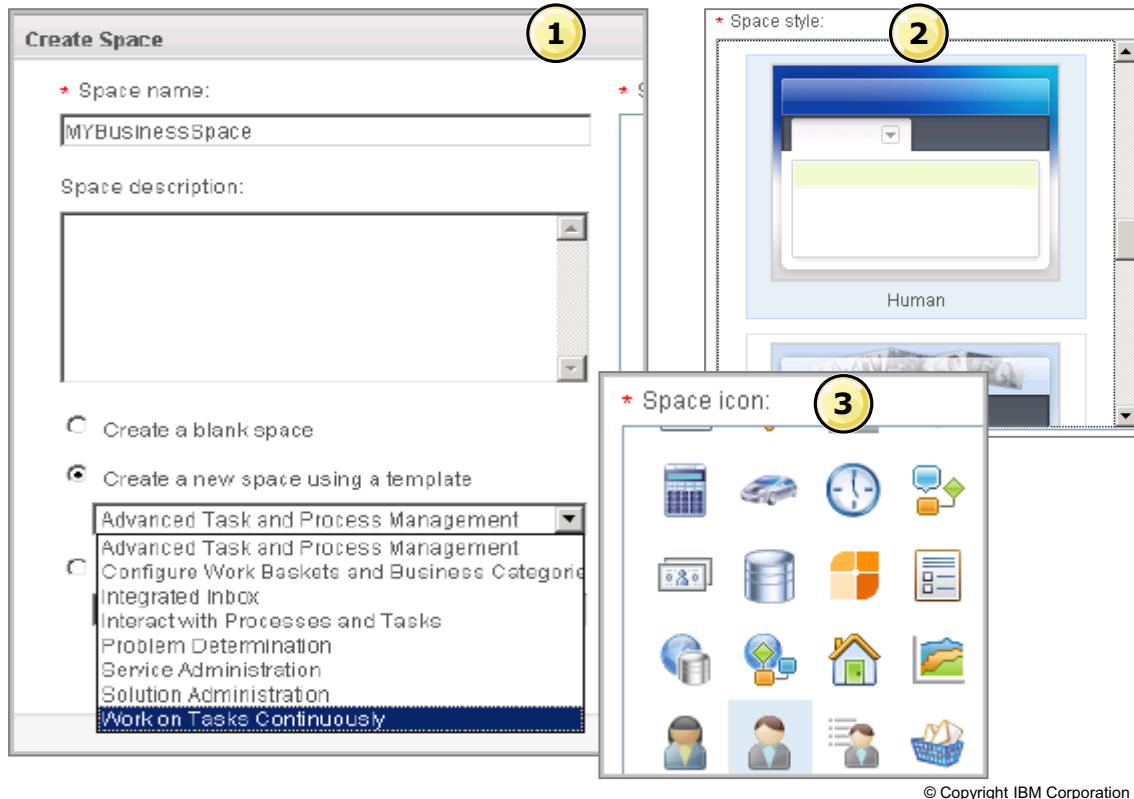
To start Business Space, you open a browser and enter `http://<host>:<port>/BusinessSpace/` in the URL field.

1. The login page for Business Space and Process Portal is the same.
2. After logging in, you can click **Go to Spaces** to go to a specific space or click **Manage Spaces** to create a new space.

WebSphere Education

IBM

Create a space (2 of 3)



© Copyright IBM Corporation 2016

Figure 15-25. Create a space (2 of 3)

WB8601.2

Notes:

A business space is a collection of related web content that provides you with insight into your business and the capability to react to changes in it. To contain a collection of related pages, you can create a business space.

1. In the Create Space window, you must provide a unique space name that describes your business space. When you create a space, the title and description that you provide serve as default values in all languages. You can then switch languages and replace those default values with the appropriate translations.

Select how you want to create the space:

- Create a blank space, which creates an empty space.
- Create a space by using a template, which creates a space that contains the pages and widgets that the template defines. Each template supports one or more scenarios by providing the widgets that you need for the scenarios and organizing the widgets on the pages.
- Duplicate an existing space, which creates a space that contains the same pages and widgets as an existing space.

2. Select a style to set the look for the pages in the new business space, and select an icon to represent the space. In Business Space running on IBM WebSphere Portal Server, the business space automatically has the default Business Space style and icon applied. You can select a different style only when you are editing a page, and you can select a different icon only when editing the space settings.
3. A list of icons is available for selection for your business space. This icon is displayed along with the title on the banner of the business space page.

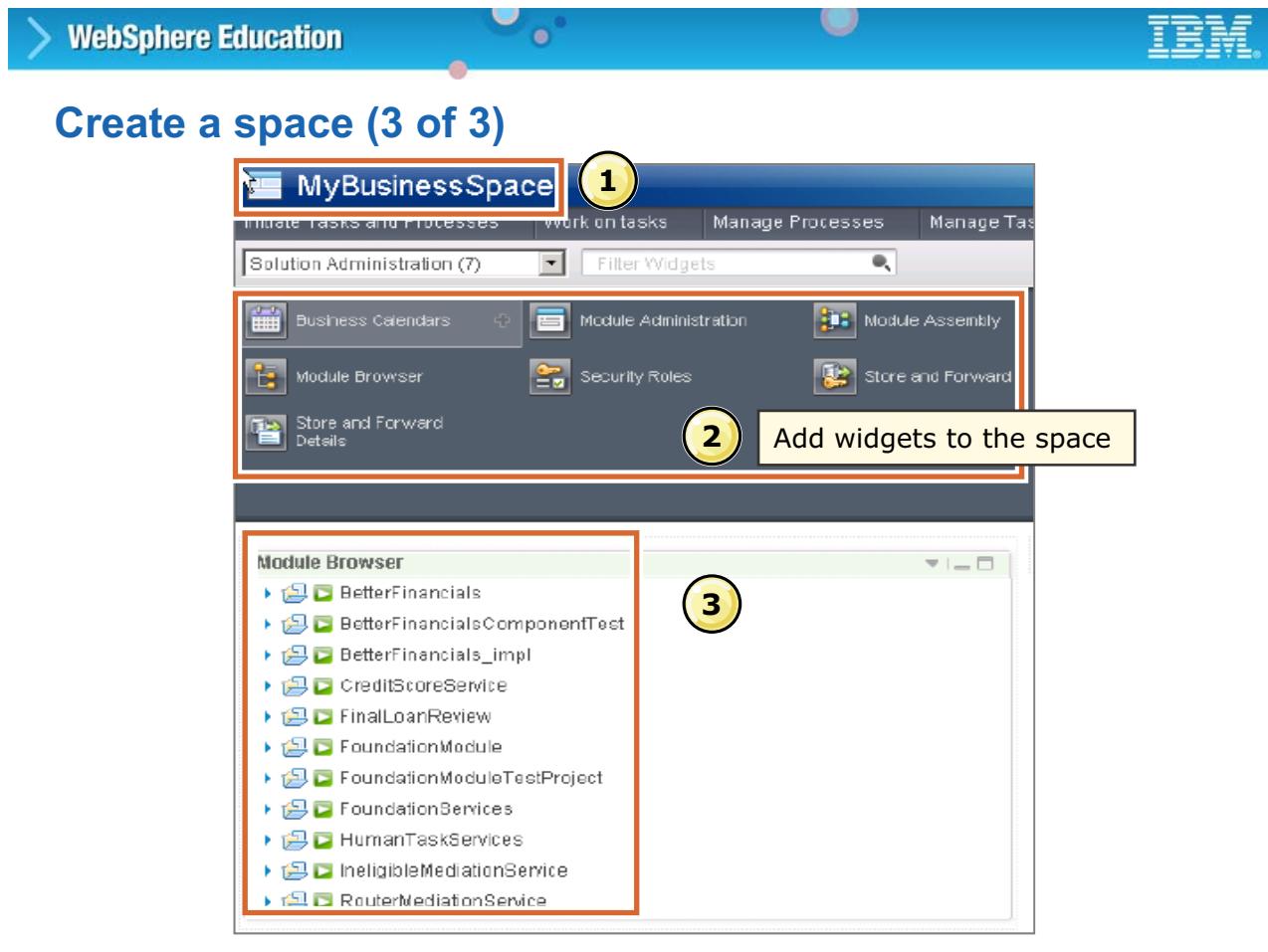


Figure 15-26. Create a space (3 of 3)

WB8601.2

Notes:

1. The business space title and icon are displayed in the banner area.
2. Select from the available widgets in the widget section and add to the page. You must click the **Edit Page** link before you can select and add the widgets.
3. The widget is displayed in a page. You can also add multiple related widgets on a single page.



Customizing a business space

- Login page
 - Make superficial changes to the appearance of the login page
- Space styles
 - Make superficial changes to a page style, which determines the color and appearance of a business space
- Banner
 - Modify the content, appearance, or behavior of the banner that is displayed at the top of each space
- Theme
 - Fully customize the structure and content of a page
 - Themes control the navigation, appearance, and layout of your space, including colors, fonts, and images that surround the widgets on the page

© Copyright IBM Corporation 2016

Figure 15-27. Customizing a business space

WB8601.2

Notes:

You can customize the behavior and the appearance of your business space in various ways, from small changes like modifying default link text to large-scale changes like customizing the entire theme.

Business Space supports the customization of these areas:

- **Login page:** You can make superficial changes to the appearance of the login page.
- **Space styles:** You can make superficial changes to a page style, which determines the color and appearance of a business space.
- **Banner:** You can modify the content, appearance, or behavior of the banner that is displayed at the top of each space.
- **Theme:** You can fully customize the structure and content of a page. Themes control the navigation, appearance, and layout of your space, including colors, fonts, and images that surround the widgets on the page.

Web-based Distributed Authoring and Versioning (WebDAV) is used to store and deploy the artifacts that are used for the login page, styles, banners, and themes. If you customize any of these artifacts, deploy them using WebDAV. WebDAV is a set of extensions to Hypertext Transfer

Protocol (HTTP) which allows users to cooperatively edit and manage files on Remote Web servers. Most operating systems provide built-in support for WebDAV.

Working with business spaces

- The level of access determines the level of control you have on the business space
- Create business spaces
- Control access to business spaces
- Change the style of business spaces
- Change the owner of a business space
- Switch from one business space to another
- Export business spaces
- Import business spaces
- Delete business spaces

© Copyright IBM Corporation 2016

Figure 15-28. Working with business spaces

WB8601.2

Notes:

Working with business spaces

Depending on your level of access, you can create business spaces, modify existing ones, and delete obsolete spaces.

- **Creating business spaces:** To contain a collection of related pages, you can create a business space.
- **Controlling access to business spaces:** If you are the owner of the business space, you can share it so that others can access it. As part of the process of sharing the space, you can set who is able to view and edit the business space.
- **Changing the style of business spaces:** If you can edit a business space, you can change its style so that it has a different color and appearance. The method that you use depends on whether you are using Business Space running on WebSphere Application Server or Business Space running on WebSphere Portal Server.
- **Changing the owner of a business space:** If you own a business space, you can transfer ownership of that business space to another person.

- **Switching from one business space to another:** When you are finished with a business space, you can switch to another business space. When you switch to another space, the first space closes automatically.
- **Exporting business spaces:** If you can edit a business space, you can export it. Exporting a space is useful if you want to give someone a copy of your space without giving them access to it or if you want to re-create the space later.
- **Importing business spaces:** You can import a business space and the pages that it contains. Importing is a way of creating a business space that is already populated with pages and configured widgets.
- **Deleting business spaces:** If you own a business space, you can delete it so that it and the pages it contains are no longer available.

Unit summary

- Describe the purpose and business value of Business Space
- Describe the types of widgets that are available in Business Space
- Describe the types of templates that are available in Business Space
- Explain how to create and configure a new space in Business Space

© Copyright IBM Corporation 2016

Figure 15-29. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. True or False: A page must be created for every widget that is added to it.
2. If only the templates specific to IBM Process Manager are available in Business Space, what can be done to add more templates?
3. True or False: Multiple pages can be added to a business space.

© Copyright IBM Corporation 2016

Figure 15-30. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.

Checkpoint answers

1. False. You can add multiple widgets with different capabilities on the same page. You can also combine widgets to interact with each other to work on related tasks. Combined widgets are called a mashup.
2. Cross-product templates are templates that contain widgets from multiple products. The administrator must first import the templates and make them available before users can create spaces that are based on these templates.
3. True. Multiple pages can be added. However, if you have too many pages in a space, it affects performance. The maximum is 10 pages. Try to limit the pages in a space to the ones that are necessary for the space.

© Copyright IBM Corporation 2016

Figure 15-31. Checkpoint answers

WB8601.2

Notes:

Exercise 13

Exploring Business Space



© Copyright IBM Corporation 2016

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

10.1

Figure 15-32. Exercise 13

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create a space by using the Business Space client
- Use the Business Space client to work with human tasks
- Create a page and display content by using widgets and Business Space templates

© Copyright IBM Corporation 2016

Figure 15-33. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 13

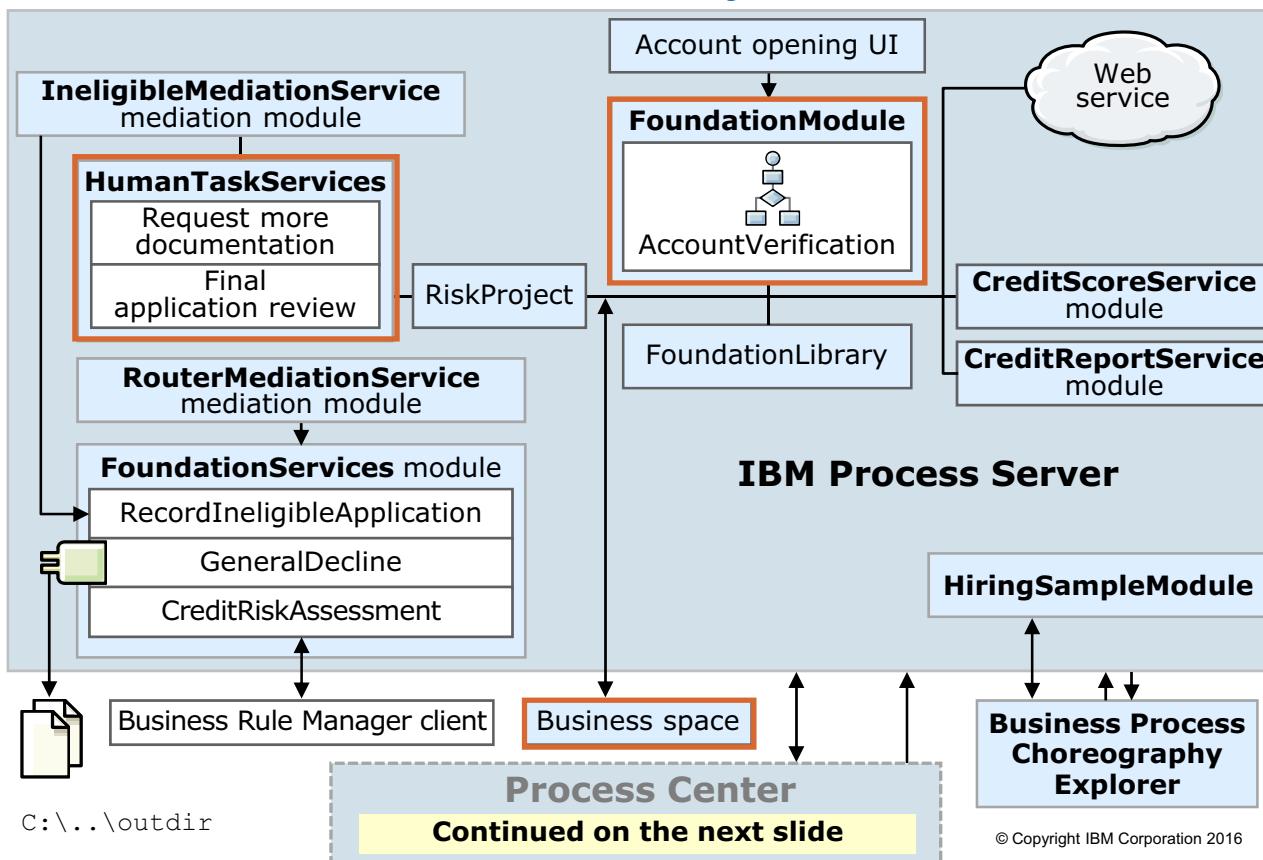
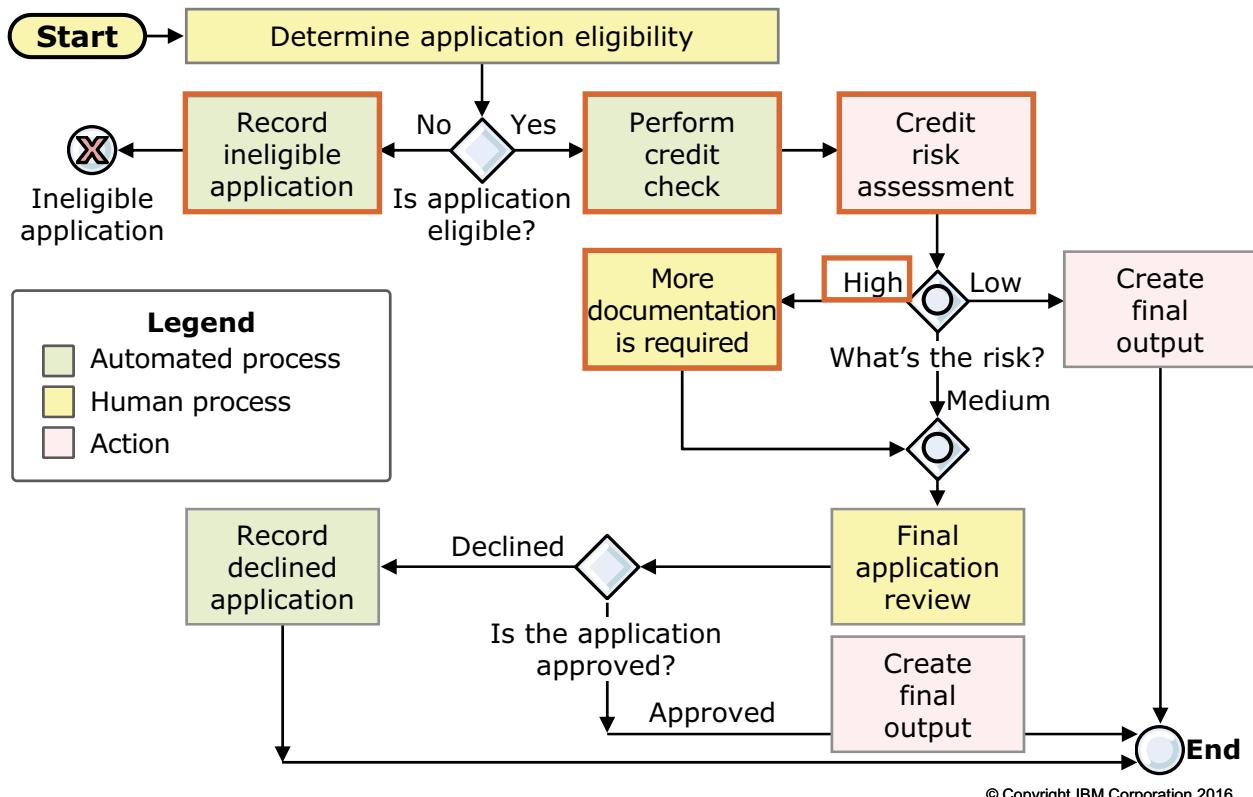


Figure 15-34. Course business scenario: What you work with in Exercise 13

WB8601.2

Notes:

Account verification process: Overview



© Copyright IBM Corporation 2016

Figure 15-35. Account verification process: Overview

WB8601.2

Notes:

Business Space is a browser-based graphical user interface that you use to view and interact with content from various products in the business process management portfolio. Business Space provides a single web-based point of access for the content, and you use Business Space to combine the content in useful and interesting ways. These combinations give you insight into your business and the capability to react to changes in it.

In this exercise, the entire “create account” solution is assembled, and you can run end-to-end tests of the applications. Different possible paths are available through the AccountVerification business process. You run a test to verify a test case to work with the Business Space client.

When you use company name ACME to test the applications, the eligibleApplication attribute is set to true and the creditRisk evaluates to MED. The creditScore returned is 6. When you use companyName ACME to submit an application, the application flows through these activities:
Account Verification Receive > Determine Application Eligibility > Map to Credit Check > Credit Check Service > Map Credit Checking Result > Credit Risk Assessment > Assign Variable > While More Documents Required > Request More Documentation.

A user interface for Request More Documentation is used to change the comment field from None to Complete. After leaving the While More Documents Required loop, the application flows through

Merge Assign > Create Output > Final Application Review. A user interface for Final Application Review is used to update the applicationDecision field. If applicationDecision is true, the application flows to **Create Output > Account Verification Reply**. If applicationDecision is false, the application flows through **Generate Decline > Record Declined Application > Account Verification Reply**.

The screenshot shows the WebSphere Education interface. At the top, there's a blue header bar with the "WebSphere Education" logo on the left and the "IBM" logo on the right. Below the header, the main title "Create business space in Exercise 13" is displayed in a large blue font. The interface is divided into several sections:

- Tasks:** A grid view showing tasks. One task, "Final Application Review", is highlighted with an orange border. It has a status of "Available" and a priority of "Very high".
- Process Information:** A column on the right containing various actions like Edit, Accept, View, Transfer, Return, Delete, Escalate, and Postpone. The "Accept" option is currently selected.
- MyBusinessSpace:** A navigation bar with links for Home, Process Portal, Go to Spaces, Manage Spaces, Actions, and a dropdown menu for "Initiate Tasks and Processes".
- Solution Administration:** A sidebar with a list of categories: All Widgets (42), Business Configuration (1), Human Task Management (13), Problem Determination (2), Solution Administration (7) (which is currently selected), Solution Operation (5), User Management (1), Viewers (1), Widgets - deprecated (10), and Wiring (2). Below this are links for Human Workflow Diagram and Administration.
- Widgets:** A main area displaying various business-related widgets such as Business Calendars, Business Categories, Create Tasks, Escalations, Google Gadgets, Mediation Policy Administration, and Module Administration.

Figure 15-36. Create business space in Exercise 13

WB8601.2

Notes:

Unit 16. Advanced testing

What this unit is about

This unit describes the testing features that are available in IBM Integration Designer.

What you should be able to do

After completing this unit, you should be able to:

- Describe the advanced testing facilities that are available in IBM Integration Designer, including the Component Test Explorer and cross-component trace
- Describe the integration debugger
- Define the purpose and function of the serviceDeploy tool
- Describe how to use serviceDeploy in single-developer and multiple-developer environments

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Describe the advanced testing facilities that are available in IBM Integration Designer, including the Component Test Explorer and cross-component trace
- Describe the integration debugger
- Define the purpose and function of the serviceDeploy tool
- Describe how to use serviceDeploy in single-developer and multiple-developer environments

© Copyright IBM Corporation 2016

Figure 16-1. Unit objectives

WB8601.2

Notes:



Topics

- Advanced testing
- Deploying integration applications with serviceDeploy

© Copyright IBM Corporation 2016

Figure 16-2. Topics

WB8601.2

Notes:

16.1. Advanced testing

Advanced testing



© Copyright IBM Corporation 2016

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

10.1

Figure 16-3. Advanced testing

WB8601.2

Notes:

Testing modules

- You can test your modules in two ways: unit testing and component testing
- Unit testing uses the integration test client to unit test the modules
 - Testing is done on interface operations of your components
 - One at a time
 - Determines whether the components are correctly implemented and references are correctly wired
- Component testing uses the test suite editor to create test suites and test cases
 - Sequentially test multiple operations as a group in the integration test client
 - Can do batch component testing
 - Use the web-based component test explorer

© Copyright IBM Corporation 2016

Figure 16-4. Testing modules

WB8601.2

Notes:

In IBM Integration Designer, you can test your modules in two ways. In unit testing, you choose components and interfaces, and then you test the operations one at a time in the integration test client. In component testing, you use the new test suite editor to create test suites and test cases.

In IBM Integration Designer, you can use the integration test client to unit test your modules. The testing is generally done on the interface operations of your components, which you can use to determine whether the components are correctly implemented and the references are correctly wired.

By using the test suite editor and associated wizards, you can create and define test cases that consist of one or more operations. You can sequentially test multiple operations as a group in the integration test client. You can also do batch component testing on either a test environment server or a stand-alone server by using test scripts or the user interface of the web-based Component Test Explorer.

Unit testing: Integration test client

- Use the integration test client to test any of the following items:
 - An individual module
 - A set of interacting modules
 - An individual component
 - A set of interacting components
- Integration test client is integrated with the assembly editor and Business Integration view
 - Open the test client from the assembly editor or open the assembly editor from the test client
 - Open the test client from the Business Integration view
 - Can open multiple instances of the test client and use them to do simultaneous testing

© Copyright IBM Corporation 2016

Figure 16-5. Unit testing: Integration test client

WB8601.2

Notes:

In IBM Integration Designer, you can use the integration test client to unit test your modules. The testing is generally done on the interface operations of your components, which you can use to determine whether the components are correctly implemented and the references are correctly wired.

Use the integration test client to test any of the following items:

- An individual module
- A set of interacting modules, including modules that are part of a process application or toolkit
- An individual component
- A set of interacting components

The integration test client is fully integrated into the workbench so that you can go through the Business Integration view and other views while you are using the test client. It is also closely integrated with the assembly editor. You can open the test client from the assembly editor, and you can open the assembly editor from the test client. Although the assembly editor is considered the primary starting point for the integration test client, you can also open the test client from the Business Integration view. Regardless of whether you open the integration test client from the

assembly editor or the Business Integration view, you can open multiple instances of the test client and use them to do simultaneous testing.



Integration test client: Events tab

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, and then click the Continue icon in the Events area to run the test. [More...](#)

Configuration: Default Module Test

Module: FoundationModule

Component: AccountVerificationExport

Interface: AccountVerification

Operation: InputCriterion

Using the **Events** tab, you can do numerous test activities to interact with your module during testing

© Copyright IBM Corporation 2016

Figure 16-6. Integration test client: Events tab

WB8601.2

Notes:

In IBM Integration Designer, the integration test client is the designated tool for testing modules and components. In the test client, you can manage and precisely control your tests.

The test client interface has two main features:

- Events page
- Configurations page

Almost all of the tasks that you can do in the integration test client are either initiated or completed in one of the two pages. The Events and Configurations pages are described in the topics “Events page” and “Configurations page.”

In the Events page of the integration test client, you can do numerous test activities to interact with your module during testing. You can select an operation to test, specify values for the operation, and invoke the operation. The image depicts the following items:

1. Module name that runs the test client
2. Control area that has the following icons:

- Continue: Initiates the invocation. Depending on the current deployment state of your module, the Deployment Location wizard might open so that you can select a test server.
 - Stop: Detaches the integration test client from the server. All running operations that are waiting on user input from the test client are terminated. All other running applications continue until they terminate, but no status is reported in the test client.
 - Invoke: Generates an Invoke event in the Events area, which you can use to select an operation, specify values for the operation, and invoke and test the operation.
 - Data pool: In the Detailed Properties area of the Events page, any values that you specify for an operation, manual emulation, or event definition in the value editor can be saved to the data pool. The data pool icon opens the data pool editor, which you can use to view, edit, select, and use the saved data pool values.
 - Filters: Opens the Event Filter window, from which you can choose an event filter that suppresses the display of selected event types.
 - Save: Opens the Save Test Trace window, with which you can save the test trace.
3. The Events area displays a hierarchical test trace for the events that are generated during a test. These events are typically grouped under a top-level event. For example, if you are testing an operation, the events are nested together under the Invoke event that was used in the invocation of the operation.
 4. The General Properties area shows the time that an event that is selected in the Events area was generated.
 5. The Detailed Properties area displays the specific properties of any event that is selected in the Events area. For some events, the specific properties can include information about the resources that are associated with the selected event, such as the name of the test configuration and module that is used in the test. If you are testing a module that is contained in a process application or a toolkit, the name of the process application or toolkit is added to the name of the module as a prefix.



Integration test client: Configurations tab

Integration Test Client [FoundationModule_Test]

Configurations

This area displays test configurations and their resources. Select a test configuration or resource to display its properties in the General Properties and Detailed Properties sections. [More...](#)

1

2

3

4

5

Using the **Configurations** tab, you can edit test configurations, and add modules, emulators, and monitors for your test

© Copyright IBM Corporation 2016

Figure 16-7. Integration test client: Configurations tab

WB8601.2

Notes:

Configurations page of the integration test client

In the Configurations page of the integration test client, you can edit the default test configuration or you can create and edit new test configurations. You can add modules to your test configurations, or add emulators and monitors to your test configuration modules, to more precisely control your tests.

1. The control area has the following icons:

- Add: Opens the New Configuration wizard, with which you can add one of the following test configuration elements to the Configurations page:
 - Test bucket configuration
 - Test configuration
 - Test case
 - Process application or toolkit
 - Module

- Emulator
 - Monitor
2. The Configurations area provides a tree view of your test configurations and test bucket configurations.
 - For any specified test configuration, the following information is shown:
 - The name of the test configuration
 - The name and version of the process application or toolkit
 - The names of the modules in the test configuration
 - The names of any emulators in each test configuration module
 - The names of any monitors in each test configuration module
 - For any specified test bucket configuration, one or more of the following items are shown:
 - The name of the test bucket configuration
 - The name of the test suite in the test bucket configuration
 - The names of the test cases in the test suite
 - The name of the test configuration in the test suite
 - The name and version of the process application or toolkit
 - The names of the modules in the test configuration
 - The names of any emulators in each test configuration module
 - The names of any monitors in each test configuration module
 3. The module name that is running the test client.
 4. The General Properties area provides the name and description (if any) of any test configuration element selected in the Configurations area.
 5. The Detailed Properties area displays the specific properties of any test configuration element that is selected in the Configurations area. For example, if a test configuration is selected in the Configurations area, the Detailed Properties area displays the names of any modules that are contained in the test configuration.

Component testing

- In component testing, you use the associated wizards to create and define test cases that comprise multiple operations
- Component test projects:
 - Provide a way to automate running test cases
 - Provide testing for components in integration (SCA) modules
 - Created as SCA modules
 - Results are displayed in the integration test client
 - Include test suites, cases, and configurations
 - Can be deployed and run on the server
- Test suites: Containers for test cases
 - You create test cases to test individual operations or groups of operations
- Test cases: Containers for operations
 - Operation testing is automated by using predefined input and output variables
 - Test cases can be authored manually or by using the integration test client execution trace
- Test configurations: Used to control tests
 - Test configurations specify required emulators and monitors

© Copyright IBM Corporation 2016

Figure 16-8. Component testing

WB8601.2

Notes:

In component testing, you use the test suite editor and associated wizards to create and define test cases that comprise multiple operations. By using the test suite editor, you can automate and simultaneously test the operations in the integration 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 invoked 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.

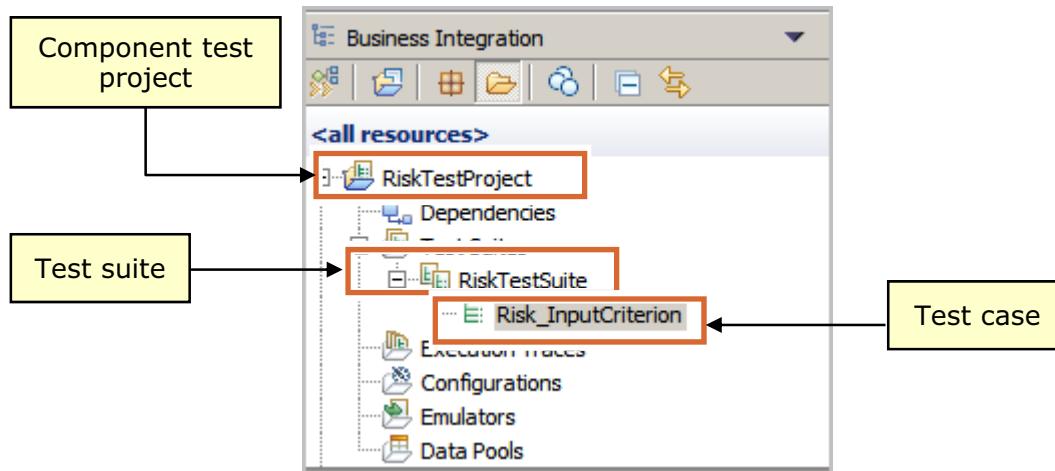
- **Component test projects:** Generally, creating a component test project to contain your test suites is the first step in preparing to do component testing.
- **Test suites:** After you create a component test project, you generally create a test suite to contain your test cases. When you create a test suite, you can choose from one of the following test patterns:
 - **Operation-level testing:** In operation-level testing, a separate test case is created for each operation that you select for component testing.

- **Scenario-based 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 use test cases to automate and simultaneously test the operations in the integration 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 of these modules can include zero or more emulators for components or references in the module and 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 is designed to closely resemble the test client, which helps you easily 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 that comprise multiple operations. You can use the test suite editor 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, and then you 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 that simplify 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.

Component test project

- Component test projects provide a way to automate running test cases
 - Component tests automate and test operations in the integration test client



© Copyright IBM Corporation 2016

Figure 16-9. Component test project

WB8601.2

Notes:

Test suites

A test suite is a collection of one or more test cases. When you create a test suite, you can choose from one of the following test patterns:

- Operation-level testing
- Scenario-based testing

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 being a container for multiple operations that you select for testing. Test cases enable automation and testing of the operations in the integration 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 variations

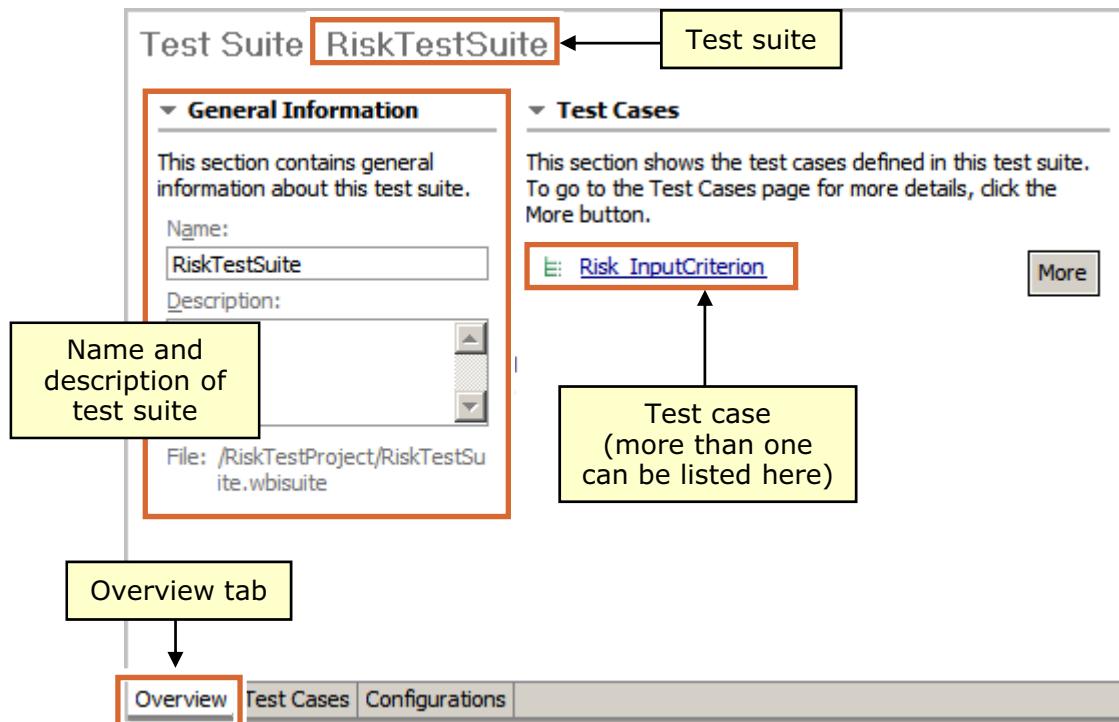
A test variation is a specific set of variable values for a test case. Although each test case is automatically assigned a default test variation, you can create multiple test variations for a test case that contain a different set of variable values. When a test case is run, all of the test variations for the test case are run unless one of the test variations fails.

Test bucket configurations

A test bucket configuration is a set of specific test suites and test cases that are run together in the same test session.



Test suite: Overview tab



© Copyright IBM Corporation 2016

Figure 16-10. Test suite: Overview tab

WB8601.2

Notes:

In the Overview page, you can quickly view the essential information for the test suite that is open in the test suite editor.

General Information area

The General Information area displays the name of the test suite, which you can rename by refactoring. It also contains an editable description for the test suite and the path and name of the test suite file.

Test Cases area

The Test Cases area displays the test cases that are defined in the test suite. It also contains a More button that opens the Test Cases page.

Test suite: Test Cases tab

Test Suite: RiskTestSuite

Test Cases

This area displays test cases and their invocations and steps. Select a test case, invocation, or step to display its properties in the General Properties and Detailed Properties sections. [More...](#)

Test case (more than one can be listed here)

Risk_InputCriterion

General Properties

This section displays the name and description of the test case selected in the Test Cases area.

Name: Risk_InputCriterion

Description:

Detailed Properties

This section displays the links to the test configuration and test variations of the test case that is selected in the Test Cases area. [More...](#)

Test configuration: RiskTestSuite

-Test variations-

Default
AbCo

Test variations variable values for a test case

© Copyright IBM Corporation 2016

Figure 16-11. Test suite: Test Cases tab

WB8601.2

Notes:

In IBM Integration Designer, the test suite editor is the designated tool for editing test suites. It features a rich user interface that enables easy management of your test suites and test cases. The test suite editor is designed to closely resemble the test client, which helps you easily transition from using one tool to the other.

In the Test Cases tab of the test suite editor, you can do numerous activities to prepare your test suite and test cases for testing. For example, you can select an operation invocation for which you want to specify values in the test data table.

Test Cases area

The Test Cases area displays the test cases, the associated invocation steps, and Wait On steps for the operations that are defined in the test suite.

Control area

The control area provides the following icons to help you manage test cases and operation invocations:

- New test case: Adds an operation invocation for the test case that is selected in the Test Cases area

- Remove: Removes selected test cases or operation invocations from the Test Cases area
- Run test case: Runs the test in the integration test client
- Move Up: Moves up selected test cases or operation invocations in the Test Cases area
- Move Down: Moves selected test cases or operation invocations down in the Test Cases area
- Show Descriptions: Toggles selected test cases or operation invocations between the default description and the user-defined description

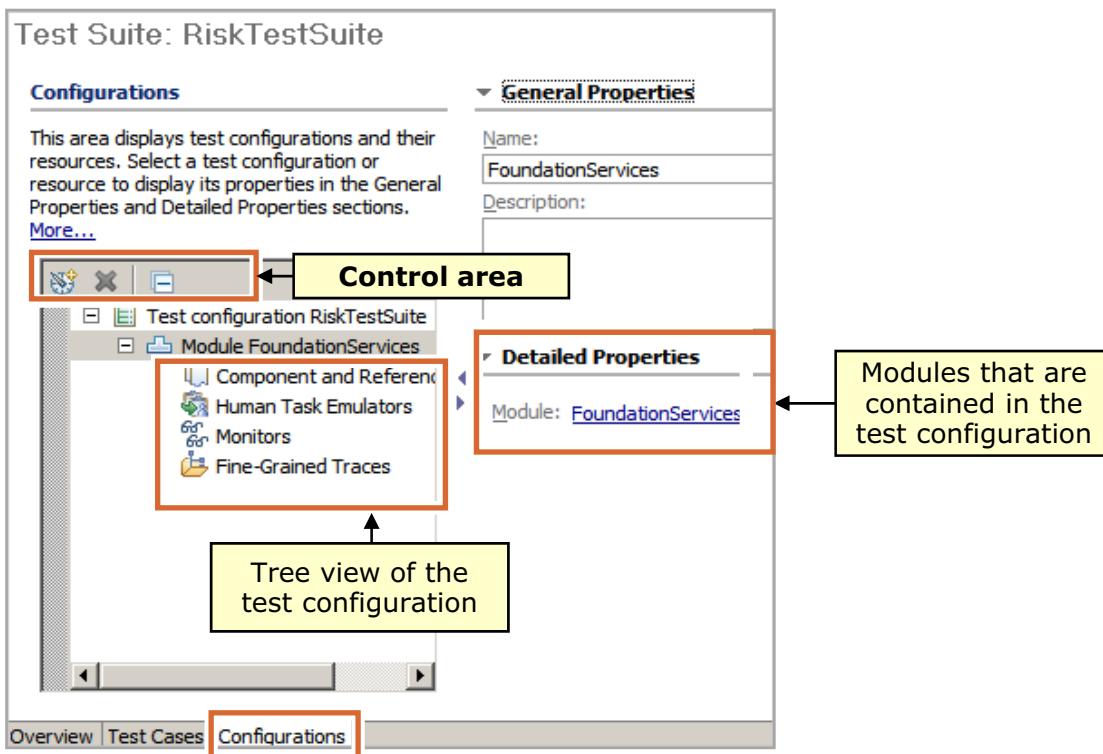
General Information

The General Information area shows name and description for a selected test case or the description for a selected operation invocation.

Detailed Properties

The Detailed Properties area displays the specific properties of any test case or operation invocation that is selected in the Events area.

Test suite: Configurations tab



© Copyright IBM Corporation 2016

Figure 16-12. Test suite: Configurations tab

WB8601.2

Notes:

In the Configurations tab of the test suite editor, you can edit the default test configuration. Using this Configuration tab, you can add modules to your test configuration, or add emulators and monitors to your test configuration modules, to more precisely control your tests.

The Configurations area provides a tree view of your default test configuration and displays the following information:

- The name of the test configuration
- The names of the modules in the test configuration
- The names of any emulators in each test configuration module
- The names of any monitors in each test configuration module

The control area provides the following icons to help you manage your default test configuration:

- Add: Opens the New Configuration wizard to add one of the following test configuration elements to the Configurations page:
 - Module

- Emulator
- Monitor
- Remove: Removes selected test configuration elements from the Configurations page

The Detailed Properties area displays the specific properties of any test configuration element that is selected in the Configurations area.

Running a test case

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

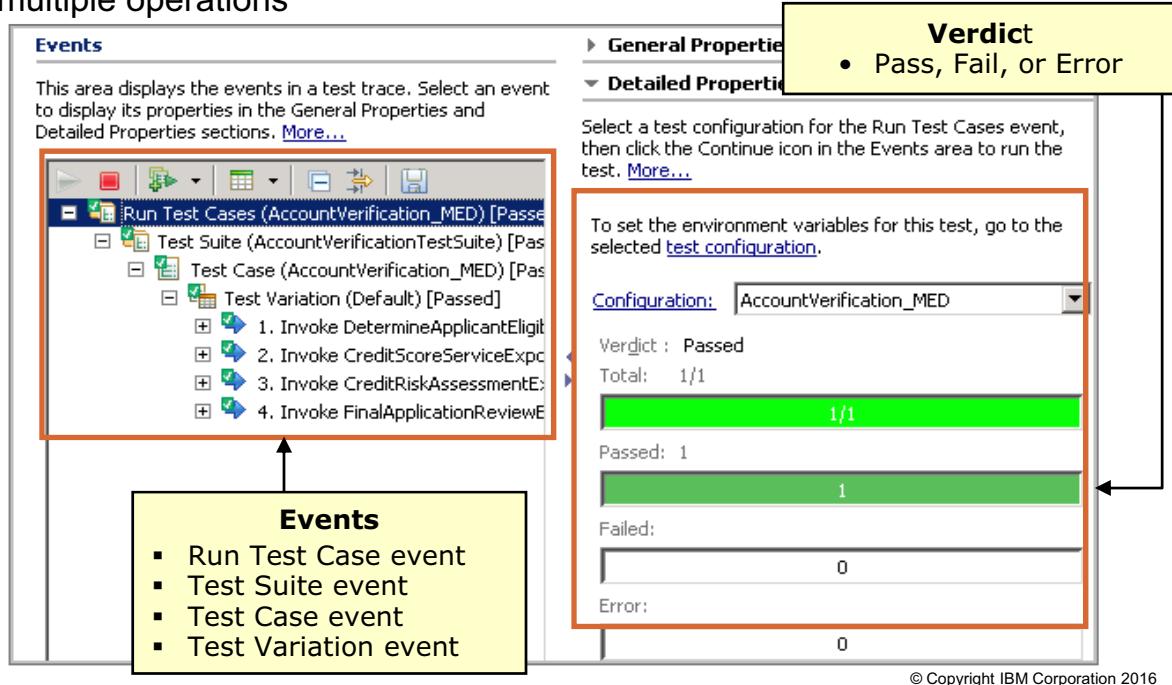


Figure 16-13. Running a test case

WB8601.2

Notes:

In addition to the standard events that are generated in the integration test client when you are unit testing, such as Invoke and Return events, component testing adds the following events in the integration test client:

- Run Test** event: An informational event that is generated when you select the **Run Test** menu item from either the integration test client or the Business Integration view. The event informs you whether the component test passed or failed. It also presents statistics on the total number of test cases that were run and the number that passed, failed, or were flagged with an error.
- Test Suite** event: An informational event that informs you whether the test suite passed or failed the test run. It also presents statistics on the total number of test cases that were run and the number that passed, failed, or were flagged with an error.
- Test Case** event: An informational event that informs you whether the test case passed or failed the test run.
- Test Variation** event: An informational event that informs you whether the test variation passed or failed the test run.

WebSphere Education 

Cross-component trace (1 of 2)

- Cross-component trace provides server logs and SCA trace in one view
- Log information is correlated to the SCA flow
 - Across multiple synchronous or asynchronous instances and processes
 - Multiple log files and multiple threads
- Cross-component trace shows the execution path of all SCA components even if the source code is not in the current workspace
 - Invocations, the error or exception and which SCA component they are related to, and a snapshot of the input/output data that is passed and returned

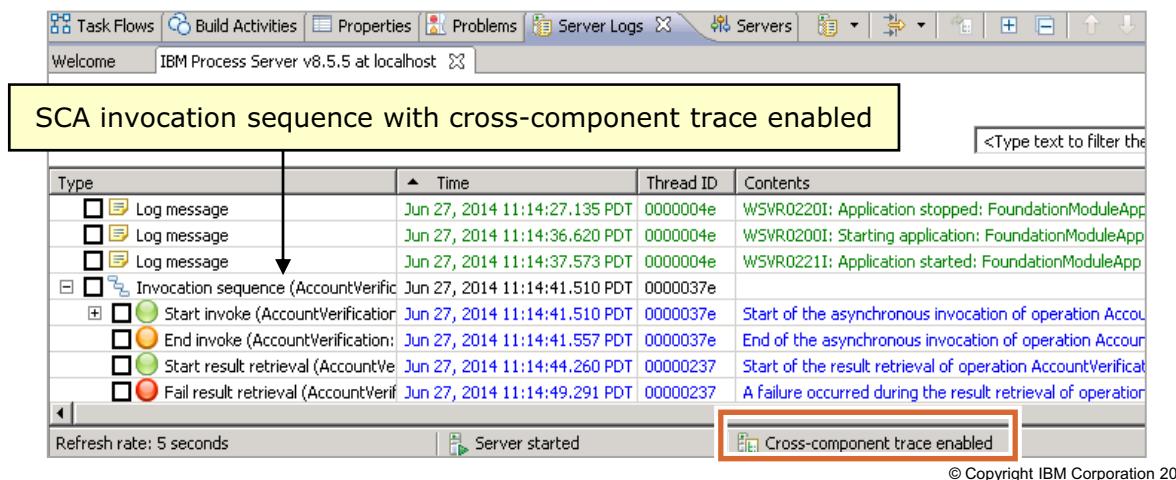


Figure 16-14. Cross-component trace (1 of 2)

WB8601.2

Notes:

By default, the Server Log view displays standard server console and log records. However, if you enable cross-component tracing, the Server Log view also displays invocation records that can contain the invocation data that is passed between the components in your application. The invocation records are displayed in hierarchical format in the Server Log view. You can more easily understand the relationships that exist between the records. When you enable cross-component tracing, the Server Log view becomes an even more powerful tool for problem determination.

When you enable cross-component tracing on a server, invocation records are generated during SCA processing of modules and components. The invocation records include information about any errors or events that occurred during processing, such as runtime exceptions. If you choose to enable cross-component tracing with the data snapshot feature, the generated invocation records also contain the invocation input and output data that is passed between the components during processing.

You can enable or disable cross-component tracing for a server from either the Server Log view or the server administrative console. If you enable cross-component tracing from the Server Log view, the tracing is enabled only during the server session. When you next stop or restart the server, the cross-component trace state is automatically disabled by default. By comparison, if you enable

cross-component tracing for a server from the server administration console, the cross-component tracing remains enabled for all sessions of the server until you choose to disable it again.

When you enable or disable cross-component tracing, you can choose from one of the following options:

- **Disabled:** This option disables cross-component tracing. No invocation records are generated in the server console and logs.
- **Enabled:** This option enables cross-component tracing. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, but the record properties do not include any invocation input and output data. The `SystemOut.log` and `trace.log` files are in the Server Log directory.
- **Enabled with data snapshot:** This option enables cross-component tracing with the data snapshot feature. Invocation records are generated in both the server console and the `SystemOut.log` and `trace.log` files, and the record properties include invocation input and output data. This data is captured in input and output files under the `logs\xfa` folder.



Cross-component trace (2 of 2)

Cross-Component Trace > server1

Set Cross-Component Trace configuration properties, including the level at which trace operations occur and properties that Cross-Component Trace records are written.

Configuration Runtime

General Properties

Trace output

Enable Cross-Component Trace
Save Cross-Component Trace output to:
trace

Server Settings

Trace all
 Enable data snapshot on this server

Module Settings

© Copyright IBM Corporation 2016

Figure 16-15. Cross-component trace (2 of 2)

WB8601.2

Notes:

Cross-component tracing allows identification of the `trace.log` data that is associated with IBM Business Process Manager modules and components. You can enable cross-component tracing on a server or on specific modules. The input and output data that is passed between Business Process Manager components can also be captured and can be used for problem determination by using IBM Integration Designer.

Configuration parameters are used for cross-component trace display on the Configuration tab.

When the server is restarted, the system applies the parameters that you set on the Configuration tab.

The console panel contains the following sections:

- **Trace output**
 - **Enable Cross-Component Trace:** Selecting **Enable Cross-Component Trace** prepares the server for the following options:
 - Cross-component trace for inbound application-specific call chains

- Enabling cross-component trace on any module that is selected under Enable tracing for the selected Service Component Architecture (SCA) modules
- **Save Cross-Component Trace output to:** Choose which file holds the data that the cross-component trace operations gather.
- **Server settings**
 - **Trace all:** Select this option to turn on cross-component trace for the creation of call chain information for all SCA modules in the server. Even with Trace all selected, you can add more SCA modules to the table of modules under Enable tracing for the selected Service Component Architecture (SCA) modules.
 - **Enable data snapshot on this server:** Select this option to enable the data snapshot feature of cross-component trace. When data snapshot is enabled, the system captures data that is sent in and passed between SCA components. This extra data (about what is passed between SCA components) can be large and is kept in separate files and not in the trace.log or systemout.log file.
- **Module settings**

Enable tracing for the selected Service Component Architecture (SCA) modules. This table provides a list of modules for which cross-component trace is enabled.

Fine-grained trace

- Fine-grained trace combines the test client and debugger to overlay the editors with a visualization to track steps in the path of events
 - Quickly visualize the path that is taken for a single execution of a component
 - Split screen feature while testing highlights activities as you pass through
- Fine-grained trace can be used in BPEL, state machines, and mediations and is enabled by default
 - Appropriate editors are auto-opened by clicking the event
 - The path of events is highlighted in the business process editor, mediation flow editor, and state machine editor

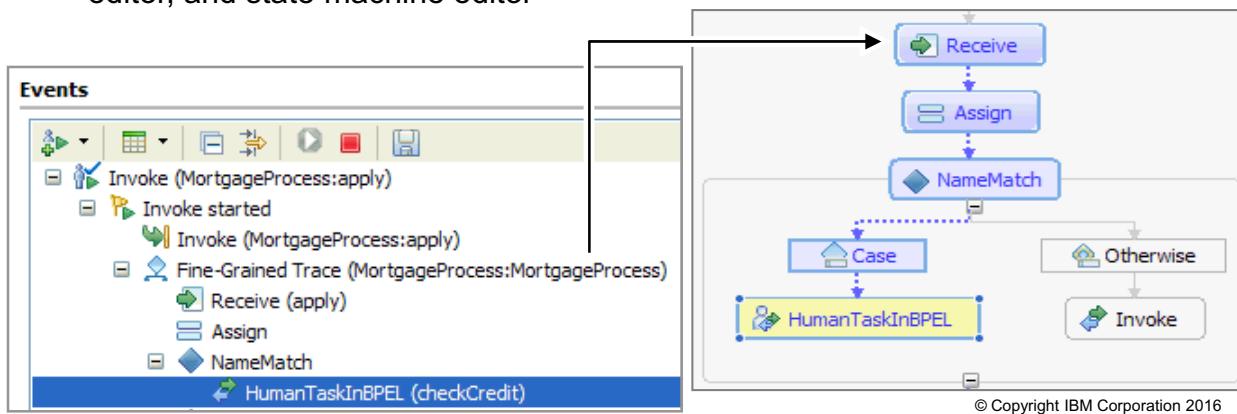


Figure 16-16. Fine-grained trace

WB8601.2

Notes:

When you are testing business processes, state machines, or mediation flows in the integration test client, you can choose to run your tests either with or without fine-grained trace. If you run your tests with fine-grained trace, the events area of the test client is populated with more events. These events correspond to the elements encountered in the execution path of the component that is being tested. If the associated component editor is open, the execution path is traced and highlighted in the component editor so you can easily see the specific path that was taken and tested.

Although fine-grained trace is enabled by default, you can set a preference to disable it, as described in the topic “Enabling or disabling fine-grained trace” in the product documentation. If you disable fine-grained trace and you later encounter an exception during your testing, you can enable fine-grained trace again to help you determine the source of the problem. When you test with fine-grained trace and an exception is encountered, the events area of the test client displays an event. This event corresponds to the last element encountered in the component execution path before the exception was thrown.

Command-line test invocation

- Integration test client includes command-line invocation
- Automates tests by using Ant scripts
 - Schedule the tests during a low usage time
 - Result is an XML file that describes the results
 - Batch files (for Windows O/S) and shell scripts (for Linux O/S) are provided to run the Ant scripts
- Servlet that the test client calls to run test cases remotely in test project
 - Generated in a side Java EE project along with modules
 - URL for the servlet is in the form:
`http://<host>:<port>/<testproject>Web/TestServlet`
- Ant has core support for CVS
 - Test projects and modules can be extracted from CVS, built, deployed, tested, and removed
 - User must know CVS commands

© Copyright IBM Corporation 2016

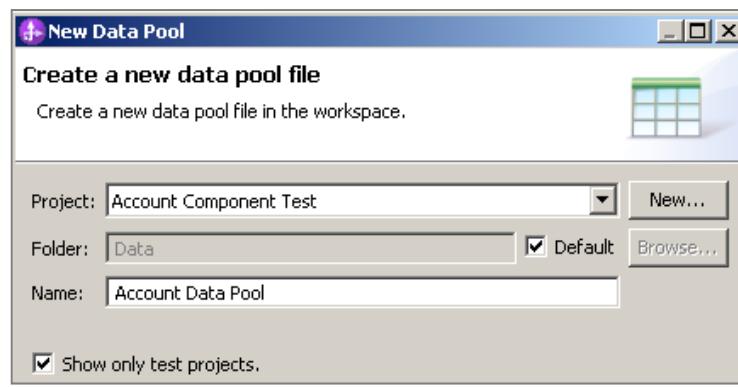
Figure 16-17. Command-line test invocation

WB8601.2

Notes:

Stand-alone data pool

- The integration test client has a stand-alone data pool wizard
- The data pool:
 - Is saved in your workspace
 - Can be exported with a project
 - Can be checked into source control
- By using the data pool editor:
 - You can open multiple data pools
 - You can copy values between pools



© Copyright IBM Corporation 2016

Figure 16-18. Stand-alone data pool

WB8601.2

Notes:

On the events page of the integration test client, a value editor is provided. By using this editor, you can specify, view, edit, and pass values for operations, manual emulations, and event definitions. You can also save values to a data pool, where you can use the data pool editor to view and edit the values, and later reuse them in the value editor.

In the integration test client, you can use one or more data pools. You can save values to the data pools, edit values in the data pool editor, and reuse values from the data pools. You can more easily manage the input values for your operations and output values for your manual emulations. When you create a data pool, it is added to a test project.



Component Test Explorer (1 of 2)

Component Test Explorer

Test Cases Human Task Emulators Component Emulators Scheduler Help Logout

Properties

Path: FoundationModuleTestProject->AccountVerificationTest

Description: (empty)

Environment Variables

Name	Value
------	-------

Run Log

Start Time	Path	Result
2014-06-26 14:35	FoundationModuleTestProject->AccountVerificationTestSuite	pass

Execution history

All test cases on the server

By using the Component Test Explorer web application, you can:

- Display component test projects on the server
- Query, run, and schedule tests
- Globally emulate SCA components or human tasks
- View component test projects, test suites, and test cases that run on the server
- View test case results: pass, fail, or exception

© Copyright IBM Corporation 2016

Figure 16-19. Component Test Explorer (1 of 2)

WB8601.2

Notes:

With the Component Test Explorer, you can manage and run test cases that are deployed to a test server or a stand-alone server. Like the Business Process Choreographer Explorer, the Component Test Explorer is a web client that you can invoke and run from inside or outside IBM Integration Designer. Using the Component Test Explorer, you can:

- Query and display component test projects, test suites, and test cases that are deployed on the server
- Select and run one or more component test projects, test suites, or test cases and then display and automatically save the results
- Schedule specific times to automatically run one or more component test projects, test suites, or test cases
- Define global server-wide emulators for components and import interfaces that are deployed on the server
- Specify global server-wide rules to automatically process and claim human tasks that are deployed on the server

WebSphere Education

Component Test Explorer (2 of 2)

- Component Test Explorer emulators:
 - Use if the component is not available at test time or if the execution can produce errors
 - Define a global emulator for any component in any module on server
 - Groovy assistance to emulate interface information for request and response parameters
 - Define and manage global emulators for specified human tasks

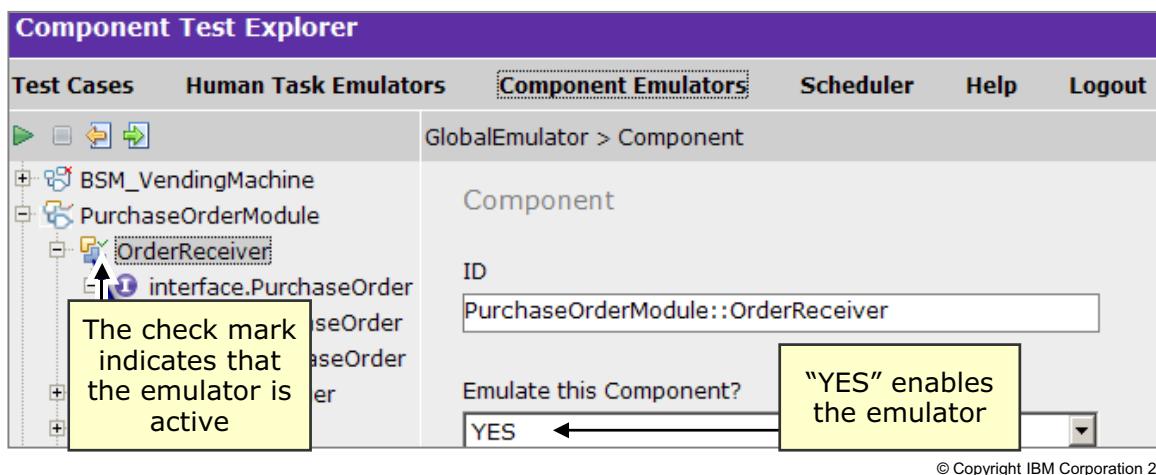


Figure 16-20. Component Test Explorer (2 of 2)

WB8601.2

Notes:

Using the component emulators page, you can define a global emulator for any component in any module on the system. You can globally emulate a component that is not yet available but is needed for testing. You can also globally emulate a component that is available for testing but is not yet implemented.

If you want to define an emulator for an operation, expand the component and interface, then select the operation name, and click **Create**. The details area displays the operation name and a table that contains the emulator rules as you create them. Each rule has a condition that describes the conditions under which the rule must be applied and a response that describes the output from the operation.

The condition section includes the Groovy statements that are constructed when you select the options in the request and operations sections. You can also enter a Groovy statement directly. An evaluated Groovy statement determines whether the response must be returned for the operation invocation. The condition section can contain many Groovy statements, but the last statement must evaluate to a Boolean string to determine whether the rule is used. Only the final statement is evaluated as the condition.

Server Logs view

- Console view returns a sizable volume of messages
- Server Logs view is filterable view of server logs in one view
 - SystemOut.log and SystemErr.log
- Able to view SCA errors along with server errors
- Highlight events with different colors
- Displays only server logs
- Potential performance effect
 - Full logs are loaded in memory

File (filtered): C:\IBM\...\server1\native_stderr.log; C:\IBM\...\server1\native_stdout.log...

Show All Record Types (Hierarchical) > with only Server State, Error and Warning Contents (Page 12 of 25)

Type	Time	Thread ID	Contents
Log message	Jun 19, 2014 19:29:30.319 PDT	00000001	WSVR0001I: Server server1 open for e-business
Log message	Jun 19, 2014 19:29:30.741 PDT	000001e7	The dependency javax.servlet.ServletContext is not supported!
Log message	Jun 19, 2014 19:30:05.740 PDT	000001f0	J2CA0086W: Shareable connection MCWrapper id 1c07b3cb Managed connection Ws
FFDC	Jun 19, 2014 19:30:06.084 PDT	000001f0	com.ibm.ws.ffdc.impl.FfdcProvider logIncident FFDC1003I: FFDC Incident emitted on
Exception	Jun 19, 2014 19:30:06.240 PDT	000001f0	CWLLG2015E: An unexpected runtime exception occurred. See exception for more de
Exception	Jun 19, 2014 19:30:06.428 PDT	000001f0	CWLLG0594E: An exception occurred while initializing the runnable thread. Error: com.
FFDC	Jun 19, 2014 19:30:06.428 PDT	000001f0	com.ibm.ws.ffdc.impl.FfdcProvider logIncident FFDC1003I: FFDC Incident emitted on

© Copyright IBM Corporation 2016

Figure 16-21. Server Logs view

WB8601.2

Notes:

The Server Logs view is used to display server console and log file records. Although the Server Logs view automatically displays console records for each server that is started, you can also manually load and display the server console and log file records for any server. When cross-component tracing is enabled, the Server Logs view also displays invocation records that can contain the invocation data that passes between components. The Server Logs view is the 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 integration 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 clicking **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 in to the Server Logs view
 - 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 display the records of server consoles and server logs that you load into the Server Logs view. The following four columns in the Server console and Log tabs provide detailed information about the records:
 - **Type:** Displays the types of the records, such as log messages, FFDC records, exception records, and invocation records
 - **Time:** Displays the date and time that the records were generated
 - **Thread ID:** Displays the thread IDs of the records
 - **Contents:** Displays the first line of the contents for the records
- You can also right-click any record and click **Properties** to open a properties dialog box that displays 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 native format in which they were originally generated. Also, if you open the properties dialog box for an FFDC record or an invocation record, such as a “start,” “fail,” or “end” invocation record, the properties dialog box contains another data field. This field displays the invocation data that is passed between components.
- **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, a status area returns messages that are related 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 displayed in hierarchical format and have check boxes that you can select for direct loading into the integration test client. Although you can choose to display records in a flattened format rather than a hierarchical format, the invocation records do not have check boxes. You cannot select them for direct loading into the integration test client.



The IBM Guided Activity Assistant

- Can assist you with every step of problem determination
- For new users: IBM Guided Activity Assistant explains in detail what steps to take
 - Provides supporting information to help you understand the problem and the suggested solutions
- For experienced users: IBM Guided Activity Assistant provides quick reminders of what steps to take
 - Shows only the information that is necessary for you to quickly move through the steps
 - More information is only a click away if you must brush up on a specific section
- For more information, see:
www.ibm.com/support/docview.wss?uid=swg27010135

© Copyright IBM Corporation 2016

Figure 16-22. The IBM Guided Activity Assistant

WB8601.2

Notes:

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

16.2.Deploying integration applications with serviceDeploy

Deploying integration applications with serviceDeploy



© Copyright IBM Corporation 2016

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

10.1

Figure 16-23. Deploying integration applications with serviceDeploy

WB8601.2

Notes:

Overview of serviceDeploy

- Automate deployment (assemble, generate, and compile) of Service Component Architecture applications
- Helpful in situations where not all SCA component resources are in a single workspace
- Can be invoked by using either a command line or Ant tasks
- SCA resources (modules) and other application components (WAR, EJB JAR, RAR, and utility JAR) are passed to `serviceDeploy` as a single input archive
- Provides basic assembly for SCA components and generates Java EE application artifacts and packages into a single EAR ready for installation

```
serviceDeploy archive [<-workingDirectory temppath>
    <-outputApplication outputpathname.ear>
    <-nojeeDeploy true|false> <-freeform true|false >
    <-cleanStagingModules true|false> <-keep true|false>
    <-ignoreErrors true|false>
    <-classpath jarpathname;rarpathname;warpathname;... -help]
```

© Copyright IBM Corporation 2016

Figure 16-24. Overview of serviceDeploy

WB8601.2

Notes:

The `serviceDeploy` tool is designed to help automate deployment of SCA modules. With `serviceDeploy`, you can take multiple components that multiple developers build, and bring them together in an application that can be installed on IBM Process Server. The `serviceDeploy` tool is available as a separate executable script that can be called from the command line. An Ant task is also available, which calls `serviceDeploy` for larger build and deploy processes.

You can specify various parameters for the `serviceDeploy` utility. The main parameters are the input archive (including the name of the module that contains the components that you want to generate into an installable `.ear` file) and the output name of the `.ear` file. Other options for troubleshooting or providing more resources can be used at build and deploy time.

Parameters: With the `serviceDeploy` tool, you can specify the following parameters:

- `-inputarchive`

A required positional parameter that specifies the `.jar`, `.zip`, or `.ear` file that contains the application to deploy. If the command is not run from the path in which the file exists, it must contain the full path for the file. The `.zip` file can be either a nested archive or a project interchange file.

- **-classpath**
An optional parameter that specifies the locations of required resource files (.jar and .rar). The path to each file must be a fully qualified path that is separated with semicolons (;) with no spaces.
- **-cleanStagingModules**
An optional parameter that specifies whether to delete staging modules within an input .ear file before deployment. By default, the serviceDeploy command imports the existing staging modules and their contents.
- **-freeform**
An optional parameter that specifies that the Java EE subdirectory in the service.jar file must be treated as a free-form project.
- **-help**
An optional parameter that is used to display the parameters for this command.
- **-ignoreErrors**
An optional parameter that specifies that the serviceDeploy command builds a .ear file regardless of errors while building or validating the application. By default, the serviceDeploy command does not generate a .ear file if an application has errors.
- **-keep**
An optional parameter that specifies whether to save any temporary files that are generated after deployment. By default, the serviceDeploy command deletes the temporary workspace.
- **-novalidate**
An optional parameter that specifies whether to disable validation of all artifacts when using serviceDeploy. By default, the serviceDeploy validates all artifacts for the application.
- **-outputApplication**
An optional parameter that specifies the name of the .ear file that the serviceDeploy command creates. The default is `inputarchiveApp.ear`, where `inputarchive` is the file name minus the extension that is specified for the input .jar file.
- **-uniqueCellID**
An optional parameter that specifies a string identifier that is used to create a unique instance of the application. The identifier must be unique in the context of the cell.
- **-workingDirectory**
An optional parameter that specifies a directory that the serviceDeploy command uses to write temporary files.

Input: The following file types can be used as input to the serviceDeploy command:

- **.jar**
This file type is the most useful type for simple applications. The resulting .ear file contains a single .jar file and the required staging modules. The .jar file must contain the `service.module` file.

- .zip (project interchange)

You can use IBM Integration Designer to export an archive file in project interchange format. This format is unique to Eclipse development. The exported .zip file must contain exactly one project with the service.module file. The resulting .ear file contains a number of modules, depending upon exactly what is in the project interchange.

- .zip

You can create a .zip file that contains .jar files, .war files, and .rar files. Exactly one .jar file must contain the service.module file. All contained archives are included in the final exported .ear file.

- .ear

You can always run the serviceDeploy command against a .ear file while exactly one .jar file in the .ear file contains a service.module file.

Output: When serviceDeploy completes processing, it creates a .ear file in the directory from which the command is run unless the -outputApplication parameter is specified.

- **Example of the serviceDeploy command:**

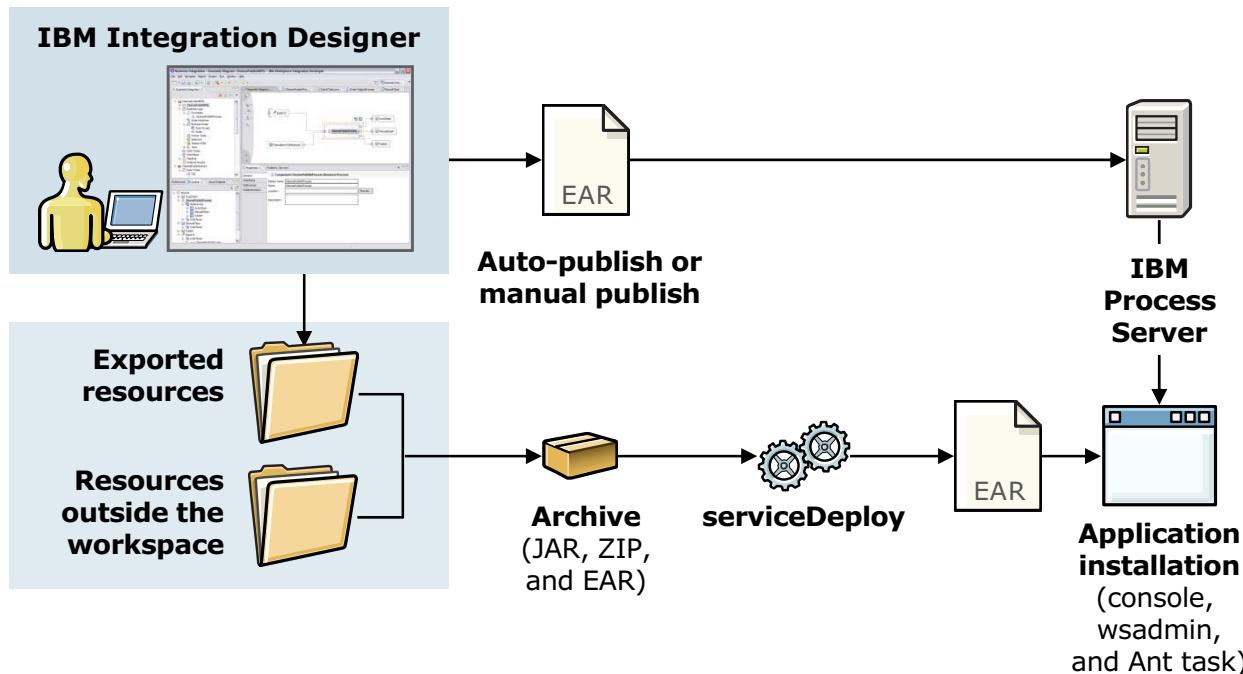
Consider the following command example:

```
servicedeploy MyValueModule.jar -classpath  
"c:\java\myvaluemoduleres.rar;c:\java\commonres.jar" -freeform true -keep
```

This command:

- Creates an application file that is called MyValueModule.ear from the MyValueModule.jar file
- Specifies that the resources exist in the directories c:\java\myvaluemoduleres.rar and c:\java\commonres.jar
- Enables the Java subdirectory within the .jar file as free-form
- Keeps the temporary files that are generated during deployment

serviceDeploy: Single developer



© Copyright IBM Corporation 2016

Figure 16-25. serviceDeploy: Single developer

WB8601.2

Notes:

This page demonstrates a single developer who is using a `serviceDeploy` to do all the tasks necessary for deployment of an application. When all the components are placed on the assembly editor that is wired, and saved, all the necessary code is generated into an EAR file. This file can be deployed manually or by using the “add and remove projects” option to publish modules to the server. A single user might want to use `serviceDeploy` for situations where resources are available outside the workspace that the user wants to add into the EAR file.

You can also use `serviceDeploy` to bring multiple components together in a single EAR file; `serviceDeploy` can include other resources and dependencies (such as a web module). This capability includes artifacts that you do not have in your workspace but are available in another location, for instance on a network server. You first create an archive, you then pass it to a `serviceDeploy` script, and the output is an EAR file. At that point, you can use some other installation mechanism such as the administrative console, `wsadmin`, or an Ant task. `serviceDeploy` is a script to assist with deployment. `serviceDeploy` does not install the application.

Multiple developers and serviceDeploy

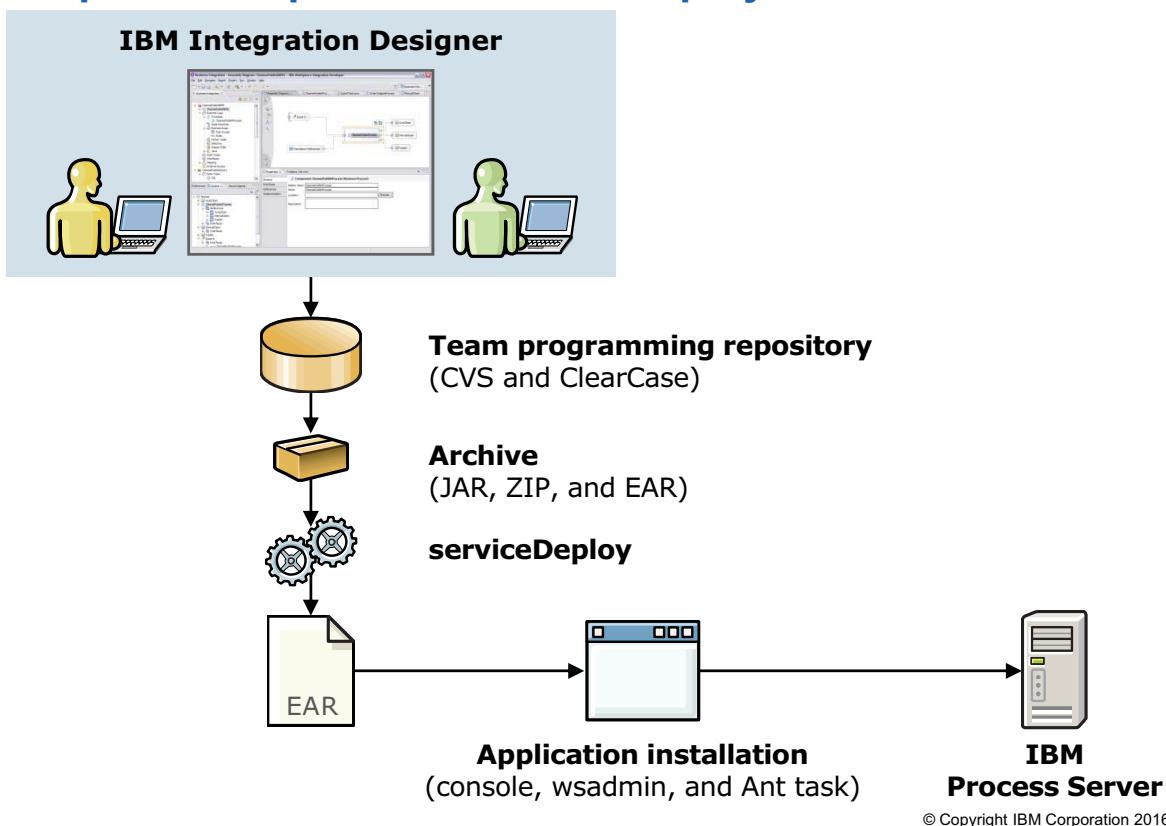


Figure 16-26. Multiple developers and serviceDeploy

WB8601.2

Notes:

This page demonstrates the use of `serviceDeploy` in an environment with multiple developers who are working on different components. The work can be brought together, for instance in a nightly batch job, and the application can be assembled and installed on the server for testing the next morning.

Developers can work in a team programming environment, storing resources in a repository such as Rational ClearCase or Concurrent Versions System (CVS). The appropriate APIs or commands to retrieve those artifacts out of the repository are used. The artifacts are placed in an archive, and the archive is passed to the `serviceDeploy` tool. `serviceDeploy` compiles the resources into an EAR file, and the application is installed by using the administrative console, `wsadmin`, or an Ant task. When your SCA application is installed on IBM Process Server, you can manage the runtime environment through the administrative console.



Unit summary

- Describe the advanced testing facilities that are available in IBM Integration Designer, including the Component Test Explorer and cross-component trace
- Describe the integration debugger
- Define the purpose and function of the serviceDeploy tool
- Describe how to use serviceDeploy in single-developer and multiple-developer environments

© Copyright IBM Corporation 2016

Figure 16-27. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. True or False: You can open multiple instances of the test client and use them to do simultaneous testing.
2. True or False: An EAR file that is ready for deployment to the IBM Process Server runtime environment is the output of the `serviceDeploy` tool.
3. True or False: Component projects include test suites, which are containers for test cases.
4. True or False: Cross-component trace provides a separate server logs view and a separate SCA trace view.

© Copyright IBM Corporation 2016

Figure 16-28. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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



Checkpoint answers

1. True.
2. True.
3. True.
4. False. Cross-component trace provides server logs and SCA trace in one single view.

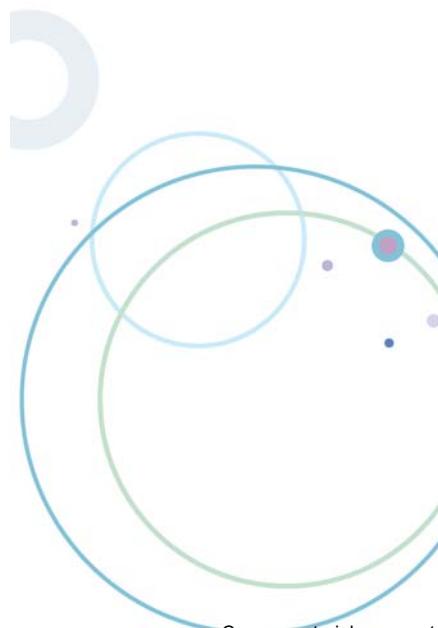
© Copyright IBM Corporation 2016

Figure 16-29. Checkpoint answers

WB8601.2

Notes:

Exercise 14



Using component tests

© Copyright IBM Corporation 2016

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

10.1

Figure 16-30. Exercise 14

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Create a component test project with an operation-level test case
- Create a component test project with a scenario-based test case
- Run component test project test suites in the IBM Integration Designer integrated test environment

© Copyright IBM Corporation 2016

Figure 16-31. Exercise objectives

WB8601.2

Notes:

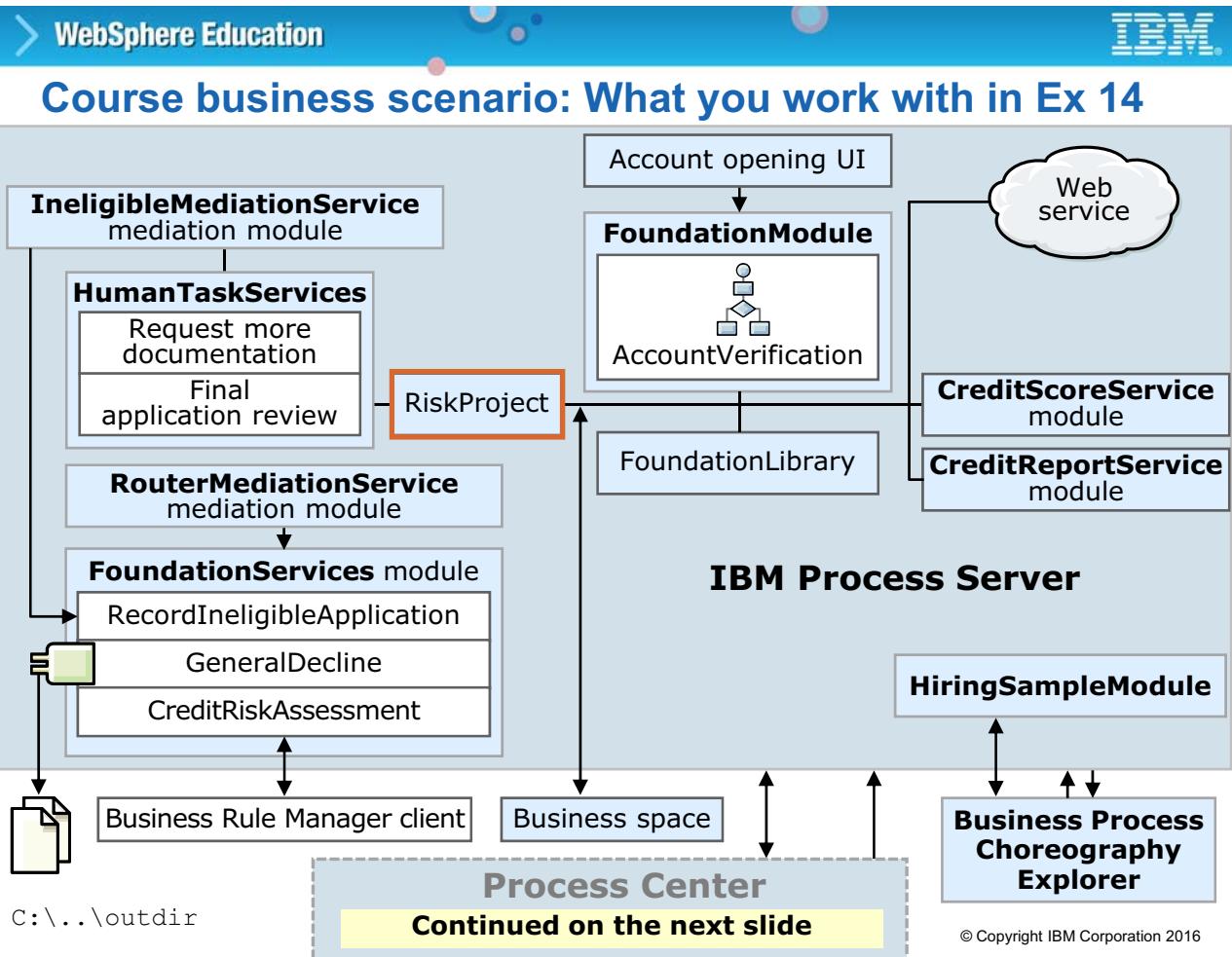


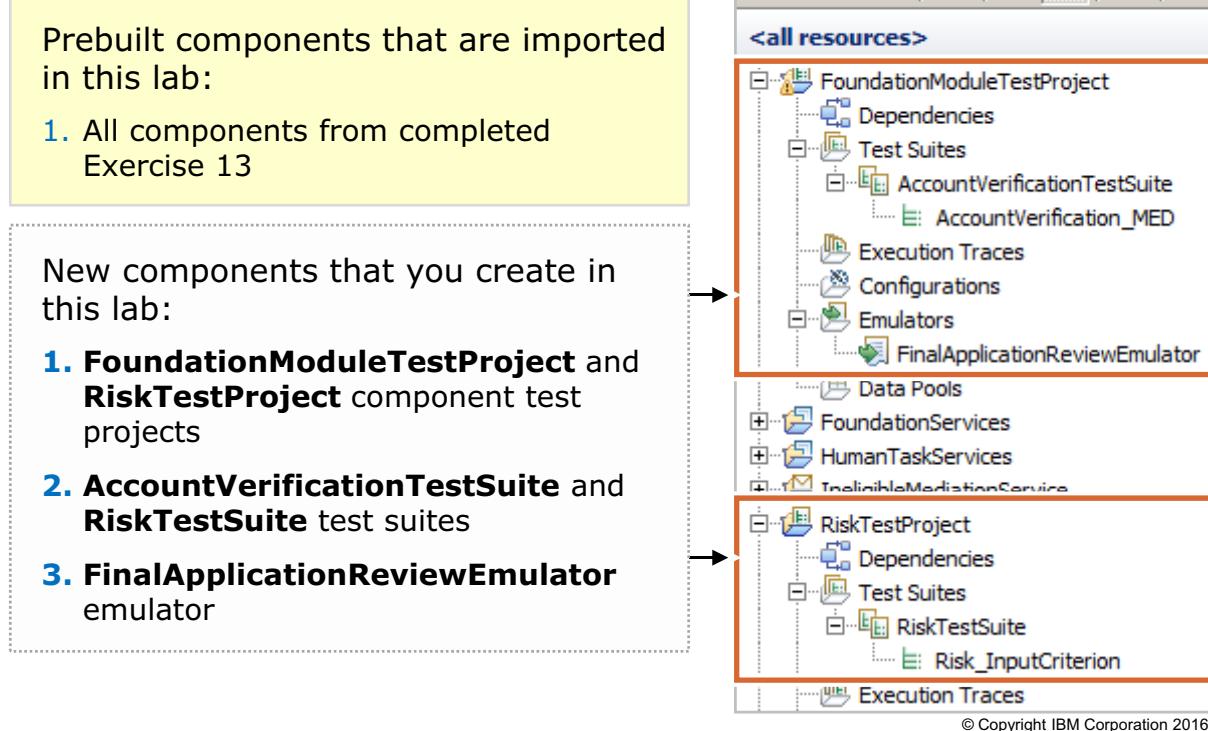
Figure 16-32. Course business scenario: What you work with in Exercise 14

WB8601.2

Notes:

WebSphere Education

Components that are required for Exercise 14



© Copyright IBM Corporation 2016

Figure 16-33. Components that are required for Exercise 14

WB8601.2

Notes:

If applicationDecision is set to false during FinalApplicationReview (the application is declined) and the customer's creditRisk is HIGH, the application is routed through the "generate decline" component. If applicationDecision is set to false during FinalApplicationReview and the customer's creditRisk is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the RouteRequest component. The RouteRequest flow component contains the mediation logic that routes the application to the appropriate decline service.

The RouteRequest mediation flow consists of both a request flow and a response flow. In the flow, the CustomerApplication is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the AccountVerification process.

Unit 17. IBM Process Center

What this unit is about

This unit teaches you how to use IBM Process Center to deploy, test, and manage applications.

What you should be able to do

After completing this unit, you should be able to:

- Describe the purpose and business value of IBM Process Center
- Define the components of IBM Process Center
- Describe how to use the IBM Process Center Console to deploy, test, and manage IBM BPM applications

How you will check your progress

- Checkpoint questions
- Exercise



Unit objectives

- Describe the purpose and business value of IBM Process Center
- Define the components of IBM Process Center
- Describe how to use the IBM Process Center Console to deploy, test, and manage IBM BPM applications

© Copyright IBM Corporation 2016

Figure 17-1. Unit objectives

WB8601.2

Notes:



Topics

- Introduction to IBM Process Center
- Components of IBM Process Center
- Manage, test, and deploy IBM BPM applications

© Copyright IBM Corporation 2016

Figure 17-2. Topics

WB8601.2

Notes:

17.1.Introduction to IBM Process Center

Introduction to IBM Process Center



© Copyright IBM Corporation 2016

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

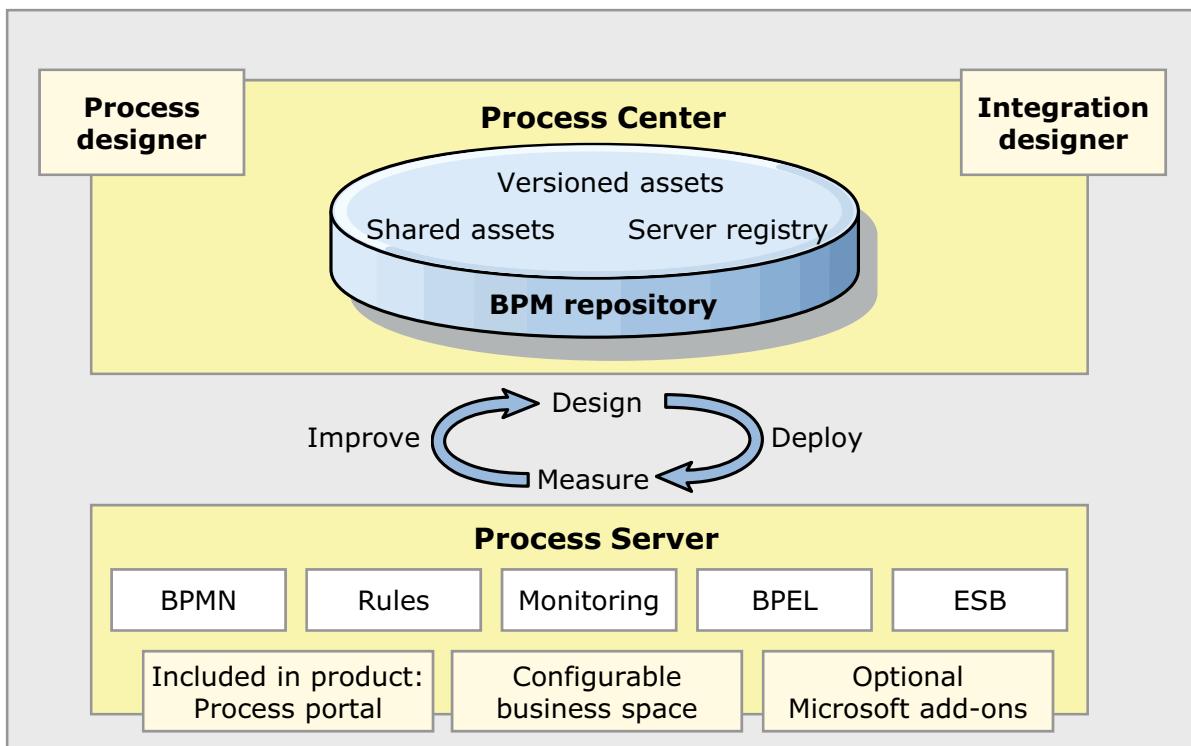
10.1

Figure 17-3. Introduction to IBM Process Center

WB8601.2

Notes:

IBM Business Process Manager



© Copyright IBM Corporation 2016

Figure 17-4. IBM Business Process Manager

WB8601.2

Notes:

IBM Business Process Manager uses a concept that is called the shared model. It means that no matter what is being done within the overall solution, it has only one common repository along with a single representation of that solution. Therefore, it is impossible to get two phases of the same solution out of sync with each other.

This shared model is realized through the IBM Business Process Manager component that is called the IBM Process Center. IBM Process Center is a key component within IBM Business Process Manager. Part of the IBM Process Center is a data repository, which is called the repository. Within the repository, there exists the representation of the solution. The IBM BPM tool connects as a client to the IBM Process Center to obtain copies of the solution to work on. When a user changes an artifact and saves those changes, the results are written back to the repository.

The IBM Process Center repository is implemented as tables within a database (commonly DB2).

IBM Process Center capabilities

- Repository for all IBM BPM assets
 - Process applications, reusable toolkits, monitor models, and many more
- Lifecycle management and deployment of all applications
 - Manage dependencies, versions, and deployment to servers
- Includes execution environment for development and testing
- Equally accessible from IBM Process Designer and from IBM Integration Designer
- Central governance
- Web interface that uses the IBM Process Center console

© Copyright IBM Corporation 2016

Figure 17-5. IBM Process Center capabilities

WB8601.2

Notes:

You can use the IBM Process Center repository to share artifacts with other users who are developing process applications and toolkits.

17.2.Components of IBM Process Center

Components of IBM Process Center



© Copyright IBM Corporation 2016

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

10.1

Figure 17-6. Components of IBM Process Center

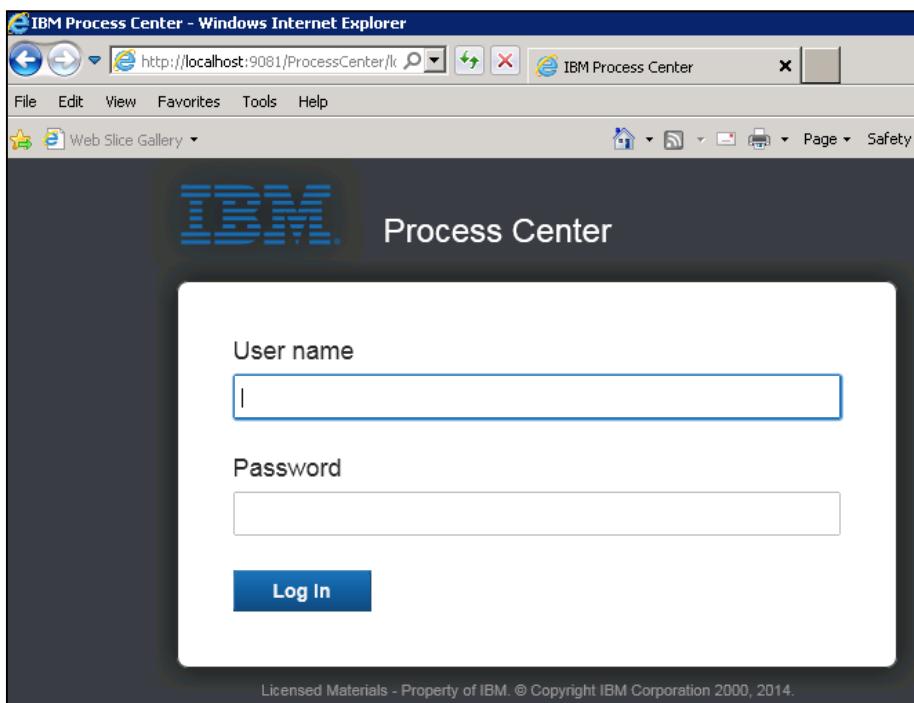
WB8601.2

Notes:



IBM Process Center Console

- The IBM Process Center Console provides a web-based interface for managing the Process Center maintained projects
- It provides the tools that are needed to maintain the repository



© Copyright IBM Corporation 2016

Figure 17-7. IBM Process Center Console

WB8601.2

Notes:

The Process Center Console provides a web-based interface for managing the Process Center maintained projects. The default URL for IBM Process Center Console is:

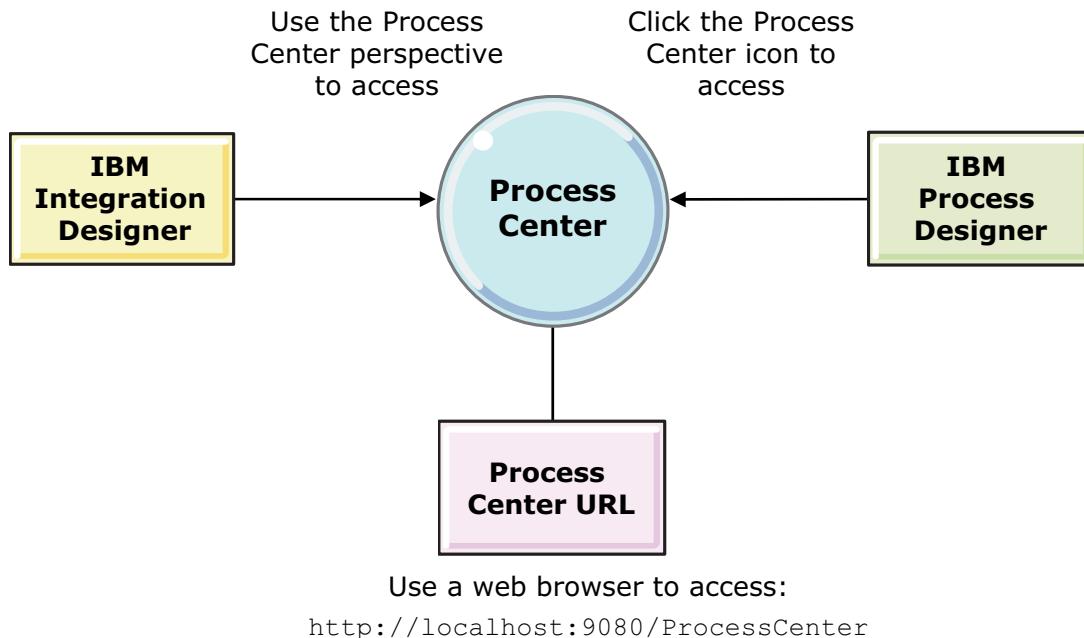
<http://localhost:9080/ProcessCenter>

The IBM Process Center includes a repository for all processes, services, and other assets. The IBM Process Center Console provides the tools that you need for maintaining the repository.

From the Process Center console:

- You can create process applications and toolkits and grant other users access to those process applications and toolkits.
- Administrators install process applications that are ready for testing or production on the IBM Process Servers in those environments.
- Administrators manage running instances of process applications in configured environments.

Accessing IBM Process Center



© Copyright IBM Corporation 2016

Figure 17-8. Accessing IBM Process Center

WB8601.2

Notes:

IBM Process Center can be accessed in several ways:

- Switching to the Process Center perspective in the IBM Integration Designer
- Clicking the IBM Process Center icon at the upper right in the IBM Process Designer
- Using a web browser at the default URL: `http://localhost:9080/ProcessCenter`

The Process Center view and capability can vary slightly, depending which tool you are using. The tool dictates the capability, which depends on its function.



IBM Process Center: Process Apps

- A process application is the container for a solution
- Initially created through the Process Center console
- It is given a name and a tag that is called an *acronym*
- The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages

© Copyright IBM Corporation 2016

Figure 17-9. IBM Process Center: Process Apps

WB8601.2

Notes:

A process application is the container for a solution. You can loosely think of it as a project. The process application is initially created through the Process Center console. It is given a name and a tag that is called an *acronym*. The acronym must be unique and can be no more than 7 characters in length. When the process application container is created, artifacts can then be further created within it using the IBM Process Designer.

The process application and its artifact contents are stored within a repository that the IBM Process Center hosts and manages. The main Process Apps page has a button to create a process application.

- Container for artifacts that are used in solutions
 - Does not result in a deployable application
- Can be “included” or “used” by one or more process applications
 - Similar to a library with artifacts
- Can be added as a dependency to a process application

© Copyright IBM Corporation 2016

Figure 17-10. IBM Process Center: Toolkits

WB8601.2

Notes:

Similar to process applications, a toolkit can also be thought of as a container for artifacts that are used in solutions. Unlike a process application, a toolkit does not result in a deployable application. Instead, the contents of the toolkit can be “included” or “used” by one or more process applications.

When Process Center is installed and configured, an IBM supplied toolkit that is called “System Data” is automatically imported into the repository. This toolkit is marked as read-only and is implicitly dependent on all other process applications and toolkits. It is the System Data toolkit that contains the core definitions for data structures and other items that are common across all process applications.

Toolkits have their own tabs in the Process Center consoles. From that point, new toolkits can be created or exported, and otherwise managed in a similar fashion to the ones of the process applications.



- A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made
- Allows users to revert in time to the state of the snapshot
- Allows for creating a version of toolkits and process applications

© Copyright IBM Corporation 2016

Figure 17-11. IBM Process Center: Snapshots

WB8601.2

Notes:

A snapshot is a copy of the state of all the artifacts in a process application or toolkit at the point in time when the snapshot was made. The purpose of taking a snapshot is so you can revert in time to the state of the snapshot that is needed. A snapshot can be captured by clicking the **Snapshot** icon in IBM Process Designer.

A snapshot is required in some circumstances such as:

- A snapshot of a toolkit is required before it can be added as a dependency on other toolkits or process applications.
- A snapshot of a process application is required before that application can be installed on IBM Process Server.
- A snapshot is required before a workspace can be created.

Just like process applications, toolkits can have snapshots that are taken of them, allowing all the artifacts in a toolkit to have versions.

To add a toolkit as a dependency to a process application, the toolkit must first have a snapshot that is associated with it. This requirement is because the dependency added to the process application is **not** just the name of the toolkit, but is instead a specific snapshot of that toolkit.

The screenshot shows the 'Servers' tab selected in the IBM Process Center. The interface is clean and modern, designed for managing multiple servers. The single server listed is clearly identified by its name and environment.

- The Servers tab lists the IBM Process Servers that are connected to the IBM Process Center
 - IBM Process Server can be a stand-alone server
 - IBM Process Server can be a server that is running inside the IBM Integration Designer test environment
 - Multiple servers can be connected
 - Multiple environments can be connected: development, testing, staging, and production

© Copyright IBM Corporation 2016

Figure 17-12. IBM Process Center: Servers

WB8601.2

Notes:

The servers that are shown on the Servers tab are the IBM Process Servers that are connected to the IBM Process Center. Authorized users can install snapshots of process applications on connected IBM Process Servers. For each server, you can view the snapshots that are currently installed.

The screenshot shows the IBM Process Center Admin interface. At the top, there are tabs for 'Process Apps', 'Toolkits', 'Servers', and 'Admin'. The 'Admin' tab is highlighted with a red border. Below the tabs, there are three buttons: 'Manage Users' (highlighted with a green border), 'Activity Log', and 'Registration'. Under the 'Manage Users' button, there is a section titled 'Admin' with a red border. This section contains two entries: 'tw_admins' and 'tw_authors', each represented by a user icon and a label.

- Add new users and groups to the list of authorized users
 - Granting users authority to access the repository allows them to log in
- Process Apps and Toolkits are individually controlled with their own access control

© Copyright IBM Corporation 2016

Figure 17-13. IBM Process Center: Admin

WB8601.2

Notes:

Granting users authority to access the repository allows them to log in to the IBM Process Center console. However, this access does not give privileges to work on or even see all the process applications in the environment. Process applications and toolkits are individually controlled with their own access control lists. From within the Process Apps section of the IBM Process Designer or the IBM Process Center console, an application can be selected and the **Manage** tab clicked. On that tab, it contains a section with which users and groups can be associated. These groups define the permissions for those entities.

Three roles that a user or group can have are provided here:

- **Read:** This role allows a user or group to see the project and see the artifacts within it. The read role cannot be removed without removing the user or group association completely. If a user or group is *not* associated with a process application, then the user or group has no authorities on that application. If an artifact is opened and the user has read authority only, the artifact is flagged as read-only in the editor.
- **Write:** This role allows the user or group to update or add artifacts into the process application.
- **Admin:** This role allows the user or group to administer the process application.

17.3. Manage, test, and deploy IBM BPM applications

Manage, test, and deploy IBM BPM applications



© Copyright IBM Corporation 2016

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

10.1

Figure 17-14. Manage, test, and deploy IBM BPM applications

WB8601.2

Notes:

The screenshot shows the WebSphere Education interface with the following elements:

- Header:** WebSphere Education and IBM logos.
- Navigation Bar:** Process Apps, Toolkits, Servers, Admin.
- Toolbar:** Sort By: Acronym dropdown, All, Favorites.
- Content Area:**
 - Two items listed: Account Verification Skeleton (AVS) and Hiring Sample Advanced (HSAV1).
 - Each item has a "Last updated on 6/18/14 by pcodeadmin" timestamp.
 - Each item has an "Open in workspace" button with a red border.

- Import process applications and toolkits into your workspace from the Process Center repository
 - Can now use them with your modules and libraries
- Must be in the Process Center perspective to import the process application or toolkit into the workspace
- View and work with the newly imported workspace in the Business Integration perspective

© Copyright IBM Corporation 2016

Figure 17-15. Importing from the IBM Process Center repository

WB8601.2

Notes:

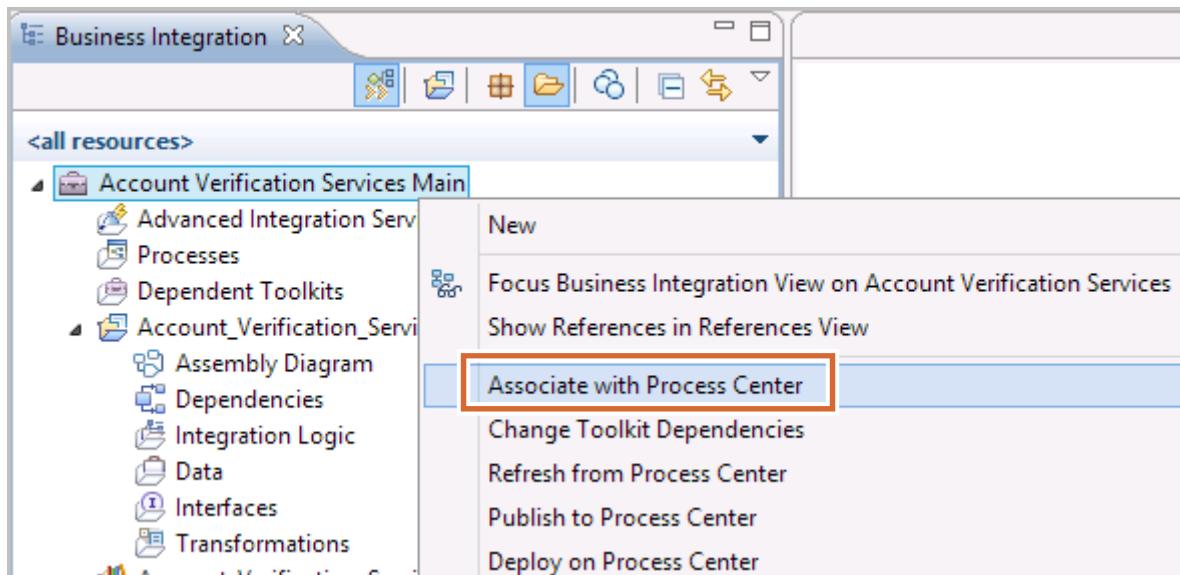
You can import process applications and toolkits into your workspace from the IBM Process Center repository, and then you can use them with your modules and libraries.

To open the process application into a workspace in the Integration Designer, you must use the IBM Integration Designer to access the IBM Process Center.

WebSphere Education

Associating a module or library

- Associate a module or library with a process application or toolkit to:
 - Add more functions to the application
 - Take advantage of version control on the Process Center



© Copyright IBM Corporation 2016

Figure 17-16. Associating a module or library

WB8601.2

Notes:

You can associate a module or library with a process application or toolkit to add more functions to the application, or to take advantage of version control on the IBM Process Center.

Suppose a module that contains a long-running process is deployed to a process server and you now want to associate the module with a process application. In this case, you must first consider whether you want to migrate your process instance:

- If you want to migrate the process instance, you must create a process version before you associate the module with the process application. You can create a process version by right-clicking your module in the Business Integration view and clicking **New Process Version**.
- If you do not want to migrate the process instance, you can proceed to associate the module with the process application.

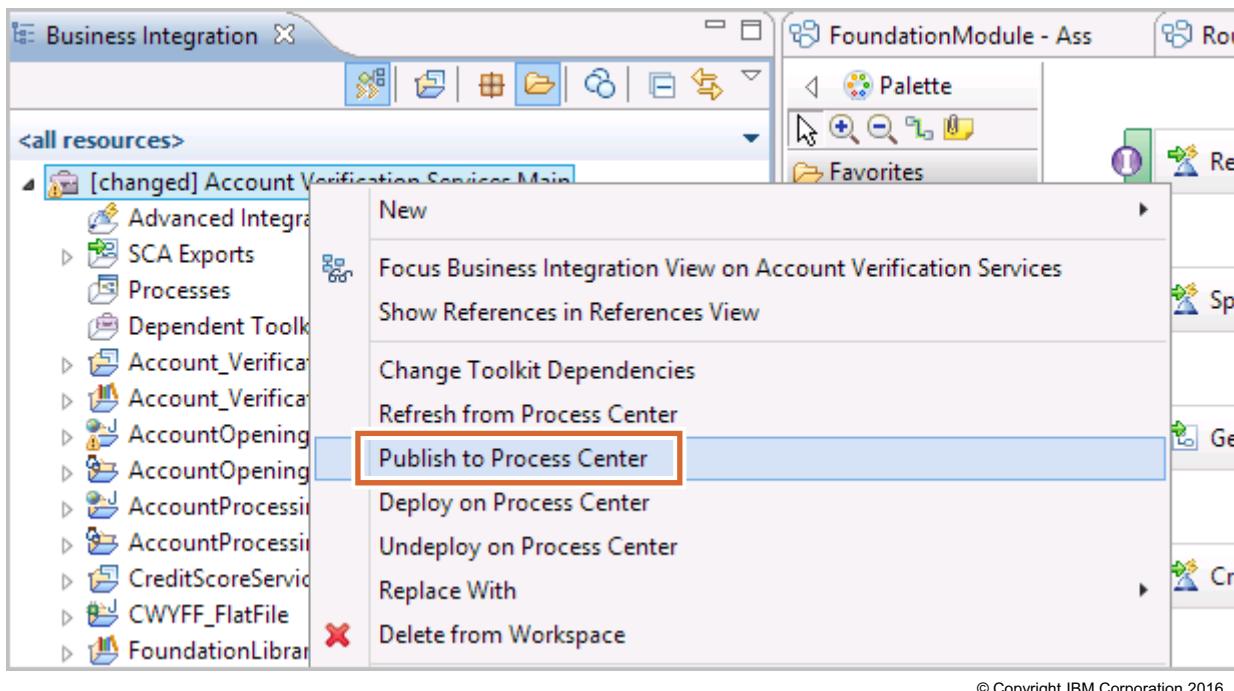
Associating a module or library with a process application or toolkit has wider implications than you might first see. Modules and libraries that are associated with toolkits can be shared with other process applications in addition to the one in your workspace. Modules and libraries that are associated with process applications are also visible within the process application. Remember too

that when you bring a process application or toolkit into your workspace, you might be bringing in a snapshot from a previous point in time.

WebSphere Education

Publish and synchronize

- The publish command updates the IBM Process Center with the changes in your workspace



© Copyright IBM Corporation 2016

Figure 17-17. Publish and synchronize

WB8601.2

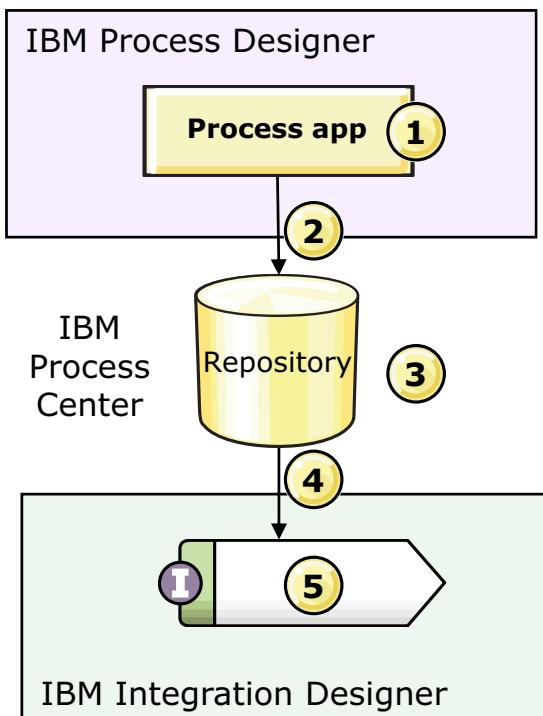
Notes:

When you update your process applications and toolkits, you must update the corresponding process applications and toolkits in the IBM Process Center with your changes. The publish command updates the IBM Process Center with the changes in your workspace.

The process application or toolkit is then updated. If a conflict exists because an element in your workspace and an element in the IBM Process Center are identical, then you are warned if you want to proceed. If you proceed, the element in the IBM Process Center is overridden.

While you are updating an artifact, another user might be updating the same artifact at the same time. To make sure that you are using the current version of an artifact, you must synchronize the versions that exist in the workspace and the Process Center repository. When you try to publish the artifact to the IBM Process Center, synchronization automatically starts to merge changes into the workspace. If a conflict exists, the conflict shows in the synchronization dialog box. You must resolve the conflict before you can publish your changes to the IBM Process Center. To make sure that you are using the current version of an artifact, you must synchronize the versions that exist in the workspace and the IBM Process Center repository. Select the file that contains the changes you want to keep, and then click **Commit** to publish your changes to the repository.

IBM Process Designer artifacts in IBM Integration Designer



1. Assets are built in IBM Process Designer
2. A snapshot of an asset is stored in IBM Process Center
3. Developer uses the Process Center perspective in IBM Integration Designer to read a repository
4. Click **Open in Workspace**
 - Assets are added to synchronized project
5. Use business processes as export or import components

© Copyright IBM Corporation 2016

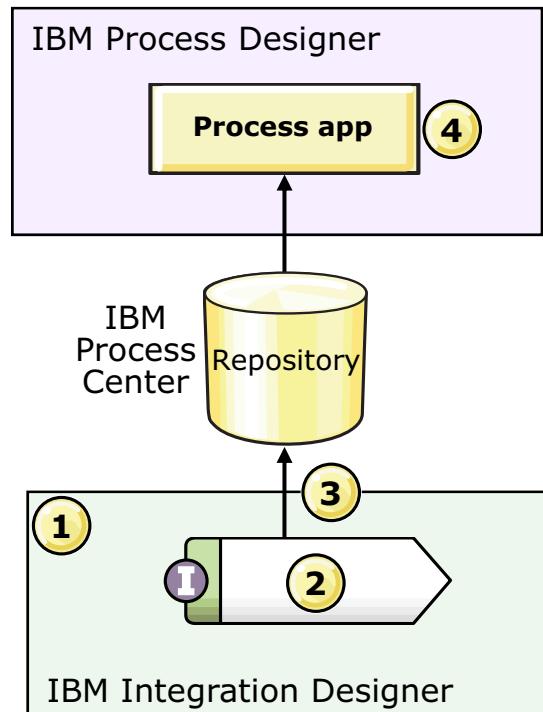
Figure 17-18. IBM Process Designer artifacts in IBM Integration Designer

WB8601.2

Notes:

IBM Integration Designer artifacts in IBM Process Designer

- Modules that are associated with a process app or toolkit
- “Make operations visible to IBM Process Designer” on import or export component
- Changes published to repository
- Open updated process app or toolkit
 - Imports and exports are displayed as “Advanced Integration service” implementations
 - Supporting artifacts, such as business objects, are read-only



© Copyright IBM Corporation 2016

Figure 17-19. IBM Integration Designer artifacts in IBM Process Designer

WB8601.2

Notes:

- Removing process applications from the Process Center repository
 - Use the Process Designer to first archive the process application and then delete it

© Copyright IBM Corporation 2016

Figure 17-20. Manage

WB8601.2

Notes:

To remove the process application, follow these steps:

1. From the Process Apps view of the Process Designer, click the process application that you want to remove.
2. Click **Manage**.
3. From the Manage page, click **Archive Process App**.
4. Click **Process Apps** to return to the list of process applications.
5. Click **Archived** to display a list of archived processes.
6. Click the **X** icon next to the process application you want to remove. When you are prompted, confirm the action.

Cleanup snapshots

- Use the `BPMSSnapshotCleanup` command to delete all the unnamed and archived snapshots of a process application on a Process Center server
- Restrictions
 - You must be a repository administrator
 - You cannot delete the first snapshot of a process application even though it might be unnamed or archived
 - The first snapshot contains original information about the snapshot that is displayed in the history pane in Process Designer
 - You must archive named snapshots before you delete them
 - To avoid conflicts between operations and snapshots on the Process Center, you must run the `BPMSSnapshotCleanup` command at specified intervals
 - Intervals refer to when no operations are on the Process Center and no connections are between the Process Designer and the Process Center

© Copyright IBM Corporation 2016

Figure 17-21. Cleanup snapshots

WB8601.2

Notes:

You can delete unnamed snapshots and archived snapshots with the same `BPMSSnapshotCleanup` command by using different parameters. You can also delete snapshots in batches for better performance.

For details about command parameters and examples for Jython and Jacl, see the `BPMSSnapshotCleanup` command in the product documentation.

WebSphere Education

IBM.

Deploy

Process Apps Toolkits Servers Admin

AccountServicesApp (ASA001) ★ Solutions **Snapshots** History Manage

Sort Snapshots By: Date All Deployed Archived

ASA snapshot 1 Edit | Export | Clone | Deploy | Activate | Archive

- Snapshots can be deployed to IBM Process Servers connected to the IBM Process Center
- Both the IBM Process Center server and the IBM Process Server where the application is being deployed must be running
- When deployed, dependencies are deployed as well

© Copyright IBM Corporation 2016

Figure 17-22. Deploy

WB8601.2

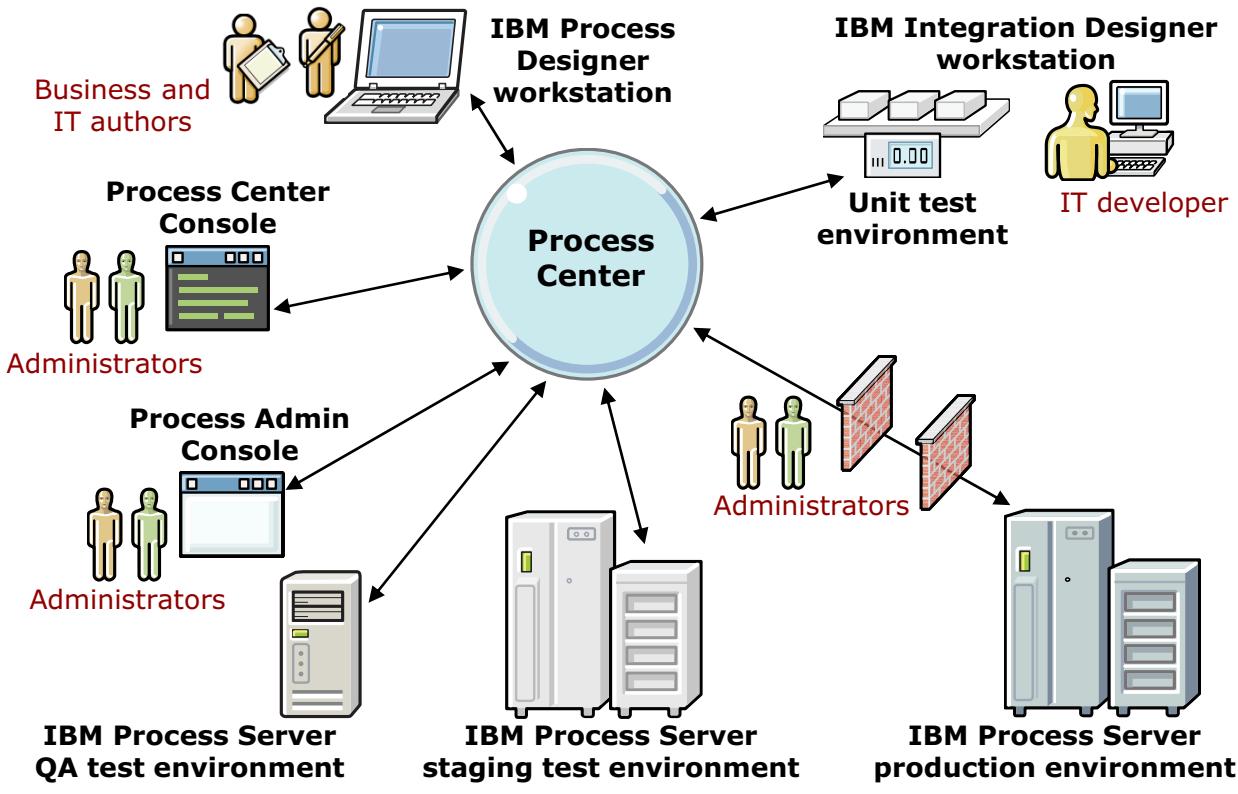
Notes:

Snapshots can be deployed to IBM Process Servers that are connected to the IBM Process Center.

Both the IBM Process Center server and the IBM Process Server where the application is being deployed must be running.

When applications are deployed, dependencies are also deployed.

Typical development and deployment scenario



© Copyright IBM Corporation 2016

Figure 17-23. Typical development and deployment scenario

WB8601.2

Notes:

The diagram displays a typical development and deployment topology that you saw in the first unit. Now that you are already familiar with the course lab environment after working on exercises, it helps to reinforce some concepts. For in-depth topology and IBM BPM administration coverage, see the IBM Business Process Manager Standard and Advanced administration courses.

A test process server is installed with IBM Integration Designer in a unit test environment (UTE). In this mode, the unit tester can test SCA modules locally in the IBM Process Server that is 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. Connected to the central IBM Process Center are independent QA, staging, and production environments, each with its own full stand-alone IBM Process Server. 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 organization requirements.



Unit summary

- Describe the purpose and business value of IBM Process Center
- Define the components of IBM Process Center
- Describe how to use the IBM Process Center Console to deploy, test, and manage IBM BPM applications

© Copyright IBM Corporation 2016

Figure 17-24. Unit summary

WB8601.2

Notes:



Checkpoint questions

1. True or False: The IBM Process Center console can be accessed in only two ways; the first is through IBM Process Designer, and the second is through a web browser by using the Process Center URL.
2. True or False: Snapshots must be created before you can deploy a process application.
3. True or False: You can import both the toolkit and the process application inside an IBM Integration Designer workspace.

© Copyright IBM Corporation 2016

Figure 17-25. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. False. The third way is to use the IBM Integration Designer perspective.
2. True.
3. True. You import both the toolkit and the process application through the IBM Process Center repository.

© Copyright IBM Corporation 2016

Figure 17-26. Checkpoint answers

WB8601.2

Notes:

Exercise 15



Bringing the UTE Process Server
online

© Copyright IBM Corporation 2016

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

10.1

Figure 17-27. Exercise 15

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Verify the status of the connected Process Server
- Run wsadmin commands for environment configuration
- Exchange SSL certificates between Process Center and Process Server

© Copyright IBM Corporation 2016

Figure 17-28. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 15 (1 of 2)

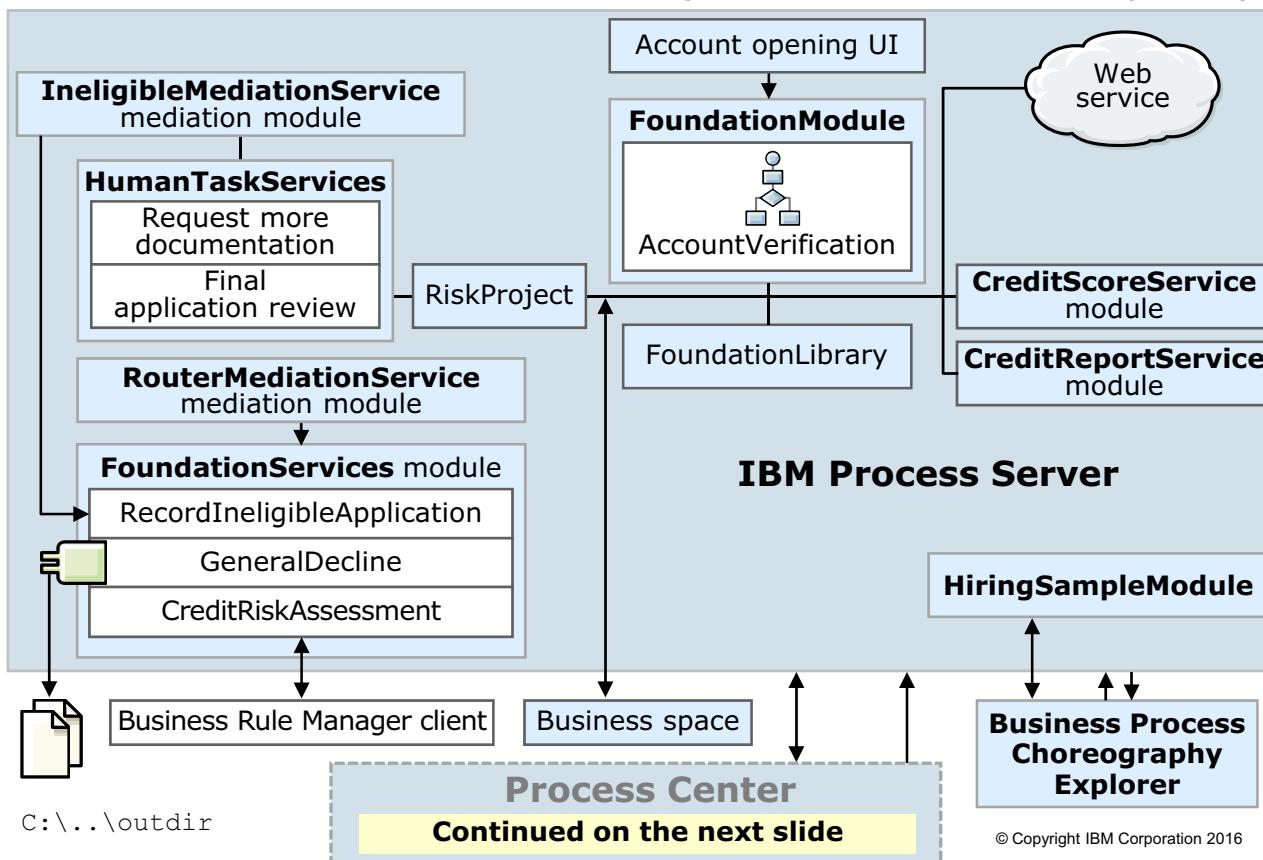


Figure 17-29. Course business scenario: What you work with in Exercise 15 (1 of 2)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 15 (2 of 2)

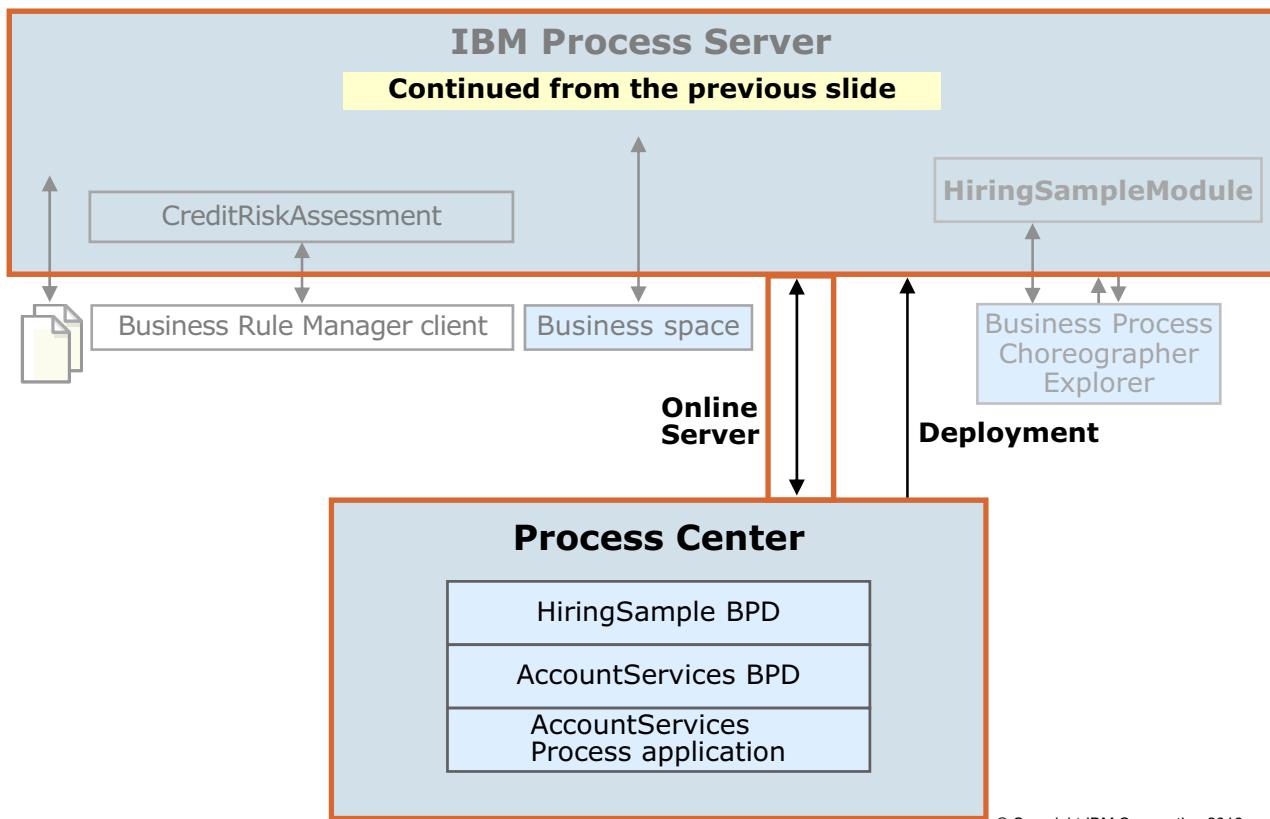


Figure 17-30. Course business scenario: What you work with in Exercise 15 (2 of 2)

WB8601.2

Notes:

Components that are required for Exercise 15

Prebuilt components that are imported in this lab:

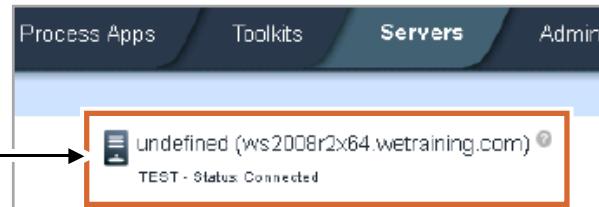
None

New components that you create in this lab:

None

Environment changes that you make in this lab:

1. You do not work with the Integration Designer in this exercise
2. Run the **wsadmin** command to change the connection properties
3. Change the environment through the administrative console



© Copyright IBM Corporation 2016

Figure 17-31. Components that are required for Exercise 15

WB8601.2

Notes:

If `applicationDecision` is set to false during `FinalApplicationReview` (the application is declined) and the customer's `creditRisk` is HIGH, the application is routed through the "generate decline" component. If `applicationDecision` is set to false during `FinalApplicationReview` and the customer's `creditRisk` is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the `RouteRequest` component. The `RouteRequest` flow component contains the mediation logic that routes the application to the appropriate decline service.

The `RouteRequest` mediation flow consists of both a request flow and a response flow. In the flow, the `CustomerApplication` is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the `AccountVerification` process.

Exercise 16

Exploring IBM Process Center



© Copyright IBM Corporation 2016

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

10.1

Figure 17-32. Exercise 16

WB8601.2

Notes:



Exercise objectives

After completing this exercise, you should be able to:

- Explore the IBM Process Center repository
- Create a toolkit in IBM Process Center
- Associate IBM Integration Designer artifacts with the toolkit
- Generate access to process applications and toolkits
- Archive and delete process applications

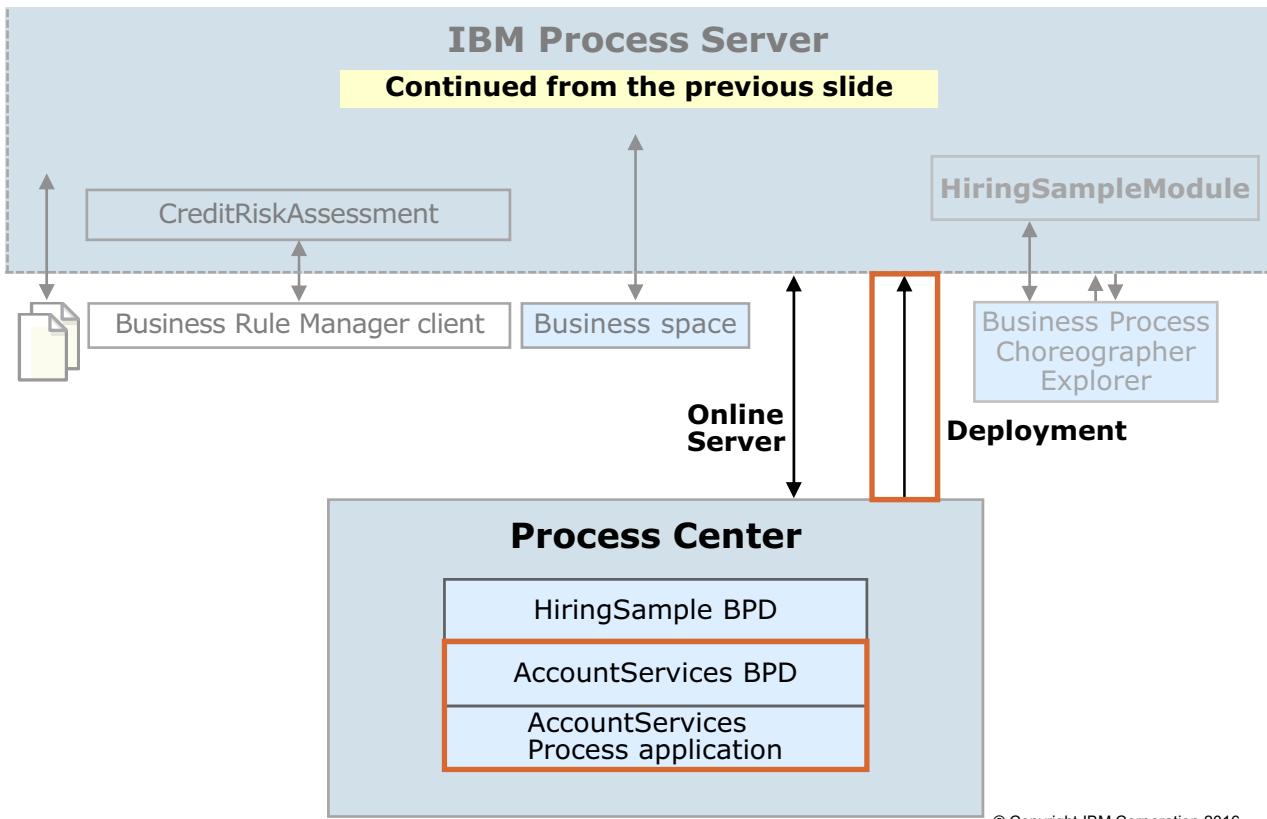
© Copyright IBM Corporation 2016

Figure 17-33. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 16



© Copyright IBM Corporation 2016

Figure 17-34. Course business scenario: What you work with in Exercise 16

WB8601.2

Notes:

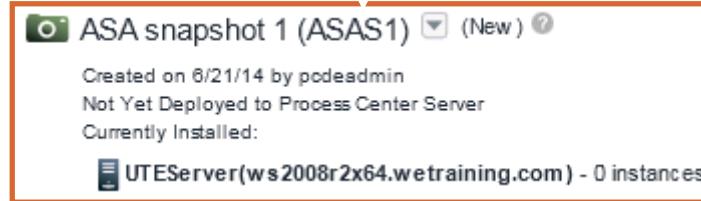
Components that are required for Exercise 16

Prebuilt components that are imported in this lab:

1. All components from the completed Exercise 13

New components that you create in this lab:

1. **Account Verification Services toolkit**
2. **AccountServicesapp** process application



© Copyright IBM Corporation 2016

Figure 17-35. Components that are required for Exercise 16

WB8601.2

Notes:

If `applicationDecision` is set to false during `FinalApplicationReview` (the application is declined) and the customer's `creditRisk` is HIGH, the application is routed through the "generate decline" component. If `applicationDecision` is set to false during `FinalApplicationReview` and the customer's `creditRisk` is MED (short for medium), the application is routed through the "special decline" component.

In this portion of the exercise, you implement the mediation flow for the `RouteRequest` component. The `RouteRequest` flow component contains the mediation logic that routes the application to the appropriate decline service.

The `RouteRequest` mediation flow consists of both a request flow and a response flow. In the flow, the `CustomerApplication` is routed to the appropriate decline service by a router mediation primitive. After processing, the response from the decline service is sent back to the `AccountVerification` process.

Unit 18. Advanced Integration services

What this unit is about

This unit is an overview of Advanced Integration services.

What you should be able to do

After completing this unit, you should be able to:

- Explain Advanced Integration services (AIS)
- Describe the business value of AIS
- Describe the steps to create an AIS

How you will check your progress

- Checkpoint questions
- Exercise

Unit objectives

- Explain Advanced Integration services (AIS)
- Describe the business value of AIS
- Describe the steps to create an AIS

© Copyright IBM Corporation 2016

Figure 18-1. Unit objectives

WB8601.2

Notes:



Topics

- AIS introduction
- Approaches in creation of AIS

© Copyright IBM Corporation 2016

Figure 18-2. Topics

WB8601.2

Notes:

18.1.AIS introduction

AIS introduction



© Copyright IBM Corporation 2016

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

10.1

Figure 18-3. AIS introduction

WB8601.2

Notes:

What is an Advanced Integration service?

- An Advanced Integration service is a collaboration between a business user who is working with IBM Process Designer and an integration developer who is working with IBM Integration Designer
- An Advanced Integration service is used to call a service that is implemented in IBM Integration Designer from a business process definition (BPD)
- SCA modules encapsulate Advanced Integration services
 - You can change services without affecting users of the service
- Advanced Integration services are available only with IBM Business Process Manager Advanced

© Copyright IBM Corporation 2016

Figure 18-4. What is an Advanced Integration service?

WB8601.2

Notes:

An Advanced Integration service is used to call a service implemented in IBM Integration Designer from a business process definition (BPD) (through a system task) or another service (through a nested service). An Advanced Integration service is a collaboration between a business user who is working with IBM Process Designer and an integration developer who is working with IBM Integration Designer.

When using IBM Integration Designer and IBM Process Designer together, collaborate before defining your Advanced Integration service.

Note: Advanced Integration services are available only with IBM Business Process Manager Advanced.

Business value of an Advanced Integration service

- Many business processes require a combination of human tasks and integrations with back-end systems
- Integrations might be complex in nature and require the orchestration of multiple service invocations
 - Higher level of transactional integrity and compensation
- Such complex integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL)
- Used by a process developer without the need to know much technical detail about the implementation of the technical service invoked
 - Process developer does not need to know Service Component Architecture (SCA), Java programming, transactional qualifiers, JDBC, and other implementation details
 - The only thing that the process developer needs to know is the I/O interface of the AIS and a bit of functional specification about its behavior

© Copyright IBM Corporation 2016

Figure 18-5. Business value of an Advanced Integration service

WB8601.2

Notes:

Most business processes require a combination of human tasks and integrations with back-end systems. Certain integrations might be complex in nature and require the orchestration of multiple service invocations, and a higher level of transactional integrity and compensation. Such integrations are best developed by using the IBM Integration Designer tool with constructs such as those offered by the Business Process Execution Language (BPEL). Process Designer has Advanced Integration services (AIS) and you can use it to invoke those IBM Integration Designer implementations.

For example, your business process might need a list of computer parts in your warehouses in Canada. Checking with an integration developer, you realize that a service is being built in Integration Designer to query the Canadian warehouses and return an inventory list of the computer parts available. You can create an Advanced Integration service that would use this Integration Designer service as an activity in your business process.

18.2.Approaches in creation of AIS

Approaches in creation of AIS



© Copyright IBM Corporation 2016

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

10.1

Figure 18-6. Approaches in creation of AIS

WB8601.2

Notes:

Two approaches for working with AIS

- Top-down approach
 - Create the AIS by using IBM Process Designer
 - This AIS that has the high-level details of the interface (inputs, outputs, and exceptions)
 - The more technical integration developer completes the implementation in IBM Integration Designer
- Bottom-up approach
 - Integration developers create a number of integration services ahead of time in IBM Integration Designer
 - Process developers discover the AIS and reuse them later on in IBM Process Designer

© Copyright IBM Corporation 2016

Figure 18-7. Two approaches for working with AIS

WB8601.2

Notes:

A business process can take advantage of integration services in two fundamental ways:

- The business process designer, by using Process Designer, can create an AIS that has the high-level details of the interface (inputs, outputs, and exceptions), and have the more technical integration developer complete the implementation in IBM Integration Designer. It is a **top-down** approach.
- Conversely, integration developers can create a number of integration services ahead of time, and the process developer can discover them and reuse them later on, for example, by packaging them in a Toolkit. It is called the **bottom-up** approach.

Top-down approach

- In a top-down approach, the Process Developer defines the inputs and outputs of the AIS and sends this information to the Process Center repository
- The integration developer receives the unimplemented Advanced Integration service and provides the implementation to be delivered back to the Process Center
- The Application Developer retrieves the implementation back in Process Designer for testing and playback
- The top-down approach works well when the service does not exist

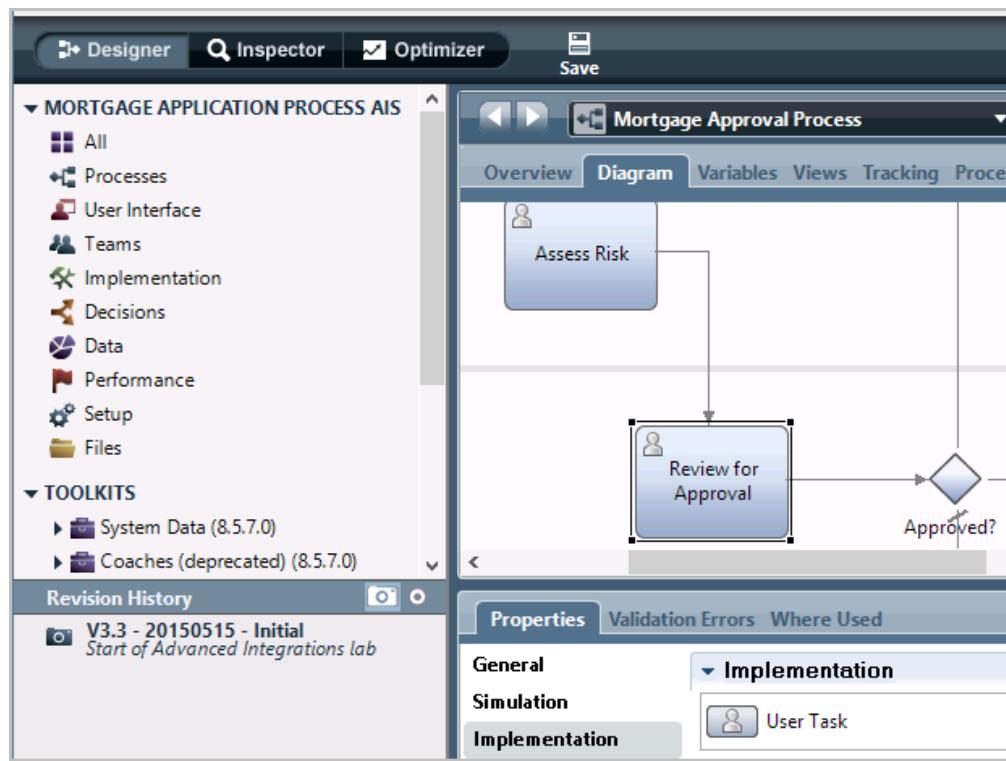
© Copyright IBM Corporation 2016

Figure 18-8. Top-down approach

WB8601.2

Notes:

Step 1: Create process application in Process Designer



© Copyright IBM Corporation 2016

Figure 18-9. Step 1: Create process application in Process Designer

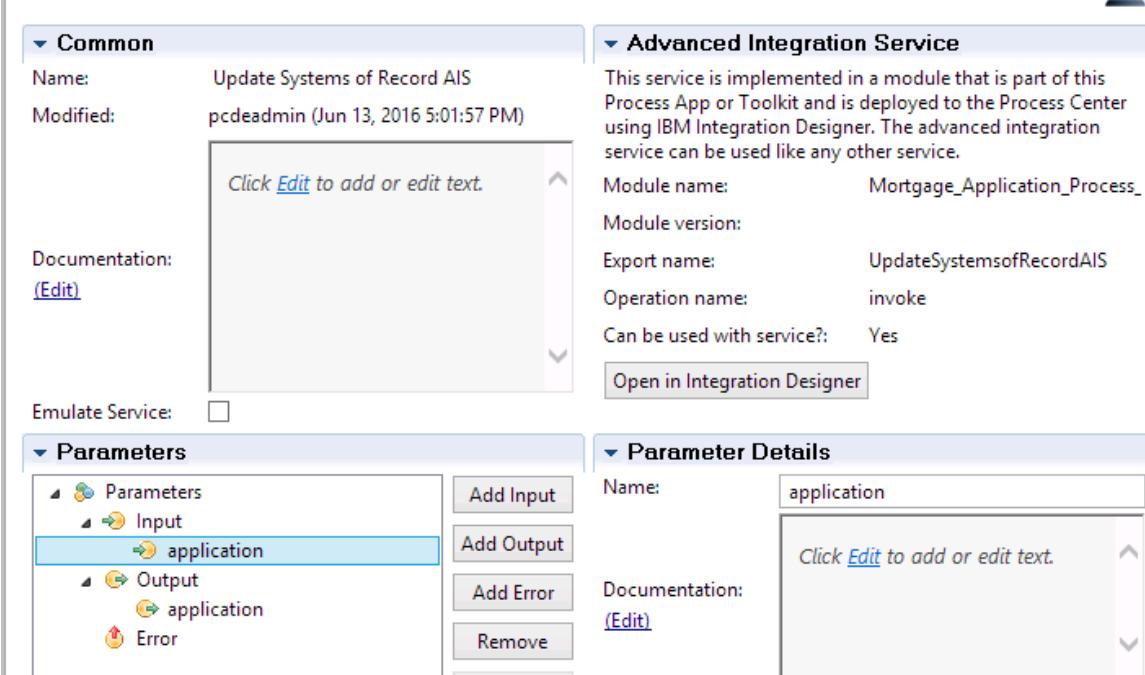
WB8601.2

Notes:

 WebSphere Education 

Step 2: Define inputs and outputs in AIS

Advanced Integration Service



Common		Advanced Integration Service	
Name:	Update Systems of Record AIS	Module name:	Mortgage_Application_Process_
Modified:	pcdeadmin (Jun 13, 2016 5:01:57 PM)	Module version:	
Documentation:	(Edit)	Export name:	UpdateSystemsofRecordAIS
Emulate Service:	<input type="checkbox"/>	Operation name:	invoke
		Can be used with service?:	Yes
		Open in Integration Designer	

Parameters		Parameter Details	
<ul style="list-style-type: none"> Parameters Input application Output application Error 		Name: application Documentation: (Edit) Click Edit to add or edit text.	
Add Input Add Output Add Error Remove Move Up			

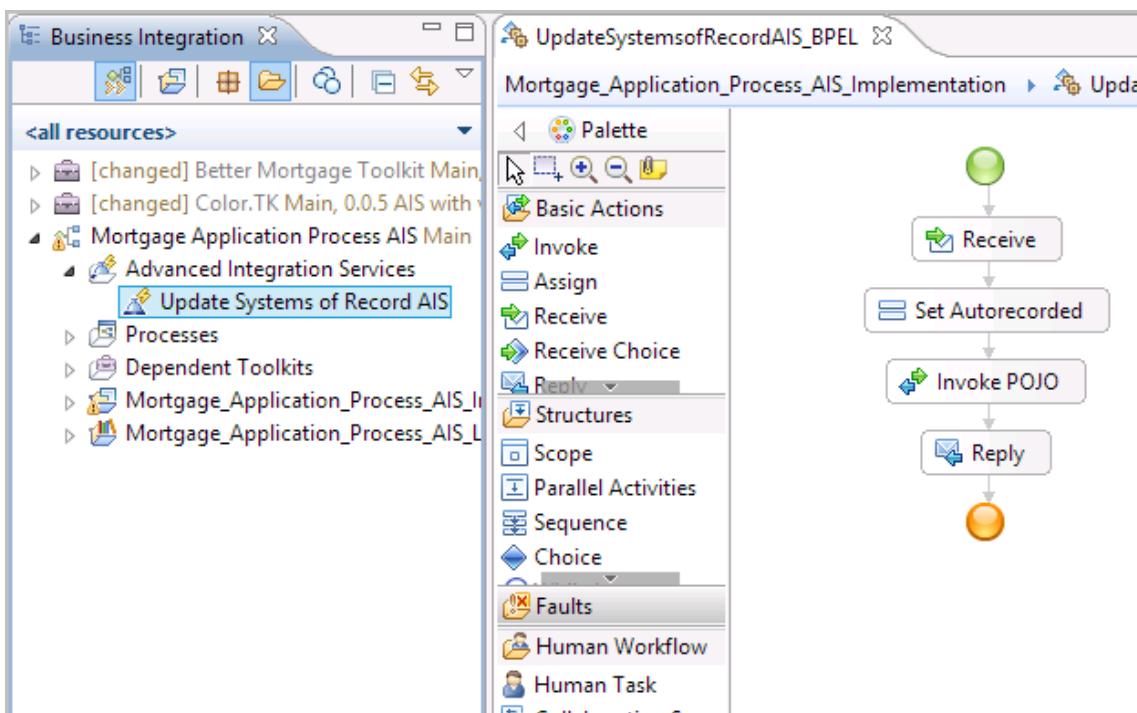
© Copyright IBM Corporation 2016

Figure 18-10. Step 2: Define inputs and outputs in AIS

WB8601.2

Notes:

Step 3: Implement the AIS



© Copyright IBM Corporation 2016

Figure 18-11. Step 3: Implement the AIS

WB8601.2

Notes:



Step 4: Refresh and publish changes to Process Center

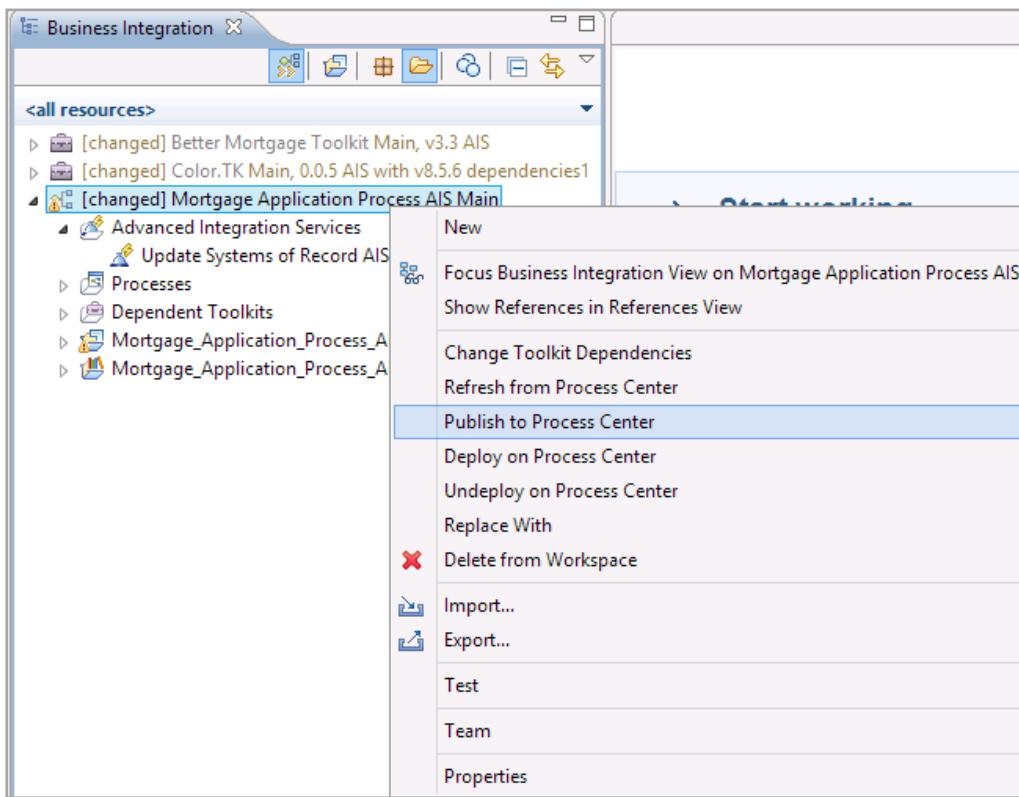


Figure 18-12. Step 4: Refresh and Publish changes to Process Center

WB8601.2

Notes:

Bottom-up approach

- Design and develop the Advanced Integration service in Integration Designer
- Deliver the interface and the implementation to IBM Process Designer
- Define a BPD that uses the Advanced Integration service
- The bottom-up approach works well when a mature SOA architecture with reusable services exists

© Copyright IBM Corporation 2016

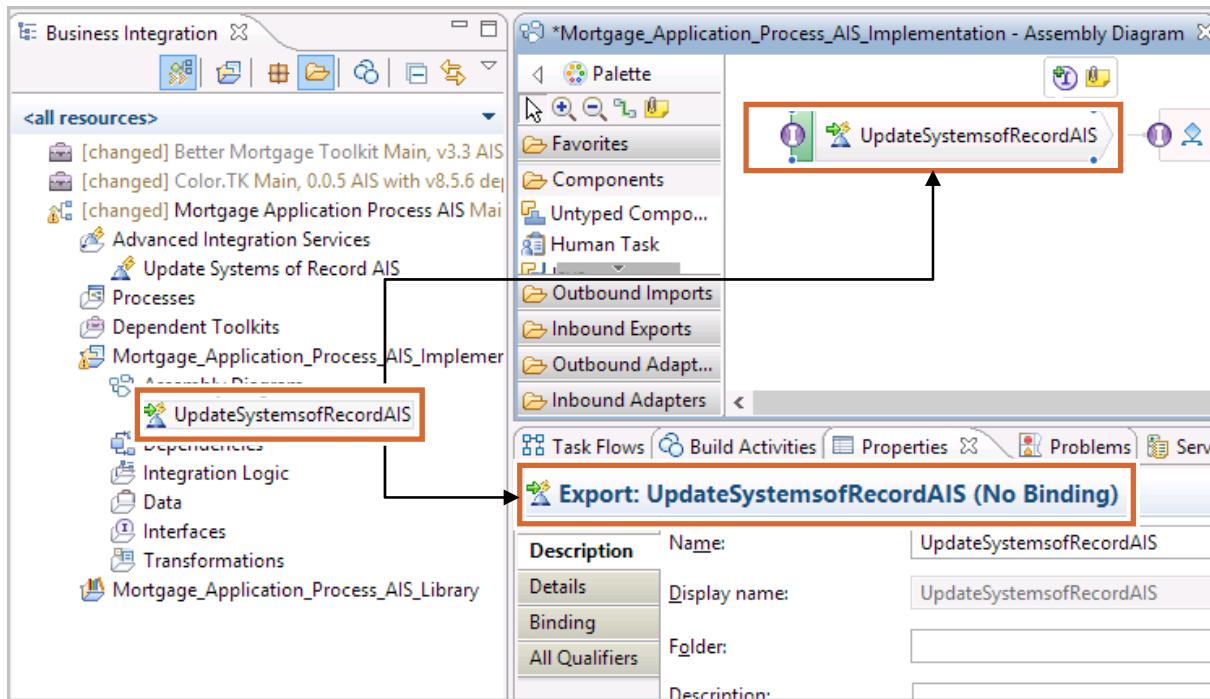
Figure 18-13. Bottom-up approach

WB8601.2

Notes:

WebSphere Education

Bottom-up approach, Step 1: Create export in Integration Designer



© Copyright IBM Corporation 2016

Figure 18-14. Step 1: Create export in Integration Designer

WB8601.2

Notes:



Bottom-up approach: Steps 2, 3, 4

- Step 2: Expose the export as AIS in Process Designer
- Step 3: Playback
- Step 4: Deploy the process application

© Copyright IBM Corporation 2016

Figure 18-15. Bottom-up approach

WB8601.2

Notes:

Unit summary

- Explain Advanced Integration services (AIS)
- Describe the business value of AIS
- Describe the steps to create an AIS

© Copyright IBM Corporation 2016

Figure 18-16. Unit summary

WB8601.2

Notes:

Checkpoint questions

1. True or False. Advanced Integration service is available in both the Standard and the Advanced editions of IBM Business Process Manager.
2. True or False. An Advanced Integration service is used to call a service that is implemented in IBM Integration Designer from a business process definition (BPD).
3. True or False. In the top-down approach of creating AIS, integration developers create a number of integration services ahead of time by using IBM Integration Designer.

© Copyright IBM Corporation 2016

Figure 18-17. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

- 1.
- 2.
- 3.



Checkpoint answers

1. False. AIS is available only in IBM BPM Advanced.
2. True.
3. False. In the bottom-up approach, integration developers create a number of integration services ahead of time by using IBM Integration Designer.

© Copyright IBM Corporation 2016

Figure 18-18. Checkpoint answers

WB8601.2

Notes:

Exercise 17



Implementing Advanced Integration services

© Copyright IBM Corporation 2016

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

10.1

Figure 18-19. Exercise 17

WB8601.2

Notes:

Exercise objectives

After completing this exercise, you should be able to:

- Create an AIS in IBM Process Designer
- Implement an AIS in IBM Integration Designer
- Unit test an AIS in IBM Integration Designer
- Invoke a BPEL process from a coach and BPD

© Copyright IBM Corporation 2016

Figure 18-20. Exercise objectives

WB8601.2

Notes:

Course business scenario: What you work with in Ex 17 (1 of 2)

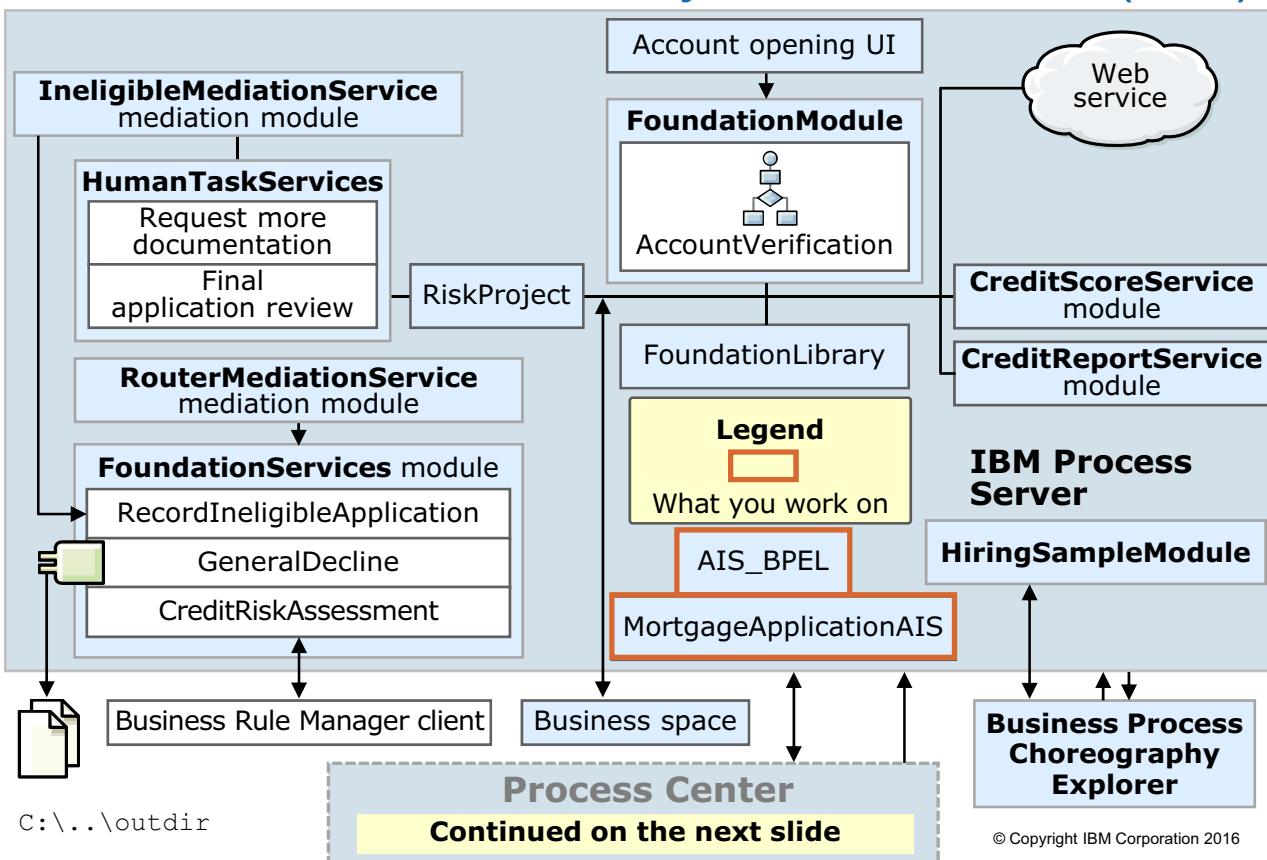
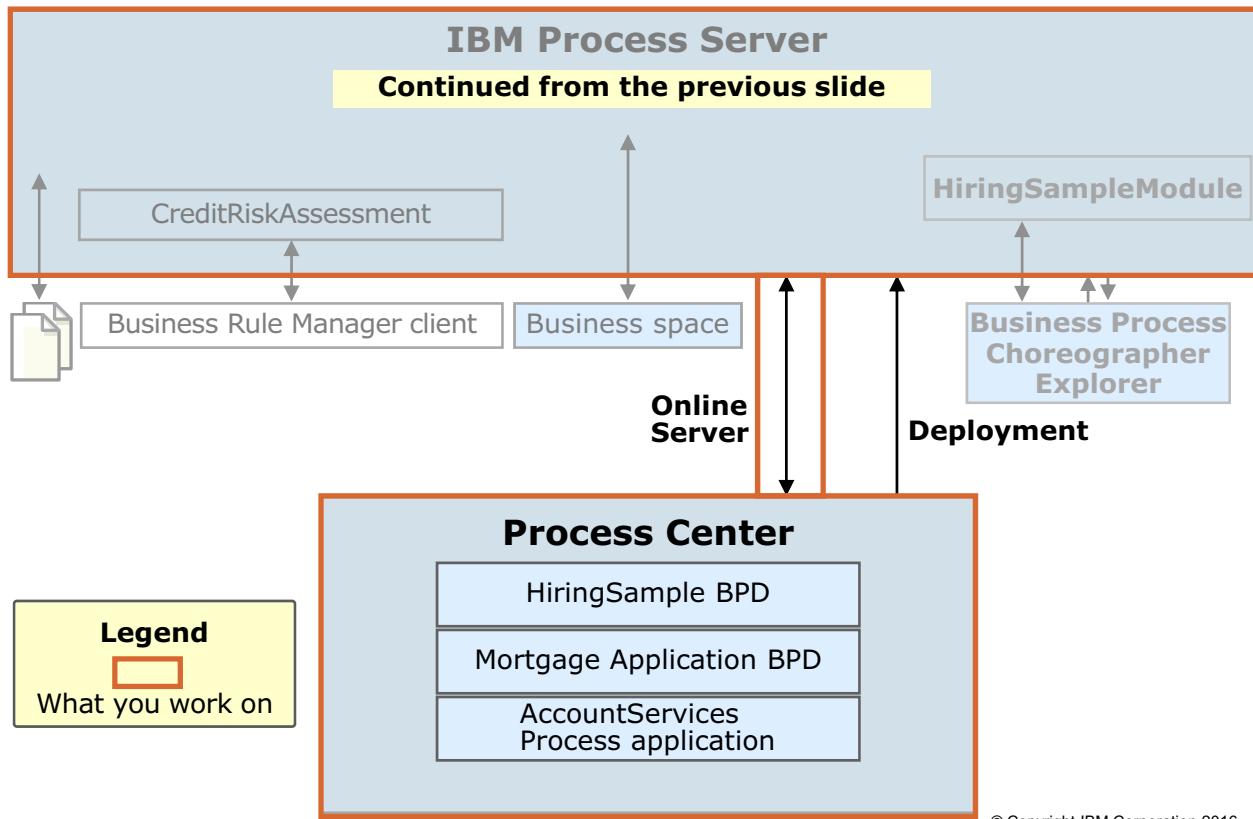


Figure 18-21. Course business scenario: What you work with in Exercise 17 (1 of 2)

WB8601.2

Notes:

Course business scenario: What you work with in Ex 17 (2 of 2)



© Copyright IBM Corporation 2016

Figure 18-22. Course business scenario: What you work with in Exercise 17 (2 of 2)

WB8601.2

Notes:

Unit 19. Course summary

What this unit is about

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

What you should be able to do

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study



Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

© Copyright IBM Corporation 2016

Figure 19-1. Unit objectives

WB8601.2

Notes:

Course learning objectives (1 of 3)

After completing this course, you should be able to:

- Describe the purpose and business value of a service-oriented architecture
- Describe the purpose and business value of the tools included in IBM Business Process Manager Advanced V8.5.7: IBM Process Designer, IBM Integration Designer, IBM Process Server, IBM Process Center, IBM Process Portal, and Business Space
- Identify and describe the features that are available in IBM Process Designer, IBM Integration Designer, and the Service Component Architecture
- Describe the structure of modules and libraries
- Create business objects, shared interfaces, and business calendars
- Model a complex business process diagram in IBM Process Designer
- Create a complex business process that includes basic and structured WS-BPEL activities in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 19-2. Course learning objectives (1 of 3)

WB8601.2

Notes:

Course learning objectives (2 of 3)

After completing this course, you should be able to:

- Use SCA bindings to communicate with web services
- Use WebSphere Java EE Connector Architecture (JCA) adapters in applications
- Implement data maps to transform business data
- Implement mediation modules to route and transform messages
- Implement business rules and use the Business Rules Manager web client to interact with business rules at run time
- Create a business space by using widgets and templates that are available for IBM Business Process Manager Advanced V8.5.7
- Create and run component test projects in IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 19-3. Course learning objectives (2 of 3)

WB8601.2

Notes:



Course learning objectives (3 of 3)

After completing this course, you should be able to:

- Bring the UTE Process Server online and connect with the Process Center
- Explore the IBM Process Center repository
- Create and implement an Advanced Integration service (AIS) in IBM Process Designer and IBM Integration Designer

© Copyright IBM Corporation 2016

Figure 19-4. Course learning objectives (3 of 3)

WB8601.2

Notes:

 WebSphere Education 

Earn an IBM Badge

- After completing this course, you are almost 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/zz-en?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



© Copyright IBM Corporation 2016

Figure 19-5. Earn an IBM Badge

WB8601.2

Notes:



To learn more on the subject

- IBM Training website:
 - www.ibm.com/training
- 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
- BlueWorks Live
 - <https://www.blueworkslive.com/>
- IBM developerWorks for Business Process Management
 - <http://www.ibm.com/developerworks/websphere/zones/bpm/>

© Copyright IBM Corporation 2016

Figure 19-6. To learn more on the subject

WB8601.2

Notes:



Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

© Copyright IBM Corporation 2016

Figure 19-7. Unit summary

WB8601.2

Notes:

Appendix A. IBM BPM on Cloud

What this unit is about

In this unit, you learn about BPM on Cloud.



Introduction to IBM BPM on Cloud

- Enterprise-grade BPM cloud service for development, testing, and production
- Cloud-based, collaborative, 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

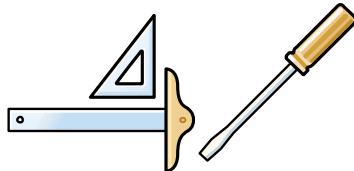
© Copyright IBM Corporation 2016

Figure A-1. Introduction to IBM BPM on Cloud

WB8601.2

Notes:

Accelerate decision management solution deployment with IBM BPM on Cloud



Build-it-yourself

- Separately procure and install software, service, and hardware
- Sign multiple contracts
- Requires capital investment to procure software, hardware, and implementation services
- Ongoing management, maintenance, and upgrades

IBM BPM on Cloud

- Integrated, fast, and flexible
- Get started right away
- One business solution, one contract, and one subscription price
- Minimal capital investment
- Scale when needed
- Reduce time and effort that are needed for maintenance
- Keep up-to-date with the current releases

© Copyright IBM Corporation 2016

Figure A-2. Accelerate decision management solution deployment with IBM BPM on Cloud

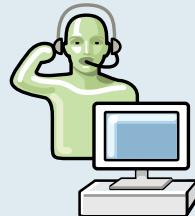
WB8601.2

Notes:

BPM on Cloud customer focus: Manage and automate decisions

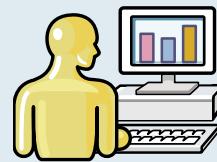
IBM manages:

- Uptime
- Monitoring
- Backup
- High availability
- Disaster recovery
- Updates
- Maintenance



Customers manage:

- Application development
- Application integration
- Application support



© Copyright IBM Corporation 2016

Figure A-3. BPM on Cloud customer focus: Manage and automate decisions

WB8601.2

Notes:

IBM BPM on Cloud: Three runtime environments

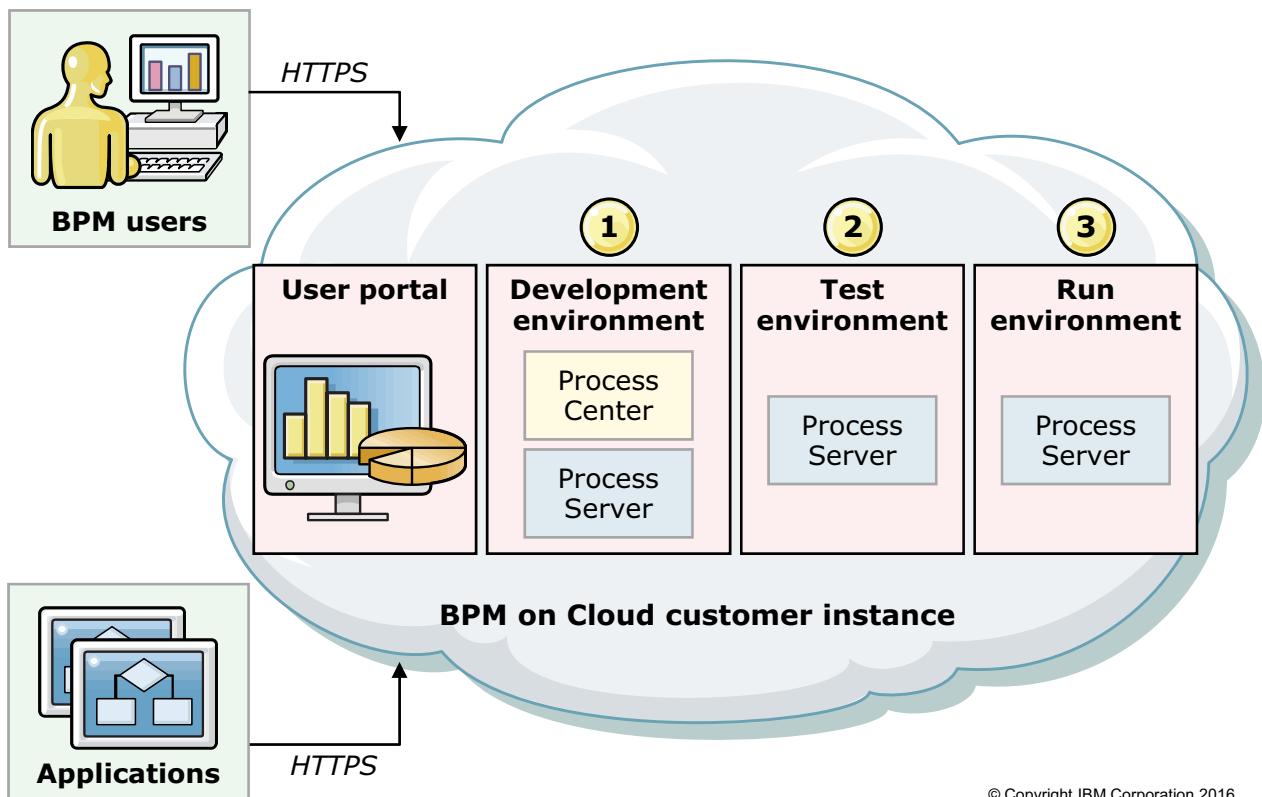


Figure A-4. IBM BPM on Cloud: Three runtime environments

WB8601.2

Notes:



IBM BPM on Cloud free trial

- Free 30-day trial for IBM BPM on Cloud is available
- Go to the following website and click **Try for free** to sign up:

<https://www.bpm.ibmcloud.com/#home>

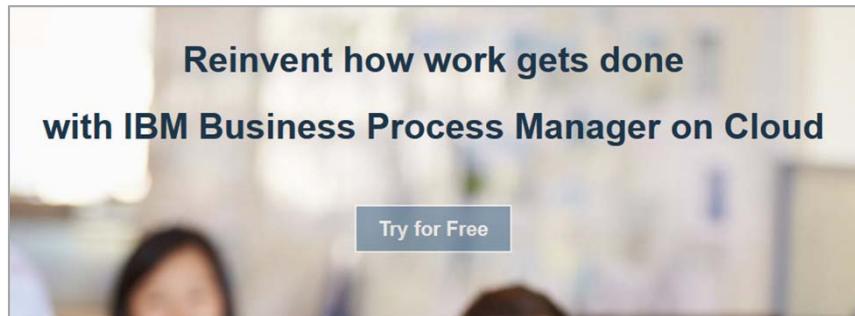


Figure A-5. IBM BPM on Cloud free trial

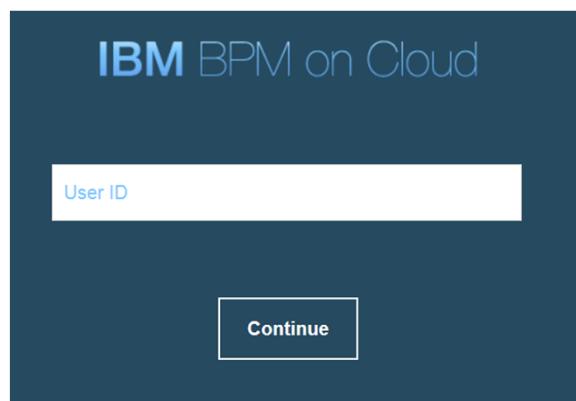
WB8601.2

Notes:



Activating access and logging in to IBM BPM on Cloud

- Welcome email includes the following information:
 - Link to activate BPM on Cloud access
 - Link to BPM on Cloud instance
- Activation link is tied to a specific email
- After activating access, you can log in to your BPM on Cloud instance



© Copyright IBM Corporation 2016

Figure A-6. Activating access and logging in to IBM BPM on Cloud

WB8601.2

Notes:



IBM

IBM BPM on Cloud user portal (1 of 3)

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

 - Development

Development Environment

Process Center 	REST UI 	Process Portal 	Process Admin Console
Install and run process applications, store performance data, and manage running instances of process applications on the Process Center servers. Launch More info Available Downloads (2) IBM® Process Designer IBM® Integration Designer	Prototype IBM BPM REST resources and their associated parameters Launch More info	Collaborate on tasks and view the performance of individuals, teams, and processes on dashboards. Launch More info	Manage the Process Center server and the process servers in your runtime environments. Launch More info
Tech Preview: Responsive Federated Portal 	Business Process Choreographer Explorer 	Business Rules Manager 	
Technical demonstrations of Responsive Federated Portal and Responsive Coach Toolkit as a sample Launch More info	Monitor and manage BPEL processes. The BPC Explorer provides a number of views that show process and task metadata Launch More info	Manage business rules Launch More info	

© Copyright IBM Corporation 2016

Figure A-7. IBM BPM on Cloud user portal (1 of 3)

WB8601.2

Notes:



IBM BPM on Cloud user portal (2 of 3)

- Access from home page to an array of tools in three environments:
 2. Test

Test Environment

Process Portal  Collaborate on tasks and view the performance of individuals, teams, and processes on dashboards. Launch More info	Process Admin Console  Manage the Process Center server and the process servers in your runtime environments. Launch More info	Tech Preview: Responsive Federated Portal  Technical demonstrations of Responsive Federated Portal and Responsive Coach Toolkit as a sample Launch More info
Business Rules Manager  Manage business rules Launch More info	Business Process Choreographer Explorer  Monitor and manage BPEL processes. The BPC Explorer provides a number of views that show process and task metadata Launch More info	

© Copyright IBM Corporation 2016

Figure A-8. IBM BPM on Cloud user portal (2 of 3)

WB8601.2

Notes:

IBM BPM on Cloud user portal (3 of 3)

- Access from home page to an array of tools in three environments:
- 3. Production Runtime Operating

Process Runtime Operating Environment

Process Portal  Collaborate on tasks and view the performance of individuals, teams, and processes on dashboards. Launch More info	Process Admin Console  Manage the Process Center server and the process servers in your runtime environments. Launch More info	Tech Preview: Responsive Federated Portal  Technical demonstrations of Responsive Federated Portal and Responsive Coach Toolkit as a sample Launch More info
Business Rules Manager  Manage business rules Launch More info	Business Process Choreographer Explorer  Monitor and manage BPEL processes. The BPC Explorer provides a number of views that show process and task metadata Launch More info	

© Copyright IBM Corporation 2016

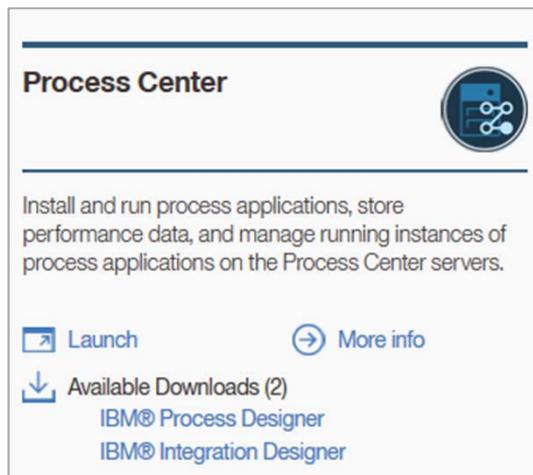
Figure A-9. IBM BPM on Cloud user portal (3 of 3)

WB8601.2

Notes:

Using the IBM Process Designer (1 of 3)

- Download a version of Process Designer that is configured for use with IBM BPM on Cloud
- Start Process Designer by double-clicking `eclipse.exe`



© Copyright IBM Corporation 2016

Figure A-10. Using the IBM Process Designer (1 of 3)

WB8601.2

Notes:

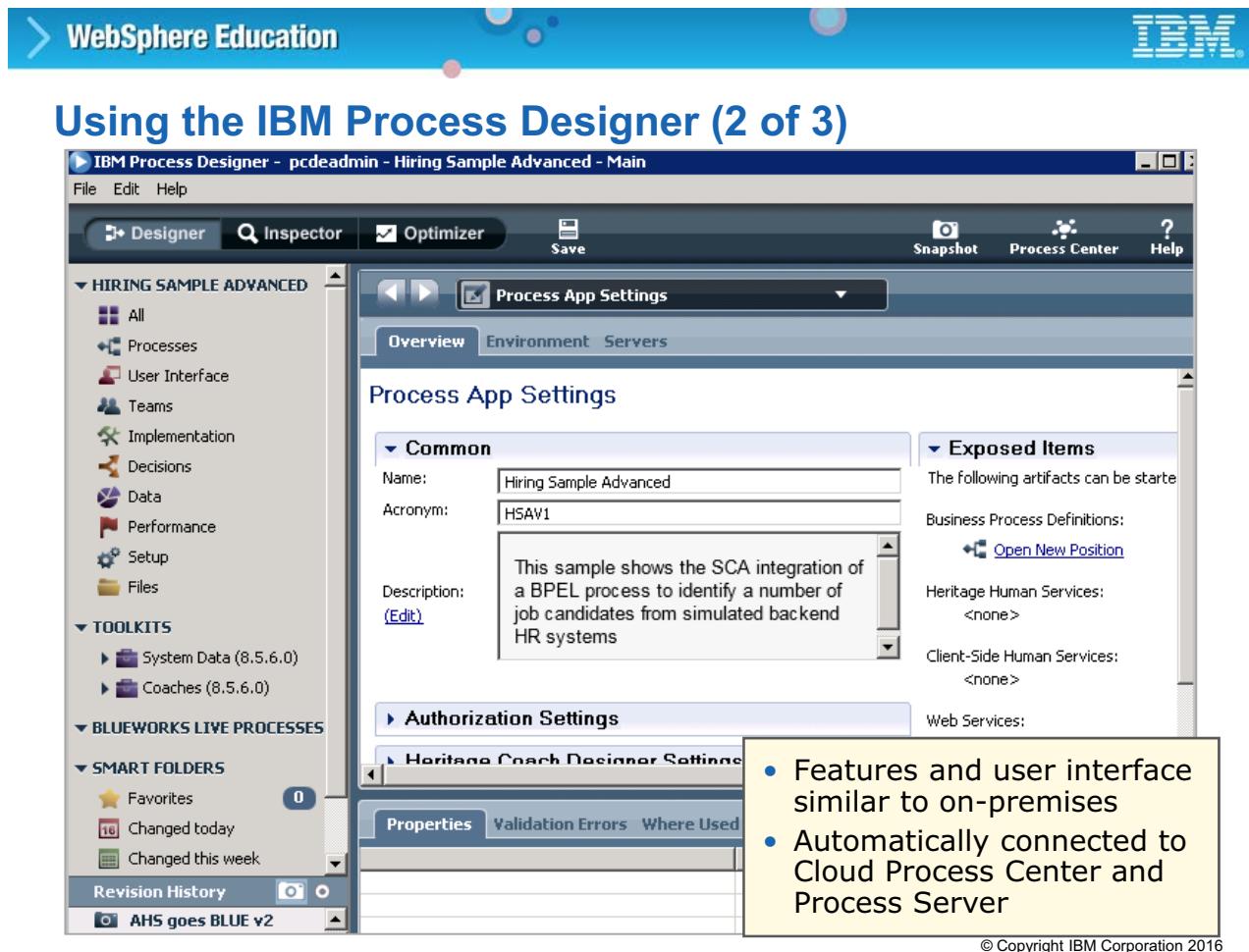


Figure A-11. Using the IBM Process Designer (2 of 3)

WB8601.2

Notes:



Using the IBM Process Designer (3 of 3)

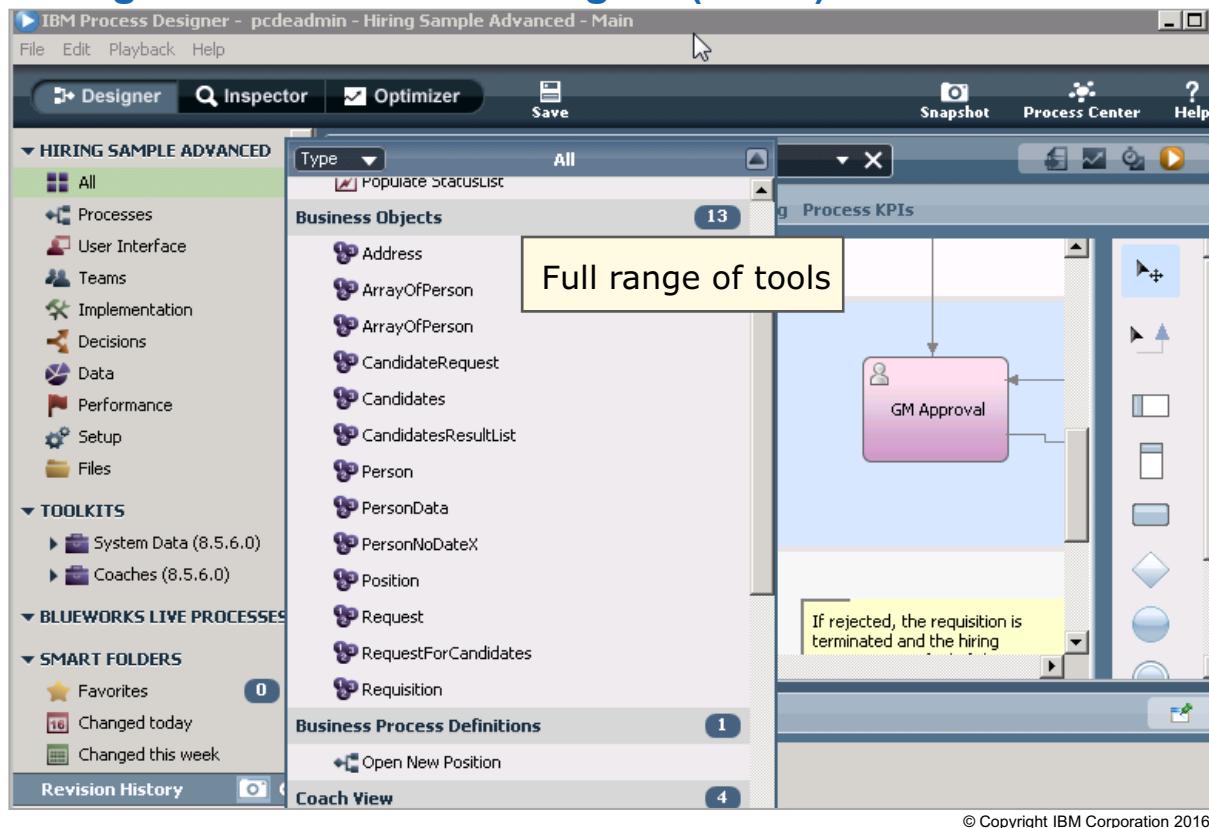


Figure A-12. Using the IBM Process Designer (3 of 3)

WB8601.2

Notes:

The screenshot shows the IBM Process Center web interface. At the top, there's a blue header with the "WebSphere Education" logo on the left and the "IBM" logo on the right. Below the header, the main title "Using the IBM Process Center (1 of 3)" is displayed in a large, bold, blue font. Underneath the title, there's a bullet point list: "• Familiar interface". The main content area is a list of process applications. The first item is "Hiring Sample Advanced (HSAV)" with a yellow star icon and a question mark. Below it, a small note says "Last updated on 2/24/16 by te@us.ibm.com". The second item is "Account Verification Skeleton (AVS)" with a yellow star icon and a question mark. Below it, a small note says "Last updated on 2/4/16 by te@us.ibm.com". To the right of the list, there's a sidebar with several links: "Create New Process App", "Import Process App", "Download Process Designer", "Download MobileFirst Adapter", and "Launch Getting Started". At the bottom right of the interface, it says "IBM | Process Center".

© Copyright IBM Corporation 2016

Figure A-13. Using the IBM Process Center (1 of 3)

WB8601.2

Notes:



Using the IBM Process Center (2 of 3)

- Cloud-based Case Designer

The screenshot shows the IBM Process Center Designer Inspector interface. On the left, a sidebar titled 'DESIGNER INSPECTOR' lists 'Account Verification Skeleton' under 'Cases'. The main panel displays 'MyHiringCase' with tabs for 'Overview', 'Activities', 'Variables', 'Folders', and 'Views'. The 'Overview' tab is selected. It shows the following details:

- Common** section: Name: MyHiringCase, Modified: Feb 4, 2016 1:57:33 PM, Documentation: (empty text area with rich text editor icons).
- Advanced** section: Instance name: MyHiringCase + tw.system.process.
- Exposing** section: Expose to start: <none>, Select... button; Expose business data: <none>, Select... button; Expose performance metrics: <none>, Select... button.
- Team** section: Instance owners: All Users, System Data, Select... button.
- Symbolic name: AVS_MyHiringCase_105.
- Starting Document** section: Starting document type: <none>, Select..., New..., X buttons.

© Copyright IBM Corporation 2016

Figure A-14. Using the IBM Process Center (2 of 3)

WB8601.2

Notes:



Using the IBM Process Center (3 of 3)

- Snapshots, export, install familiar

A screenshot of the IBM Process Center interface. The top navigation bar includes "Process Apps", "Toolkits", "Servers", and "Admin". The "Process Apps" tab is selected, showing a list of applications: "Account Verification Skeleton (AVS)" (marked as "Current" and last changed on 2/4/16), and "Skeleton Rewired (SR) (New)" (created on 2/4/16 by dsh, Not Yet Installed to Process Server). A modal dialog titled "Install Snapshot to Server" is open, prompting the user to "Select a server to install snapshot Skeleton Rewired to:". It lists two servers: "TEST ProcessServer (10.76.89.120)" (TEST - Status: Connected) with a checked checkbox, and "RUN ProcessServer (10.76.89.121)" (PRODUCTION - Status: Connected) with an unchecked checkbox. A "Sort Snapshots By" dropdown menu is visible at the top right of the dialog.

© Copyright IBM Corporation 2016

Figure A-15. Using the IBM Process Center (3 of 3)

WB8601.2

Notes:

WebSphere Education

Using the IBM Process Portal

My Work

My Tasks

Overdue (9)

Step: Enter App
Acquire Customer For L

Step: Submit job requisition

Job requisition data

Requester	Requested job position	Requested job start date and location
* Request number 1140	* Employment status	* Planned date of job start 2/4/2016
* Hiring Manager Roland Peisl	* Department	* Location
Number of employees required 1		

Position data

* Position type	* Job title Head of Product Development
-----------------	--------------------------------------------

Make your decision

Next

Users shown familiar Work, Tasks, Coaches interface

© Copyright IBM Corporation 2016

Figure A-16. Using the IBM Process Portal

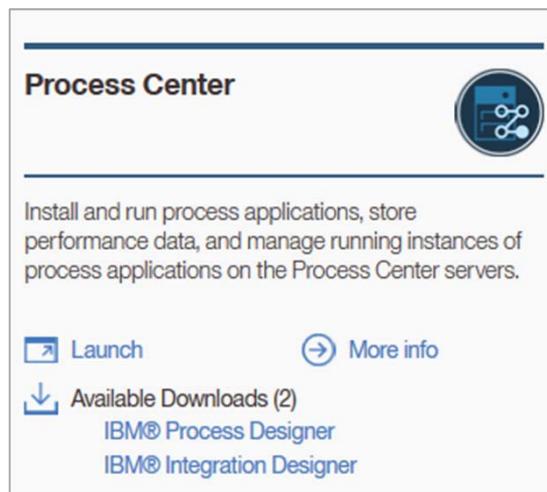
WB8601.2

Notes:



Using the IBM Integration Designer

- Download a version of the IBM Integration Designer from the BPM cloud:
 - URL for connecting to IBM BPM on Cloud provided
- Start IBM Integration Designer on local workstation as usual



© Copyright IBM Corporation 2016

Figure A-17. Using the IBM Integration Designer

WB8601.2

Notes:



Finding help for IBM BPM on Cloud

- IBM Knowledge Center for IBM BPM on Cloud

http://www.ibm.com/support/knowledgecenter/SS964W/ditamaps/product_welcome_oncloud.html

- Complete product documentation for IBM BPM on Cloud, including a “Getting Started” tutorial
- IBM BPM on Cloud user portal also has direct links to the documentation

- IBM BPM Support Portal

<https://www.ibm.com/support/entry/portal/product/websphere>

- Support Portal provides tools and resources for help with IBM Business Process Manager
- Open service requests, view fix lists, access community resources, and more

© Copyright IBM Corporation 2016

Figure A-18. Finding help for IBM BPM on Cloud

WB8601.2

Notes:

Appendix B. Supplemental information about mediations and mediation primitives

What this unit is about

This unit describes the mediation and mediation primitives available in IBM Process Server.



Callout node: Properties (1 of 2)

Callout : dealerPartQueryWMQ : IF_DealerPartQueryWMQPartner	
Description	Reference name: <input type="text" value="IF_DealerPartQueryWMQPartner"/>
Terminal	Operation name: <input type="text" value="dealerPartQueryWMQ"/>
Details	<input checked="" type="checkbox"/> Use dynamic endpoint if set in the message header <input type="checkbox"/> Require mediation flow to wait for service response when the flow c Invocation Style: <input type="text" value="Default"/>
Retry	Async timeout (seconds): <input type="text" value="5"/>
Promotable Properties	

- Dynamic endpoint: More in subsequent unit
- Timeout: Wait for responses when invoking an asynchronous deferred response
- Invocation style: Determines whether service is invoked synchronously or asynchronously

© Copyright IBM Corporation 2016

Figure B-1. Callout node: Properties (1 of 2)

WB8601.2

Notes:

Callout node: Properties (2 of 2)

Retry parameters:

- Retry condition: If and when a retry is attempted, it specifies:

- Never
- On any fault
- On a modeled fault
- On an unmodeled fault

Callout : dealerPartQueryWMQ : IF_DealerPartQueryWMQPartner	
Description	Retry on:
Terminal	<input type="checkbox"/> Never
Details	Retry count:
Retry	<input type="checkbox"/> 0
	Retry delay (seconds): <input type="checkbox"/> 0
Promotable Properties	<input checked="" type="checkbox"/> Try alternate endpoints

- Count: How many times a retry is attempted
- Delay: Seconds to wait between retry attempts
 - 0 is valid
- Alternate endpoints: If true, alternate endpoints that are found in SMO headers are used
 - “Use dynamic endpoints” must also be on

© Copyright IBM Corporation 2016

Figure B-2. Callout node: Properties (2 of 2)

WB8601.2

Notes:

Data Handler primitive (1 of 2)

- 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
 - This action allows the transformation of the physical format into the specific business object to complete in the mediation flow component, instead of the export and import
 - The Data Handler primitive can change any part of the service message object
 - You specify:
 - The data handler configuration for the primitive to use
 - The output message field refinements
 - XPath expressions that reference the source and target elements
 - The action to perform (convert from the native data format to a business object, or convert from a business object to the native data format)



Data Handler

© Copyright IBM Corporation 2016

Figure B-3. Data Handler primitive (1 of 2)

WB8601.2

Notes:

Data Handler primitive (2 of 2)

- Several data handlers are provided as part of the product, or you can write your own
- Predefined data handlers include:
 - Comma-delimited
 - Fixed-width
 - Handled by WebSphere Transformation Extender
 - Handled by WebSphere Transformation Extender invoker
 - JAX-WS-based EJB bean
 - JAX-based EJB bean
 - JSON (UTF8JSON)
 - SOAP
 - XML
 - XML (UTF8XML)
 - Atom feed

Usage examples:

- Transforming a section of the input message from one defined structure to another defined structure
 - For example, if the SMO includes a comma-delimited string value, you want to parse this value into a specific business object
- Altering the message type
 - For example, suppose that a JMS export is configured to use a JMS basic-typed data binding and within the mediation module
 - The integration developer decides that the content should be inflated to a specific business object structure

© Copyright IBM Corporation 2016

Figure B-4. Data Handler primitive (2 of 2)

WB8601.2

Notes:

WebSphere Education

Set Message Type primitive

- You can use it to treat weakly typed message fields as though they are strongly typed
 - You can use this behavior to overlay message fields with more detailed structures, and then use the more detailed structures in other mediation primitives



Set Message Type

- You specify:
 - Which fields in the message have more specific typing information to refine them, and the data types to use for them
 - A Reset message type flag: When “true”, causes the current primitive to “forget” any message field refinements that occurred in the previous Set Message Type primitives in the mediation
 - Validate input property: If set to “true,” and the input message is invalid (does not match its schema), a runtime exception occurs
 - At run time, the Set Message Type primitive does not affect the real structure or content of a message

© Copyright IBM Corporation 2016

Figure B-5. Set Message Type primitive

WB8601.2

Notes:

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 primitive:

- 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`
 - Must return a *DataObject*
 - **Caution:** Default code is generated with “`returns null;`” and you must change this code when you configure the Custom Mediation primitive

Any exception that is thrown by the custom Java code

- Thrown as is: It is not wrapped by any *MediationXXXXException* type
- If the fail terminal is wired, it is fired

© Copyright IBM Corporation 2016

Figure B-6. Custom Mediation: Error handling

WB8601.2

Notes:

WebSphere Education

IBM

Service Invoke primitive: Properties

Reference name:	IF_DealerPartQueryWMQPartner	<i>The timeout terminal fires when this threshold is exceeded</i>										
Operation name:	dealerPartQueryWMQ											
<input checked="" type="checkbox"/> Use dynamic endpoint if set in the message header												
Async timeout (seconds):	<input type="text"/>											
<input type="checkbox"/> Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback.												
Invocation Style:	Default											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Description</td> <td style="padding: 5px;"><input type="button" value="Retry on: Any fault"/></td> </tr> <tr> <td style="padding: 5px;">Terminal</td> <td style="padding: 5px;"><input type="text" value="0"/></td> </tr> <tr> <td style="padding: 5px;">Details</td> <td style="padding: 5px;"><input type="text" value="0"/></td> </tr> <tr> <td style="padding: 5px;">Retry</td> <td style="padding: 5px;"><input checked="" type="checkbox"/> Try alternate endpoints</td> </tr> <tr> <td style="padding: 5px;">Promotable Properties</td> <td style="padding: 5px;"></td> </tr> </table>			Description	<input type="button" value="Retry on: Any fault"/>	Terminal	<input type="text" value="0"/>	Details	<input type="text" value="0"/>	Retry	<input checked="" type="checkbox"/> Try alternate endpoints	Promotable Properties	
Description	<input type="button" value="Retry on: Any fault"/>											
Terminal	<input type="text" value="0"/>											
Details	<input type="text" value="0"/>											
Retry	<input checked="" type="checkbox"/> Try alternate endpoints											
Promotable Properties												

Retry options:

- Never
- On any fault
- On a modeled fault
- On an unmodeled fault

Can specify the number of retries, and the interval between each retry

© Copyright IBM Corporation 2016

Figure B-7. Service Invoke primitive: Properties

WB8601.2

Notes:

Service message object (SMO) contexts

- The SMO contains system-defined context objects, and user-defined context objects
- Four specific 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 the request and response flows
 - Shared context:** One for request flow, and a separate one for response flow (one shared context per thread, per flow)

Correlation context:	<Context object is not specified>
Transient context:	{http://DPN_DealerCommonLibrary}DealerPartQueryDEFResult
Shared context:	{http://DPN_DealerCommonLibrary}DealerPartQueryResponsesArray

© Copyright IBM Corporation 2016

Figure B-8. Service message object (SMO) contexts

WB8601.2

Notes:

Fan Out and Fan In primitives: Overview

- Enable 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

© Copyright IBM Corporation 2016

Figure B-9. Fan Out and Fan In primitives: Overview

WB8601.2

Notes:

Fan Out primitive (1 of 2)

Operating modes:

- **Default mode:**

- Out terminal is fired once for each input message, for each output wire that is connected to it
- Same message is propagated to each output wire, in succession
- Output message is identical to the input message



- **Iterate mode:**

- Output terminal is fired once for each occurrence of a repeating element in the incoming message (you specify the element with an XPath expression)
- 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 the Fan In primitive is encountered (assuming a decision point is reached); or a callout is encountered; or a stop, fail, or unwired out terminal is reached
- The next message (next occurrence of the repeated element) is then fired through the out terminal

© Copyright IBM Corporation 2016

Figure B-10. Fan Out primitive (1 of 2)

WB8601.2

Notes:



Fan Out primitive (2 of 2)

- Out terminal:
 - In *default* mode, an unmodified message is routed through the out terminal
 - In *iterate* mode, the out terminal propagates the input message, but is fired once 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
- noOccurrences terminal:
 - Used only in *iterate* mode
 - Used when the input message does not contain any occurrences in the repeating element (as if the primitive is set to the *default* mode)



© Copyright IBM Corporation 2016

Figure B-11. Fan Out primitive (2 of 2)

WB8601.2

Notes:

Fan In primitive (1 of 2)

- Is used to aggregate messages that are propagated from a corresponding Fan Out primitive
- Receives messages until a *decision point* is reached; then, the message that is received last 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



© Copyright IBM Corporation 2016

Figure B-12. Fan In primitive (1 of 2)

WB8601.2

Notes:

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 a Fan In primitive is added to the canvas
- In addition to input, output, and fail terminals, a Fan In primitive also has:
 - Stop terminal (input): Causes the incomplete terminal to fire, which stops the associated Fan Out primitive from sending any more messages
 - Incomplete terminal (output): Works with the stop terminal as stated previously; is also fired when a timeout occurs
- You specify:
 - Decision type (count, XPath, and iterate)
 - Timeout value: If the decision point is not reached before this value, it causes an incomplete terminal to fire

© Copyright IBM Corporation 2016

Figure B-13. Fan In primitive (2 of 2)

WB8601.2

Notes:

Aggregation and shared context

Problem: How to aggregate data

- Each time the Fan Out output terminal is fired, an SMO instance is created
- Each new SMO instance is a deep copy
- How do you 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
 - Single memory area
 - Not deep copied with each SMO instance
- A business object defines it (similar to transient and correlation contexts)

Shared context usage

- Flows between Fan Out and Fan In set values to aggregate into the shared context
- After the Fan In completes, subsequent primitives are used to build an aggregated message

No parallelism is allowed between a Fan Out and a Fan In

- Service Invoke primitives have the force sync property automatically set

© Copyright IBM Corporation 2016

Figure B-14. Aggregation and shared context

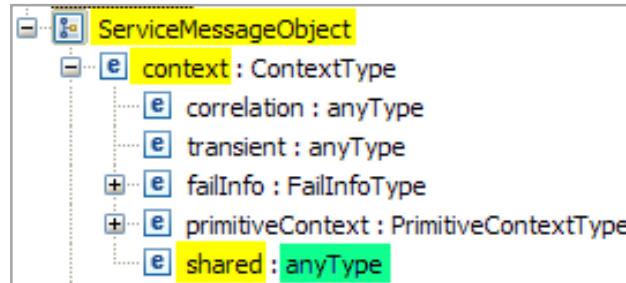
WB8601.2

Notes:

WebSphere Education

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 primitives
- All SMO instances share a single memory area
- Defined by a business object that is configured on the input node



Correlation context:	<Context object is not specified>
Transient context:	{http://DPN_DealerCommonLibrary}DealerPartQueryDEFResult
Shared context:	{http://DPN_DealerCommonLibrary}DealerPartQueryResponsesArray

© Copyright IBM Corporation 2016

Figure B-15. SMO structure: Context – shared

WB8601.2

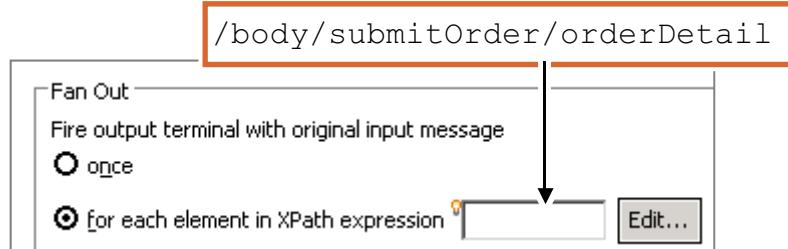
Notes:

Fan Out example: Repeating group

```

<body>
  <submitOrder>
    <orderDetail>
      <Part>
        <ItemID>WJX00192</ItemID>
        <Quantity>10</Quantity>
      </Part>
      <Part>
        <ItemID>PJR33029</ItemID>
        <Quantity>4</Quantity>
      </Part>
      <Part>
        <ItemID>CFX80010</ItemID>
        <Quantity>16</Quantity>
      </Part>
    </orderDetail>
  </submitOrder>
</body>

```



© Copyright IBM Corporation 2016

Figure B-16. Fan Out example: Repeating group

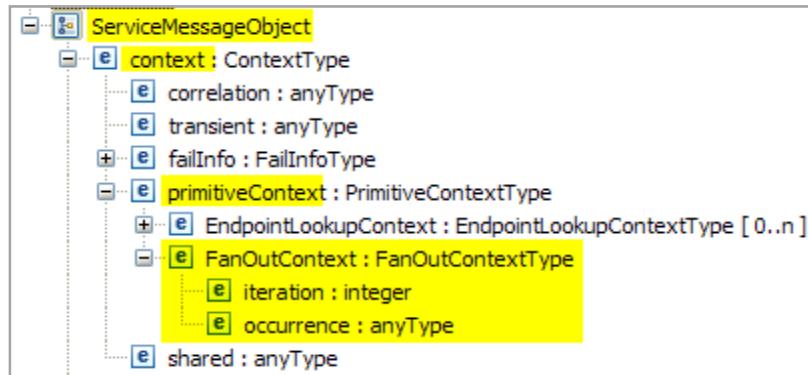
WB8601.2

Notes:

WebSphere Education

SMO structure: Context – Fan Out

- Fan Out context:
 - Used when processing a message in a Fan Out primitive using the “iterate” mode
 - Contained within the primitiveContext in the SMO



- FanOutContextType contains:
 - **Iteration:** Indicates the specific 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
 - anyType is downcast to the type of the element

© Copyright IBM Corporation 2016

Figure B-17. SMO structure: Context – Fan Out

WB8601.2

Notes:

Fan Out context: Downcasting the anyType

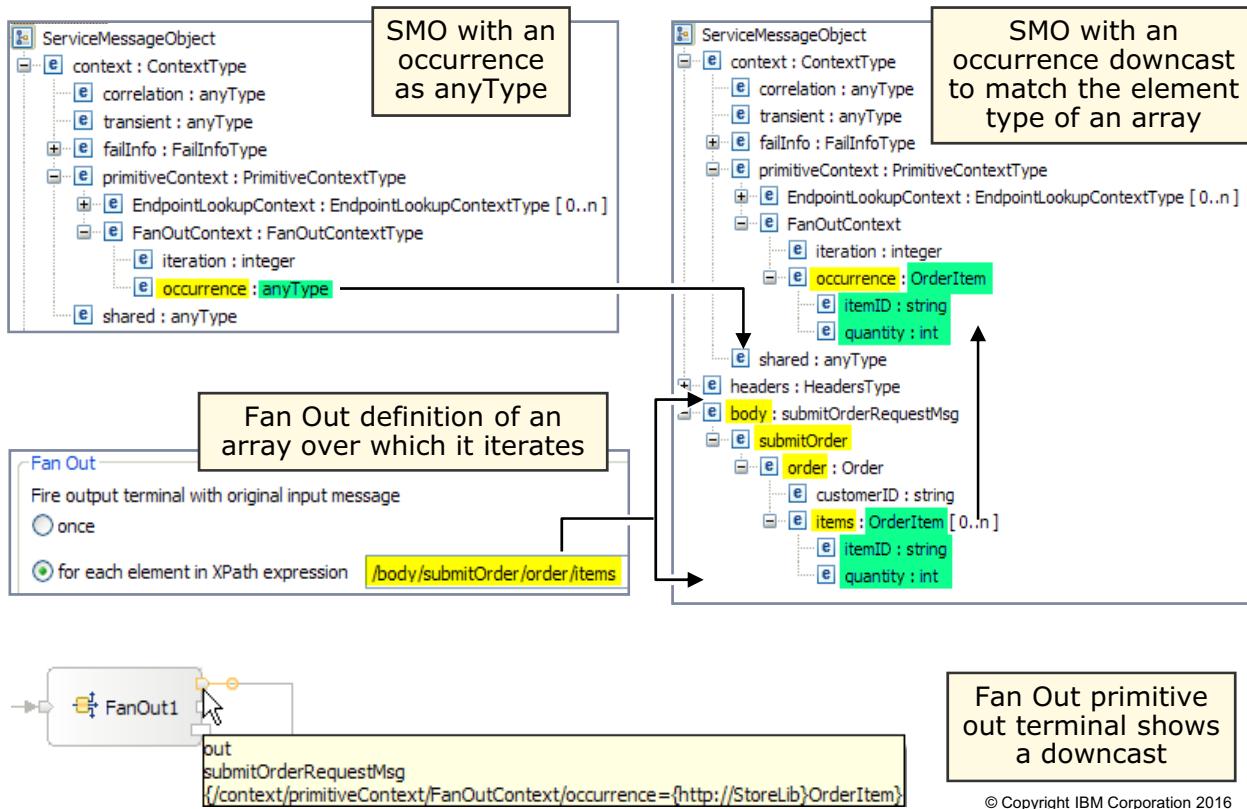
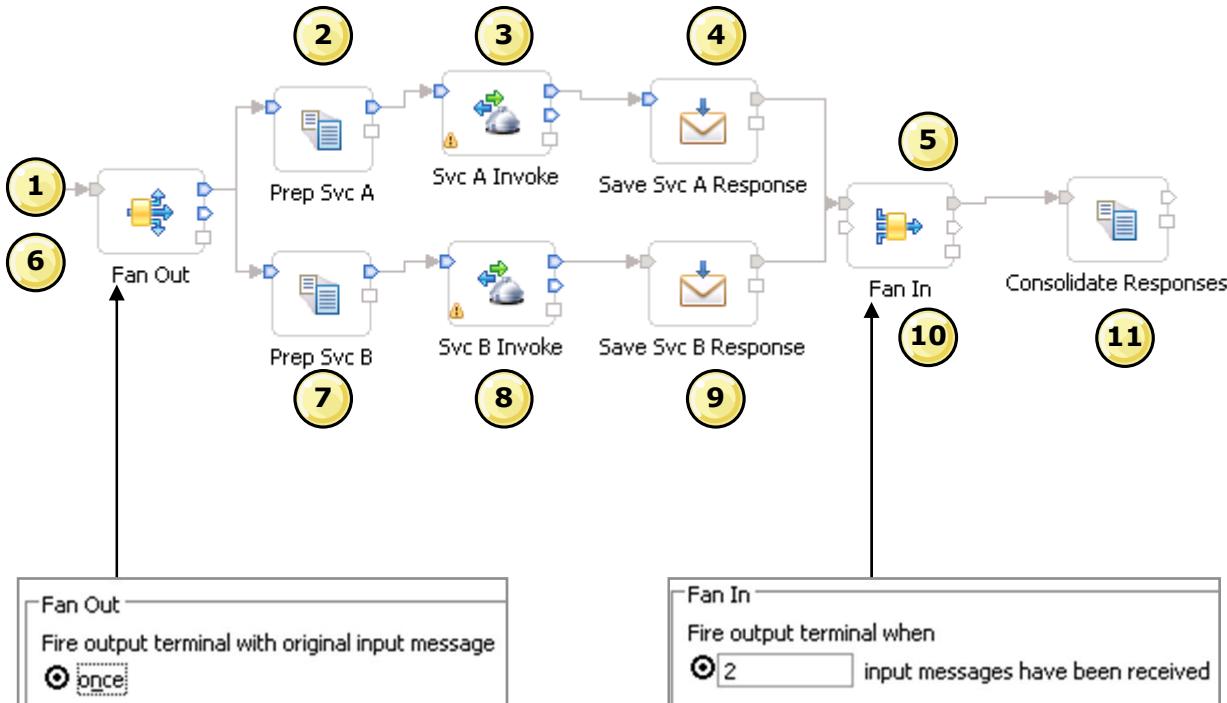


Figure B-18. Fan Out context: Downcasting the anyType

WB8601.2

Notes:

Fan Out and Fan In: Aggregation example



© Copyright IBM Corporation 2016

Figure B-19. Fan Out and Fan In: Aggregation example

WB8601.2

Notes:

Custom Mediation primitive

- Implement the custom logic: Java code, more terminals added to primitives, message types for all terminals, and any Java imports that the code requires

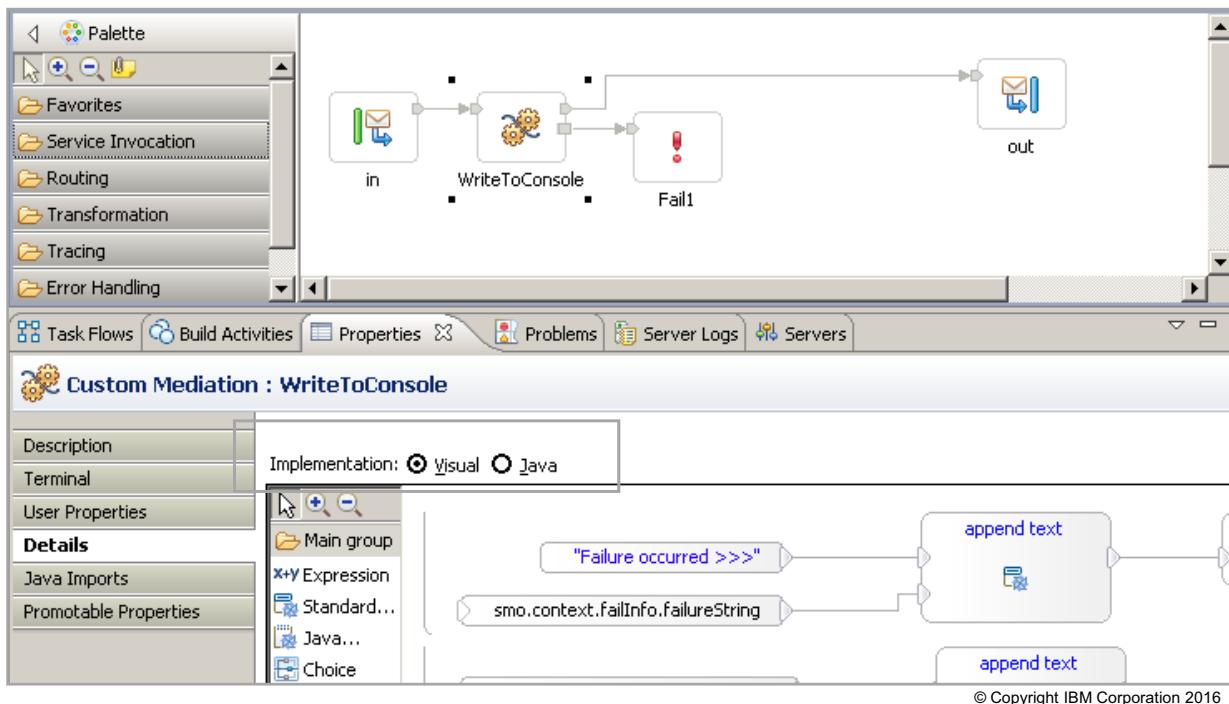


Figure B-20. Custom Mediation primitive

WB8601.2

Notes:



Database Lookup primitive (1 of 2)

- Alters the SMO with data that is retrieved from a relational database table
- You specify:
 - Data source name
 - Database table name
 - Name of the search column of the database table
 - An XPath expression that references the element of the incoming message to be used to do the lookup
 - The name of the database column from which to retrieve the data
 - The data type of the element to receive the value
 - An XPath expression that defines the message element to receive the retrieved value
 - *Validate input* property: If set to “true”, and the input message is invalid (does not match its schema), a runtime exception occurs



Database Lookup

Data source:*	<input type="text"/>
Table:*	<input type="text"/>
Search column:*	<input type="text"/>
Search location:*	<input type="text"/>

Column	Type	Target location

Choose the column to copy and the location to copy the value to.

Column:	<input type="text"/>
Type:	<input type="text"/> java.lang.String <input type="button" value="Browse..."/>
Target Location:	<input type="text"/> <input type="button" value="Browse..."/>

© Copyright IBM Corporation 2016

Figure B-21. Database Lookup primitive (1 of 2)

WB8601.2

Notes:

Database Lookup primitive (2 of 2)

- If no matching record is found in the database, the message is propagated through the `keyNotFound` terminal
- Only one database table can be read, and only one column value can be used in the retrieval
- The database, data source, and any server authentication settings must be set up for the Database Lookup mediation primitive to use
 - You can use the Integrated Solutions console (“administrative console”) to configure these settings
- You can route a message to the same location whether the key is found in the database or not
 - To configure the routing, you wire both the `out` terminal and the `keyNotFound` terminal to the same destination

© Copyright IBM Corporation 2016

Figure B-22. Database Lookup primitive (2 of 2)

WB8601.2

Notes:

Flow Order primitive

- Defines the order in which the branches of a mediation are executed
- At development time, you define a named output terminal for each branch 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 execute in the order named (out1, out2, out3, . . . outn)
 - The same input message is propagated on each iteration
- Each branch of the flow is completed before the next terminal is fired, except:
 - If an asynchronous Service Invoke is executed, the next branch from the flow order starts immediately
 - Exception that is thrown downstream and not handled locally causes the remaining flow order branches not to fire



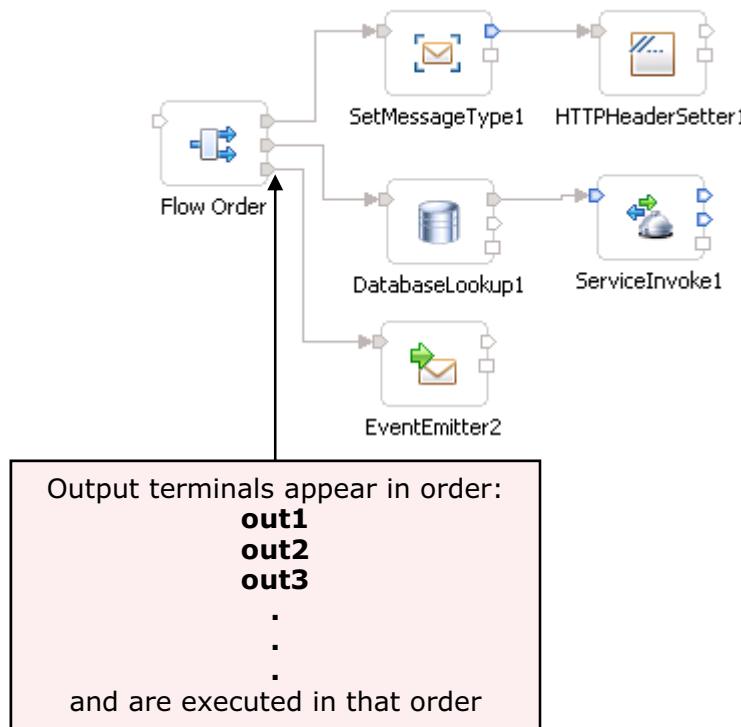
© Copyright IBM Corporation 2016

Figure B-23. Flow Order primitive

WB8601.2

Notes:

Flow Order primitive: Example



© Copyright IBM Corporation 2016

Figure B-24. Flow Order primitive: Example

WB8601.2

Notes:

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 to validate
 - Can be a specific element; for example, /root or /body
 - Note: The entire inbound message is not validated, unless that is what is specified in the **Root** property field



Message Validator

© Copyright IBM Corporation 2016

Figure B-25. Message Validator primitive

WB8601.2

Notes:



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

© Copyright IBM Corporation 2016

Figure B-26. Mediation subflow: Restrictions

WB8601.2

Notes:

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

© Copyright IBM Corporation 2016

Figure B-27. Mediation primitives: Dynamic routing

WB8601.2

Notes:

Introduction to dynamic message processing

- All of the service invocations that you saw so far were static; the connections to the external services were “fixed” or “hardwired”
- It can be useful to be able to invoke a service dynamically, by using some capability at run time to be able to determine which service to call, and how to call it
- You can also change the processing that occurs within a mediation flow and base it on policies that you retrieve from WebSphere Service Registry and Repository
- You typically discover service descriptions through a Universal Description, Discovery, and Integration (UDDI) registry or through WebSphere Service Registry and Repository
- Several mediation primitives are built to work with WebSphere Service Registry and Repository

© Copyright IBM Corporation 2016

Figure B-28. Introduction to dynamic message processing

WB8601.2

Notes:



WebSphere Service Registry and Repository

- WebSphere Service Registry and Repository is a separately purchased and installed product
- Facilitates storing, accessing, and managing service information (called metadata), so you can easily select, invoke, govern, and reuse services
- You can use it to store, access, and manage information about services
 - You can use this information to select, invoke, and reuse services
- You can use WebSphere Service Registry and Repository to store information about services in your systems or in other systems that you already use, 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 invokes 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

© Copyright IBM Corporation 2016

Figure B-29. WebSphere Service Registry and Repository

WB8601.2

Notes:

Endpoint Lookup primitive: Overview

- Uses a query that is constructed from the properties of the primitive to search WebSphere Service Registry and Repository
- 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 the one endpoint that matches the query
- Use to dynamically route messages, and base the routing upon 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



Endpoint Lookup

© Copyright IBM Corporation 2016

Figure B-30. Endpoint Lookup primitive: Overview

WB8601.2

Notes:



Endpoint Lookup primitive: Properties

Endpoint Lookup : Endpoint Lookup

Description	Name:	<input type="text"/>	<input type="button" value="Browse..."/>
Terminal	Namespace:	<input type="text"/>	
Details	Registry Name:	<Use default registry>	
Advanced	Match Policy:	Return first matching endpoint and set routing target	
Promotable Properties	Binding Type:	Web Services and SCA	
	Version:	<input type="text"/>	
	Module:	<input type="text"/>	
	Export:	<input type="text"/>	

- Properties specify how a query is presented to WebSphere Service Registry and Repository, and how the primitive handles data returned from the query
- Uses the endpoint reference structure that the WS-Addressing specification defines

© Copyright IBM Corporation 2016

Figure B-31. Endpoint Lookup primitive: Properties

WB8601.2

Notes:

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 invoke 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
- Note: The Endpoint Lookup primitive does not invoke a service; it merely sets the SMO headers for the next primitive that can use that routing information (for example, a Service Invoke)
- Primitive contains a *noMatch* terminal
 - If no policy match is found in WebSphere Service Registry and Repository, an inbound message is routed to that terminal

© Copyright IBM Corporation 2016

Figure B-32. Endpoint Lookup primitive: Actions

WB8601.2

Notes:

UDDI Endpoint Lookup primitive: Overview

- Retrieve service endpoint information from a Universal Description Discovery and Integration (UDDI) registry
 - The service endpoint information uses SOAP/HTTP or SOAP/JMS to relate directly to web services
- 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 the 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



UDDI Endpoint Lookup

© Copyright IBM Corporation 2016

Figure B-33. UDDI Endpoint Lookup primitive: Overview

WB8601.2

Notes:

UDDI Endpoint Lookup primitive: Properties

The screenshot shows a configuration dialog for the UDDI Endpoint Lookup primitive. On the left is a vertical toolbar with buttons for Description, Terminal, Details (which is selected), and Promotable Properties. The main area contains fields for Registry Name, Match Policy (set to 'Return first matching endpoint and set routing target'), Business Name, and Technical Model Names. A table lists technical model names, with buttons for Add... and Edit... at the bottom right.

- Properties specify how a query is presented to the UDDI registry, and how the primitive handles data returned from the query

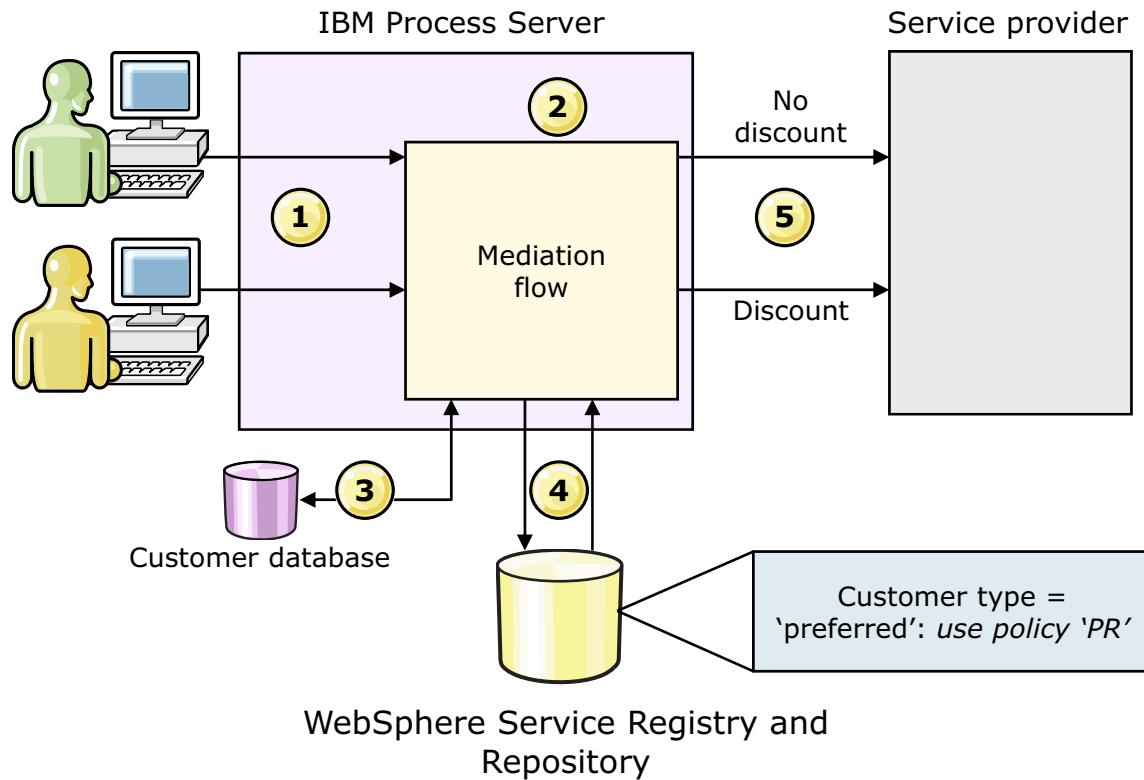
© Copyright IBM Corporation 2016

Figure B-34. UDDI Endpoint Lookup primitive: Properties

WB8601.2

Notes:

Mediation policies and dynamic service interactions



© Copyright IBM Corporation 2016

Figure B-35. Mediation policies and dynamic service interactions

WB8601.2

Notes:

Policy Resolution primitive: Overview

- With this primitive, you can use mediation policies dynamically to configure a mediation flow
 - Mediation policies are stored in, and retrieved from, WebSphere Service Registry and Repository
- You can use the Policy Resolution primitive to retrieve mediation policies that are associated with the current SCA module
- The properties of the primitive are used to create policy queries
- If valid mediation policies are found in the registry, their contents can be used 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



Policy Resolution

© Copyright IBM Corporation 2016

Figure B-36. Policy Resolution primitive: Overview

WB8601.2

Notes:



Policy Resolution primitive: Properties

Policy Resolution : Policy Resolution

Description	Registry Name: <Use default registry>								
Terminal	Policy Scope: Module								
Details	Conditions:								
Advanced	<table border="1"> <thead> <tr> <th>Policy condition name</th> <th>XPath</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Policy condition name	XPath						
Policy condition name	XPath								
Promotable Properties	<input type="checkbox"/> Add... <input type="checkbox"/> Edit... <input type="checkbox"/> Remove <input type="checkbox"/> Propagate mediation policy to response flow								

- *Policy scope* determines whether policy queries retrieve policies that are associated with the mediation module, the target service to invoke, or both
- *Policy conditions* are the names of a condition that a mediation policy must have
 - Conditions allow different mediation policies to apply in different contexts
 - At run time, the conditions must be satisfied before a conditional mediation policy can be used

© Copyright IBM Corporation 2016

Figure B-37. Policy Resolution primitive: Properties

WB8601.2

Notes:

SLA Check primitive: Overview

- Use the SLA Check mediation primitive to enforce service level agreements (SLAs)
- An IBM Process Server mediation 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
 - This primitive confirms whether 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



© Copyright IBM Corporation 2016

Figure B-38. SLA Check primitive: Overview

WB8601.2

Notes:



SLA Check primitive: Properties

SLA Check : SLA Check

Description	Registry Name:	<Use default registry>
Terminal	Endpoint:*	/headers/SMOHeader/Target/address
Details	Consumer ID:	<input type="text"/>
Promotable Properties	Context ID:	<input type="text"/>

© Copyright IBM Corporation 2016

Figure B-39. SLA Check primitive: Properties

WB8601.2

Notes:

Gateway Endpoint Lookup primitive: Overview

- Use to route service requests, which are 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
- 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 alternate 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`



Gateway Endpoint Lookup

© Copyright IBM Corporation 2016

Figure B-40. Gateway Endpoint Lookup primitive: Overview

WB8601.2

Notes:



Gateway Endpoint Lookup primitive: Properties

The screenshot shows a configuration dialog for the 'Gateway Endpoint Lookup' primitive. The title bar says 'Gateway Endpoint Lookup : Gateway Endpoint Lookup'. On the left is a vertical toolbar with icons for Description, Terminal, Details, Advanced, and Promotable Properties. The 'Details' tab is selected. The main area has a 'Lookup Method:' dropdown set to 'URL'. Below it is a 'Proxy Group Names:' section with a table containing one row and two columns. The first column is 'Proxy Group Name' and the second is an empty column. There are 'Add...' and 'Edit...' buttons to the right of the table.

- Lookup method:
 - **URL:** Use the input URL as the virtual service name to query the built-in configuration store
 - **XPath:** Use a virtual service name that is contained within the XPath expression to query the built-in configuration store
 - **Action:** Query WebSphere Service Registry and Repository for all endpoints

© Copyright IBM Corporation 2016

Figure B-41. Gateway Endpoint Lookup primitive: Properties

WB8601.2

Notes:



Event Emitter properties

Event Emitter : Event Emitter

Description	<input checked="" type="checkbox"/> Enabled
Terminal	Label: * <input type="text" value="Event emitter 11"/>
Details	Root: <input type="text" value="/root/body"/>
Promotable Properties	Transaction mode: <input type="text" value="Default"/>

- **Label:** Uniquely identifies the source of the event
- **Root:** Determines which part of the message is included with the emitted event
- **Transaction mode:** Determines whether the Event Emitter participates in a mediation flow transaction, or its own
- **Enabled:** You can suspend the mediate action by clearing the check box
 - This property is promotable, allowing the administrator to change it at run time, without having to remove the primitive (default value is “enabled”)

© Copyright IBM Corporation 2016

Figure B-42. Event Emitter properties

WB8601.2

Notes:

WebSphere Education

IBM

Message Logger properties

Message Logger : Message Logger

Description	<input checked="" type="checkbox"/> Enabled
Terminal	
Details	
Promotable Properties	
Root:	/body <input type="button" value="Edit..."/>
Transaction mode:	Same
Logging type:	Database
Data source name:	jdbc/mediation/messageLog
Handler:	com.ibm.ws.sibx.mediation.primitives.logger.WESBFileHandler
Formatter:	com.ibm.ws.sibx.mediation.primitives.logger.WESBFormatter
Filter:	com.ibm.ws.sibx.mediation.primitives.logger.WESBFilter
Literal:	{0},{1},{2},{3},{4},{5}
Level:	Info

© Copyright IBM Corporation 2016

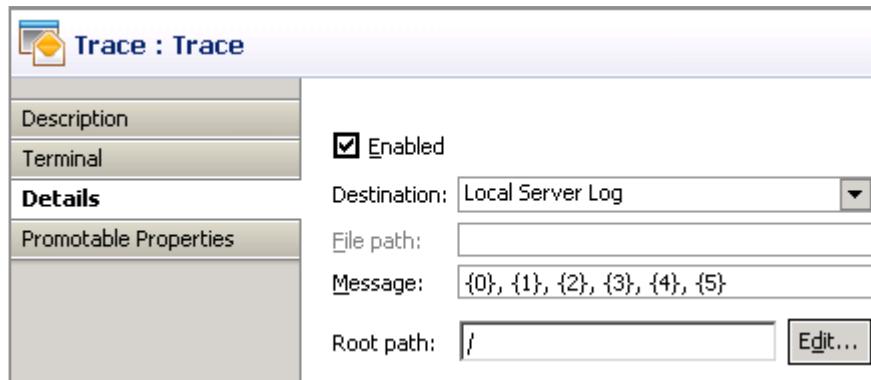
Figure B-43. Message Logger properties

WB8601.2

Notes:



Trace properties



- **Enabled:** Determines whether the Trace primitive writes trace data, and is promotable
- **Destination:** Determines where the trace data is written (a local server log, user trace file, or other file)
- **File path:** Specifies the file name to use, if tracing to a file
- **Message:** Descriptor fields to write with the message
- **Root:** XPath expression of the message to write

© Copyright IBM Corporation 2016

Figure B-44. Trace properties

WB8601.2

Notes:

Appendix C. Human tasks and human task clients

What this unit is about

In this unit, you learn about support for human tasks, and how human tasks are related to SCA components and activities in a business process.



Topics

- Introduction to human tasks
- Clients for human tasks

© Copyright IBM Corporation 2016

Figure C-1. Topics

WB8601.2

Notes:

C.1. Introduction to human tasks

Introduction to human tasks



© Copyright IBM Corporation 2016

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

10.1

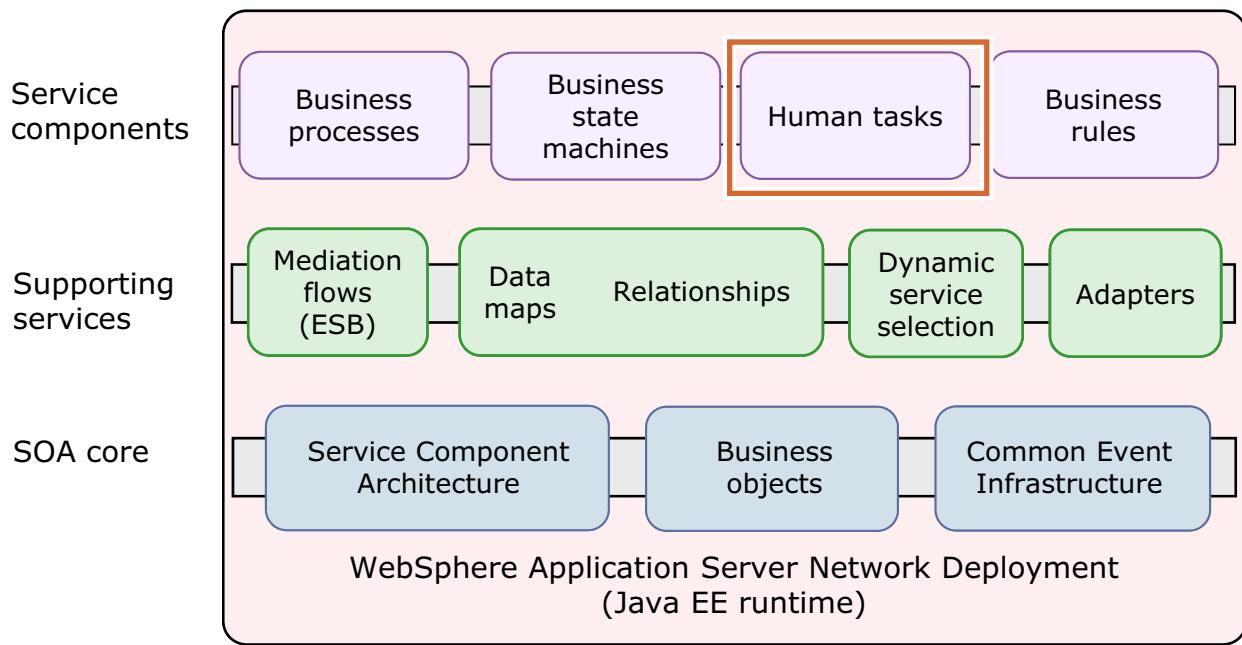
Figure C-2. Introduction to human tasks

WB8601.2

Notes:

Human tasks are service components

- The Human Task Manager runs in a separate container on IBM Process Server
 - Accessible through SCA or specific APIs



© Copyright IBM Corporation 2016

Figure C-3. Human tasks are service components

WB8601.2

Notes:



Overview of human task support

- Human interaction is key to many business integration applications
 - Human input or review is often required
 - Automation is not possible for some tasks
 - Error or exception situations might require handling by person
- Human tasks in IBM Integration Designer are robust and follow a service-oriented approach
 - Generate, assign, and store tasks for individuals and groups that are listed in organizational directories
 - Specify different levels of authority
 - Transfer and suspend tasks
 - Enable expiration, escalation, notification, subtasks, follow-on tasks, and participant substitution
 - Generate custom web clients for tasks

© Copyright IBM Corporation 2016

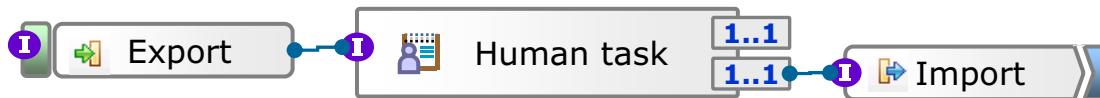
Figure C-4. Overview of human task support

WB8601.2

Notes:

Inline, stand-alone tasks and SCA

- Tasks that are implemented in a business process are called **inline** tasks
 - Inline tasks are activities in a business process, not SCA components
 - Inline tasks have access to process-related information
 - Inline tasks can be used to apply permissions to BPEL activities like invoke, receive, and receive choice, and to apply permissions to event handlers
 - Inline tasks can be used to create human tasks for process administration
- Stand-alone tasks are independent SCA components (not part of a business process)
 - Stand-alone tasks are invoked as services and can invoke other services
 - Stand-alone tasks can be invoked from other SCA components like business processes and other non-SCA components like JSP pages
 - Stand-alone tasks do not have direct access to process-related information
 - Business objects provide a standard data format for sending and receiving messages in human tasks



© Copyright IBM Corporation 2016

Figure C-5. Inline, stand-alone tasks and SCA

WB8601.2

Notes:

 WebSphere Education 

Inline tasks in business processes

- Human tasks can be used for process administration
- Grant humans administrative authorization to:
 - Suspend a process
 - Terminate a process
 - Restart processes
 - Force-retry or force-complete processes
 - Administer all process activities
 - And other tasks
- Inline tasks can also be used for authorization:
 - On incoming operations: Receive, receive choice, and event handlers
 - On other individual activities: Invoke, snippet, and other tasks

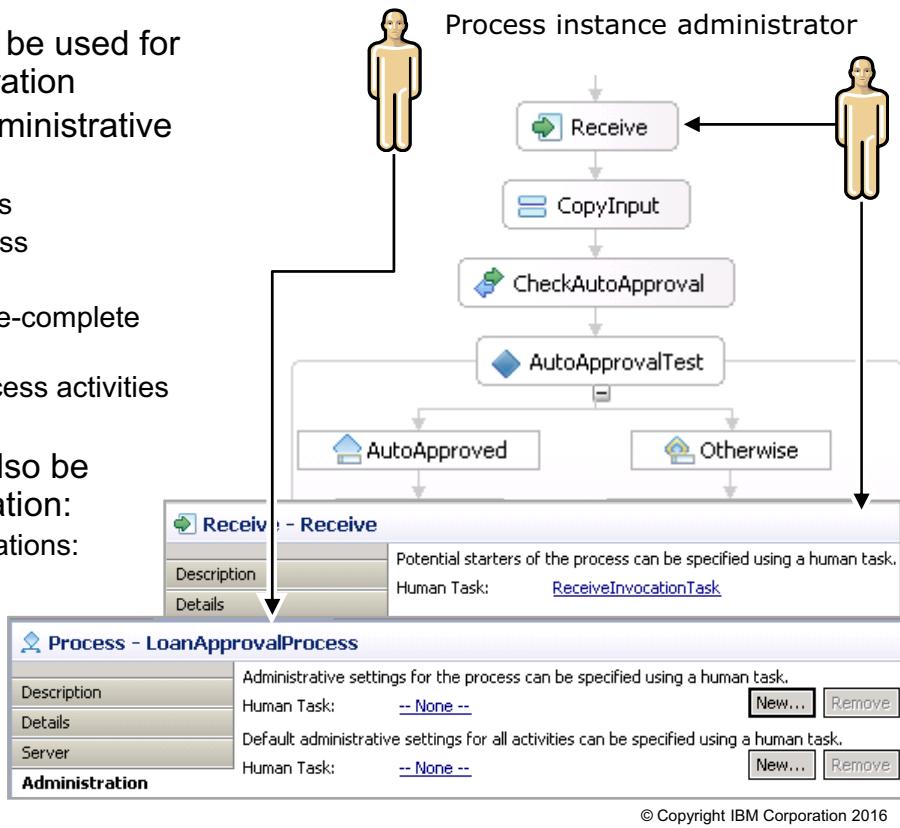


Figure C-6. Inline tasks in business processes

WB8601.2

Notes:

Types of human tasks

- Different types of human tasks are available for different integration situations

To-do task

- A service (WS-BPEL) creates work items for humans to do
- A to-do task can be either stand-alone or inline



Invocation task

- A human invokes a service
- An invocation task can be either stand-alone or inline



Collaboration task

- A human creates a task for another human
- Collaboration tasks are stand-alone
- No interaction occurs between collaboration and a business process



© Copyright IBM Corporation 2016

Figure C-7. Types of human tasks

WB8601.2

Notes:



Bind to-do task to invoking process

- Processes can manage the lifecycle of stand-alone, “child” to-do tasks
 - If the SCA by BPEL invoke activity calls the task, it behaves like a child task if bound to the process lifecycle; otherwise, it behaves like a stand-alone task
 - When a parent process is terminated, any bound stand-alone tasks are also terminated
 - “Bind the lifecycle to the invoking business process” is available only for to-do tasks

People directory:	Everyone
Task priority:	5
Work basket identifier:	<input type="text"/> <input type="button" value="Insert Vari..."/>
Business category:	<input type="text"/>
Default language: *	English - United States
Event handler name:	<input type="text"/>
Substitution policy:	No substitution
Date when task becomes valid:	<input type="button" value="None"/> <input type="button" value="Select Date"/> Time zone: America/New_York Time: <input type="text"/> : <input type="text"/> : <input type="text"/>
<input type="checkbox"/> Business-relevant <input checked="" type="checkbox"/> Enable subtask creation <input type="checkbox"/> Task can be claimed when it is suspended <input checked="" type="checkbox"/> Task can be delegated <input checked="" type="checkbox"/> Enable follow-on task creation <input type="checkbox"/> Bind the life cycle to the invoking component <input type="checkbox"/> Give owner read access to surrounding process context data	

© Copyright IBM Corporation 2016

Figure C-8. Bind to-do task to invoking process

WB8601.2

Notes:

Dynamic workflow: Subtasks

- Dynamic human task creation allows ad hoc human workflows
 - At run time, you can dynamically assign subtasks and follow-on tasks
- Subtasks allow ad hoc assignment of portions of tasks to others
 - A task can have an arbitrary number of subtasks; it remains active until all subtasks end
 - Partial delegation is assigned to one or many subtask owners
 - Subtask owners complete tasks that are assigned to them
 - Owner of the parent task remains responsible for completion
 - A subtask can be specified from inline stand-alone and to-do tasks or from collaboration tasks
- Business users can use the subtask feature by using Business Space and the human workflow widgets
- Subtasks can be created for tasks that are developed only with the “Enable subtask creation” option selected

© Copyright IBM Corporation 2016

Figure C-9. Dynamic workflow: Subtasks

WB8601.2

Notes:

Dynamic workflow: Follow-on tasks

- Follow-on tasks allow the assignment of the remaining work to others
 - A task can have at most one follow-on task
 - Follow-on tasks delegate the remaining work to other people
 - The owner of a follow-on task is responsible for completion
 - When a follow-on task starts, the original task ends
 - A follow-on task can be specified from inline stand-alone and to-do tasks or from collaboration tasks
 - Follow-on tasks also support subtasks and further follow-on tasks
- Follow-on tasks can be used to suspend a task without giving up ownership
 - Temporarily hide the task from a user's task list
 - Specify the duration for which the task is in hiding (suspend-until)
 - The task reappears on the task list after a duration
- Follow-on can be created for tasks that are developed with only the “Enable follow-on task creation” option selected

© Copyright IBM Corporation 2016

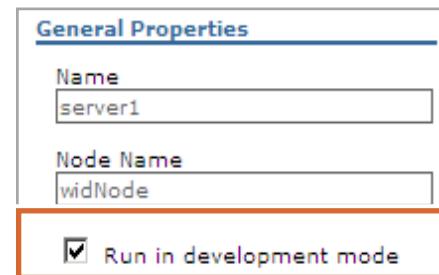
Figure C-10. Dynamic workflow: Follow-on tasks

WB8601.2

Notes:

Installing and uninstalling human task applications

- In the IBM Integration Designer test environment:
 - Server runs in “development mode” by default
 - Applications are installed and uninstalled easily by using add and remove projects
 - Removing applications does not require terminating and deleting instances or stopping task templates
- On a production server:
 - Not necessary to stop the process template or the task template before uninstalling an application; the template is stopped automatically
 - If uninstallation fails, the template is restarted
 - If instances are running, the uninstallation fails
 - A check is done after the template is stopped
 - Applications can be removed by using Jacl script even if instances exist in various states (cleans up templates and instances; should be used only during testing)



© Copyright IBM Corporation 2016

Figure C-11. Installing and uninstalling human task applications

WB8601.2

Notes:

C.2. Clients for human tasks

Clients for human tasks



© Copyright IBM Corporation 2016

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

10.1

Figure C-12. Clients for human tasks

WB8601.2

Notes:

Business calendar support (1 of 2)

- Business calendars compose meaningful time intervals into one business artifact in a module or library
 - Accessed by BPEL, human tasks, or Java API
- Business calendars can be used for any task duration:
 - Duration until due
 - Duration until expires
 - Duration until resumes
 - Duration until deletion
- Calendars can be used for any task escalation duration:
 - Duration until expires
 - Duration until repeats
- All durations of a human task use the same calendar
- Processes activities can use business calendars
 - Activities such as wait, invoke, and timeout

© Copyright IBM Corporation 2016

Figure C-13. Business calendar support (1 of 2)

WB8601.2

Notes:

The screenshot shows the WebSphere Education Business Calendar interface. At the top, there's a header with the IBM logo and navigation links. Below the header, the main title is "Business calendar support (2 of 2)".

Time Intervals: A section where you can add new intervals. It shows an interval from "9 to 5 Monday to Friday" and a button to "Add Interval". A callout box highlights "Working time intervals and exceptions are separate".

Exceptions: A section where you can add new exceptions. It shows an exception for "Lunch 12 to 1 Monday to Friday" and a button to "Add Exception".

Details for the Selected Exception: Shows the selected exception "Lunch 12 to 1 Monday to Friday" and its schedule settings.

First occurrence: Set to begin on Tuesday, January 1, 2008, at 12:00:00 PM, ending on Tuesday, January 1, 2008, at 1:00:00 PM, for one day.

Repeat weekly: Set to repeat every 1 week(s), on weekdays (Mon, Tue, Wed, Thu, Fri) but not on Saturday or Sunday.

Properties for business calendar BusinessCalendar: A summary view showing the weekly schedule for the week from 13-Oct-2008 to 20-Oct-2008. The days are color-coded: green for working time, red for off-time, and blue for net time.

Overview view: Monthly, weekly, daily, or hourly.

Intervals are color-coded:

- Green = On time
- Red = Off time
- Blue = Net time

© Copyright IBM Corporation 2016

Figure C-14. Business calendar support (2 of 2)

WB8601.2

Notes:



Clients for human tasks

- Several clients are available for working with human tasks that are specified in the task template user interface settings
- Most can be generated by using the graphical client wizard
 - Business Process Choreographer Explorer: JSF-based application for basic and administrative actions on tasks
 - IBM WebSphere Portal clients: Use an existing portlet or generate a portlet by using the portlet generator
 - IBM Forms clients: Generate a human task client, which is based on electronic forms
 - Business Space client: Web 2.0 client for creating human workflow workspaces
 - HTML-Dojo pages can be created and rendered in Business Space
 - JSF and JSP clients
 - Custom clients: Generate client by using calls to task container APIs
 - Coaches: Web-based interfaces where process participants do the work that is required to complete each task

© Copyright IBM Corporation 2016

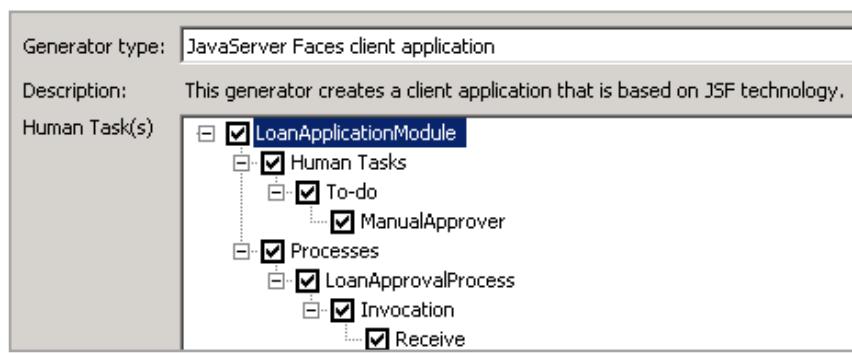
Figure C-15. Clients for human tasks

WB8601.2

Notes:

JSF and JSP clients

- Use a graphical wizard to generate JSP or JSF-based clients for human tasks
 - Stand-alone and inline human tasks for individual module or multiple modules
 - Specify which tasks to display in the generated client
 - Web or Java EE perspective is used to further customize JSF-based web clients
 - Provides an easy way to demonstrate human task capabilities that include ad hoc tasks (subtasks only)
- The wizard also supports JSF clients for tasks that are associated with business processes



© Copyright IBM Corporation 2016

Figure C-16. JSF and JSP clients

WB8601.2

Notes:

Business Process Choreographer Explorer client

Business Process Choreographer Explorer

Welcome admin | Logout | Manage Views | Customize | Help | About

Views

- Process Templates
 - Currently Valid
 - All Versions
- Process Instances
 - Started By Me
 - Administered By Me
 - Critical Processes
 - Terminated Processes
 - Failed Compensations
- Activity Instances
 - Stopped Activities
- Task Templates
 - My Task Templates
- Task Instances
 - My To-dos
 - All Tasks
 - Initiated By Me
 - Administered By Me

All Tasks

Use this page to work with task instances for which you have access rights. [?](#)

Work on | Release | Transfer | Start | Change Business Category | Refresh

Priority	Task Name	State	Kind	Owner
5	Request More Documentation	Ready	To-do Task	

Items found: 1 Items selected: 0 << Page 1 of 1 >>

- The prebuilt Business Explorer web client is included for basic actions and administrative tasks
 - Provides task administration: transfer, suspend, and resume
 - Provides basic task actions: view, claim, and complete
 - Supports dynamic tasks (called related tasks in the interface)

© Copyright IBM Corporation 2016

Figure C-17. Business Process Choreographer Explorer client

WB8601.2

Notes:

WebSphere Education

IBM WebSphere Portal clients

- Client generation wizard supports the portlet generator
 - Generates portlets for tasks
 - Used with the MyTasks portlet that is supplied with IBM WebSphere Portal
 - Supported only for to-do tasks
 - Portlet creation requires the Portal Toolkit

User Interface Wizard for Human Tasks

Client Generator Selection

Select a generator type and human tasks to generate a user interface.

Generator type: Portlet Generator

Welcome Work with Pages Workflow

Flight Book Portlet

Travel Request:

Employee	<input checked="" type="checkbox"/> Juergen Employee
Reason	Conference
Departure Date	5.10.2002, 9:00 am
Origin Airport	Frankfurt
Destination Airport	New York
Airline	Lufthansa
Class	Economy

Origin airport (IATA code):

Destination airport (IATA code):

Airport selection list:

Aalesund, Norway (AES)
Aberdeen, Scotland, United Kingdom (ABZ)
Aberdeen, SD, USA (ABR)
Abu Dhabi, United Arab Emirates (AUH)

Departure date (yyyy/mm/dd):

Return date (yyyy/mm/dd):

Departure time (hh:mm):

Return time (hh:mm):

Seat Class:

© Copyright IBM Corporation 2016

Figure C-18. IBM WebSphere Portal clients

WB8601.2

Notes:



Business Space for human workflow (1 of 2)

- Business Space is a Web 2.0 BPM client for business users
- Spaces: Pages that are built by using mashup technology that integrates widgets
 - Predefined space templates that contain task-related widgets support human-centric workspaces
 - Customized layouts can be defined by using available widgets
 - Includes graphical process widget, escalations, and task lists
- Widgets:
 - Use Representational State Transfer (REST) APIs to access runtimes
 - Dojo Toolkit and JavaScript used to create iWidget-compliant widgets
 - iWidget editor is provided as part of the web development tools
 - Widget content can be used in IBM WebSphere Portal
- Uses Business Space Manager
 - Common framework for user interfaces in IBM products

© Copyright IBM Corporation 2016

Figure C-19. Business Space for human workflow (1 of 2)

WB8601.2

Notes:



Business Space for human workflow (2 of 2)

- Business Space client for human workflow provides prebuilt templates that contain task-related widgets

The screenshot shows the Business Space for human workflow interface. It includes the following components:

- My Tasks**: A grid view for managing tasks. A callout box highlights it with the text: "Manage tasks and view available tasks with the My Tasks and Available Tasks widgets".
- Available Tasks**: A grid view for assessing available tasks. A callout box highlights it with the text: "Task escalations widget".
- Task Information**: A detailed view for a specific task named "Final Application Review". A callout box highlights it with the text: "Integrated forms". Below it, a section titled "Additional Information" displays details like owner, start date, status, and priority.
- Escalations**: A grid view for tracking task escalations. A callout box highlights it with the text: "History widget".
- History**: A log of task events. The history table shows two entries:

State	Name	Timestamp
Task created	admin	Feb 2, 2010 12:02:11 AM
Task started	admin	Feb 2, 2010 12:02:11 AM

© Copyright IBM Corporation 2016

Figure C-20. Business Space for human workflow (2 of 2)

WB8601.2

Notes:



HTML-Dojo client for Business Space

- HTML-Dojo forms are developed in IBM Integration Designer and rendered in the business user's workspace.

The screenshot shows the IBM Business Space interface. On the left, there are two lists of tasks: 'In Progress' and 'Overdue'. In the center, a 'Task Editor' window is open for a task titled 'Travel Approval'. This window contains a 'Task Form' section with fields for First Name (Fritz), Last Name (Mueller), Account (45TFD234), Customer (gold), Response (None Selected), Comments (empty), and Links (with two URLs: www.SampleURL.com and http://wap.quickr/Sampl). To the right of the Task Editor is a 'Page Explorer' window displaying the generated HTML-Dojo code. A callout box with the text 'The form that is developed in IBM Integration Designer is rendered in the Business Space UI' points from the task form in the Task Editor to the code in the Page Explorer.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict,
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>testProcessTask</title>
<script type="text/javascript" src="dojo1.0.2/dijit/form/parseOnLoad.js"></script>
<script type="text/javascript">
dojo.registerModulePath("bpc", "/bpc");

dojo.require("dojo.parser");
dojo.require("dijit.form.Button");
dojo.require("dijit.form.ComboBox");
dojo.require("dijit.form.DateTextBox");
dojo.require("dijit.form.NumberTextBox");
dojo.require("dijit.form.Textarea");
dojo.require("dijit.form.TextBox");
dojo.require("dijit.form.CheckBox");
dojo.require("dijit.form.DateTextBox");
dojo.require("dijit.form.TimeTextBox");
dojo.require("dijit.form.Form");
dojo.require("dijit.form.FilteringSelect");
dojo.require("bpc.forms.DateTimeBox");
dojo.require("bpc.forms.FormArrayContainer")
</script>

```

© Copyright IBM Corporation 2016

Figure C-21. HTML-Dojo client for Business Space

WB8601.2

Notes:

WebSphere Education

Coach

The screenshot shows the 'Submit Requisition NG' Coach interface. The top navigation bar includes 'Overview', 'Diagram', 'Variables', and 'Coaches'. The 'Coaches' tab is selected. On the left, three steps are listed: 'Create job requisition NG', 'Confirm position details NG', and 'Provide details for existing position NG'. The main area is titled 'Submit job requisition' and contains a 'Job requisition data' section. This section is divided into three columns: 'Requester' (containing requisition number 5067125469 and requestor Jane McCormick), 'Requested job position' (containing employment status Contract, department Product Development, and required employees 1), and 'Requested job : date and location' (containing planned start date 7/27/2012 and location Minneapolis). Below this is a 'Make your decision' section with a 'Next' button.

- Create rich user interfaces easily and quickly
- Create and distribute their custom coach views

© Copyright IBM Corporation 2016

Figure C-22. Coach

WB8601.2

Notes:



Coach view

- Reusable sets of user interfaces
- Can be created in a toolkit or a process application
 - A library asset under User Interface
- Create a custom coach view

© Copyright IBM Corporation 2016

Figure C-23. Coach view

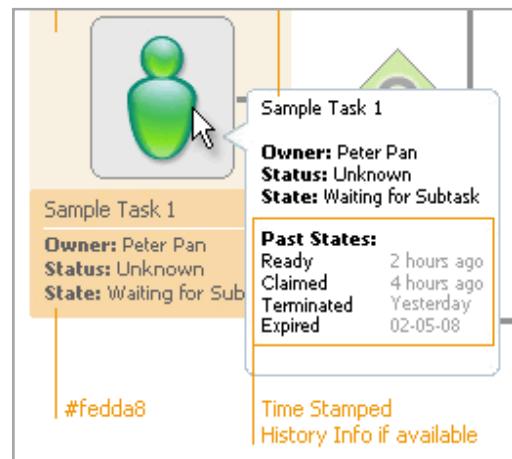
WB8601.2

Notes:

 WebSphere Education 

History log for human tasks

- History log allows business users to view the task history
- Exposed by graphical process diagram in Business Space and BPEL Explorer
- Logs any changes to a human task
- Tightly coupled with the human task
 - Events are stored until the human task is deleted
 - When a human task is deleted, its task history events are deleted
 - Audit log and CEI events are stored until manual housekeeping is done



Subtasks		
History		
Principal	Event	Timestamp
wswf	Task claimed	6/26/08 10:17 AM
bman1	Subtask created	6/26/08 11:17 AM
wswf	Subtask completed	6/26/08 12:17 PM
buser1	Task transferred	6/26/08 1:17 PM
wswf	Task suspended	6/26/08 2:12 PM
wswf	Task resumed	6/26/08 3:12 PM

© Copyright IBM Corporation 2016

Figure C-24. History log for human tasks

WB8601.2

Notes:



Checkpoint questions

1. True or False: A service creates to-do tasks.
2. True or False: A coach component can be created in the IBM Integration Designer.
3. True or False: Stand-alone tasks are activities in a business process, not SCA components.
4. True or False: Subtasks are dynamic tasks that can be created at run time.

© Copyright IBM Corporation 2016

Figure C-25. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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



Checkpoint answers

1. True. A service creates a to-do task, an invocation task invokes a service, and collaboration tasks are tasks that humans create for humans.
2. False. It is created in IBM Process Designer.
3. False. Inline tasks are activities in a business process, not SCA components. Stand-alone tasks are independent SCA components (not part of a business process).
4. True. Subtasks and follow-on tasks are both dynamic tasks that can be created at run time.

© Copyright IBM Corporation 2016

Figure C-26. Checkpoint answers

WB8601.2

Notes:



References

- developerWorks: Authorization and staff resolution in Business Process Choreographer, Part 1
 - http://www.ibm.com/developerworks/websphere/techjournal/0710_lind/0710_lind.html
- developerWorks: Implementing a human-centric business process application using IBM WebSphere Portlet Factory, Part 1
 - http://www.ibm.com/developerworks/websphere/library/techarticles/0804_ng/0804_ng.html
- WebSphere Application Server Enterprise Process Choreographer: Staff Resolution Parameter Reference
 - http://www.ibm.com/developerworks/websphere/library/techarticles/wasid/WPC_StaffReference.html

© Copyright IBM Corporation 2016

Figure C-27. References

WB8601.2

Notes:

Appendix D. Human task roles, people queries, and escalations

What this unit is about

This unit covers the various human task roles, escalations, and how to write queries against a human task repository.



Topics

- Human task assignment
- Escalation and notification

© Copyright IBM Corporation 2016

Figure D-1. Topics

WB8601.2

Notes:

D.1. Human task assignment

Human task assignment



© Copyright IBM Corporation 2016

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

10.1

Figure D-2. Human task assignment

WB8601.2

Notes:

Roles for human tasks

- Human tasks have assigned roles that determine access
 - You can assign a user or group of users to each role
 - Roles can be Java EE roles or instance-based roles
- Instance-based roles include:
 - **Potential instance creator:** This role can create an instance of the task
 - **Originator:** This role has administrative rights until the task starts
 - **Potential owner:** This role can claim, work on, and complete tasks
 - **Owner:** This role claims and completes tasks
 - **Potential starter:** This role can start an instance of the task
 - **Starter:** This role has reader authority and can do some administrative functions like transferring work items
 - **Administrator:** This role can administer tasks, task templates, and escalations
 - **Editor:** This role can work with task content, but cannot claim or complete it
 - **Reader:** This role can view tasks but cannot work on them
 - **Escalation receiver:** Members of this role have the authority of a reader for the escalation and the escalated task
- Roles that are not assigned in the task can be assigned default values

© Copyright IBM Corporation 2016

Figure D-3. Roles for human tasks

WB8601.2

Notes:



People assignment

- Assignment of users and groups to an instance-based role is done by using **assignment criteria**, which are resolved to a specific **people query** for a specific directory
- The assignment criteria define the members of the role and are narrowed according to a set of parameters, creating an authorization rule
- People queries can be tested in the human task editor

Staff role: Potential Creators

People Assignment (Receiver)	
Potential Creators	Group Members
	GroupName * IncludeSubgroups *
Potential Owners	Approvers Everybody

People assignment criteria: Group Members

Assigns members of groups. Supported by default configurations for: Virtual Member Manager, LDAP, User Registry. Use this to create individual assignments for every group member. Define a group name as a: uniqueName (VMM), DN (LDAP), security realm specific name (User Registry: Local OS, LDAP, custom).

Name	Value
GroupName *	Approvers
IncludeSubgroups *	false
Domain	

© Copyright IBM Corporation 2016

Figure D-4. People assignment

WB8601.2

Notes:

People assignment criteria

Assignment criteria	Sample parameter values
Users	Name
Users by user ID	UserID
Group members	GroupName
Group members without named users	GroupName, and Named Users
Group members without filtered users	GroupName, FilterAttribute, FilterValue, and IncludeSubgroups
Department members	DepartmentName
Role members	RoleName, and IncludeNestedRoles
Manager of employee	EmployeeName
Person search	UserID, LastName, FirstName
Group search	GroupID
Everybody	All authenticated users (no parameters)
Group	GroupID
User records by user ID	UserID
Users by user ID without named users	UserID, and NamedUsers

- Several criteria and associated parameter mappings are included in IBM Integration Designer
- Some criteria are available only with certain user registries

© Copyright IBM Corporation 2016

Figure D-5. People assignment criteria

WB8601.2

Notes:

Bindings for assignment criteria parameters

- Parameters for assignment criteria can be set at development or dynamically at run time
- Early binding (static binding):
 - Early binding uses specific hardcoded values such as John Doe, jsmith, Administrators, or WestCoastManagers
 - Values are set in the deployed query:

```
select all where Group = "Approvers"
```
- Late binding (dynamic binding):
 - Late binding uses values from the process or task instance context such as process or task variables or custom attributes or properties
 - The deployed query is parameterized to accept values at execution:

```
select all where Group = %myValue%
```
 - Late binding is flexible, but requires more resources

© Copyright IBM Corporation 2016

Figure D-6. Bindings for assignment criteria parameters

WB8601.2

Notes:



People query results: Work items

- The result of a people query is the creation of one or more work items in the Business Process Choreographer database
 - People queries are also used to send email notifications
- Work items provide an association between users and a particular business object
 - For inline tasks, the user or group is associated with a process instance
 - For stand-alone tasks, the user or group is associated with a human task instance
 - Each role grants authorization for a fixed set of actions that the user can do on the task or process instance
- Three types of work items are used to store people queries
 - Everybody work items grant authorization to every authenticated user
 - User work items grant authorization to each user whose ID is stored with the work item
 - Group work items grant authorization to a group of people
 - Authorization is done by verifying the list of group memberships that are associated with the user's login context information

© Copyright IBM Corporation 2016

Figure D-7. People query results: Work items

WB8601.2

Notes:



Assignment criteria and work items

- The chosen assignment criteria affect how many work items are created
- Work item creation is done in one of four ways:
 - Assigning an individual to a role
 - Assigning multiple individuals to a role
 - Assigning members of a group to a role
 - Assigning a group to a role (requires the “Enable group work items” option in the administrative console)
- When you assign multiple individuals to a role, or when you assign all group members to a role:
 - Many work items can be created
 - Requires one table row per role member in the work item view of the Business Process Choreographer database
 - More rows are required for other roles such as an administrator

© Copyright IBM Corporation 2016

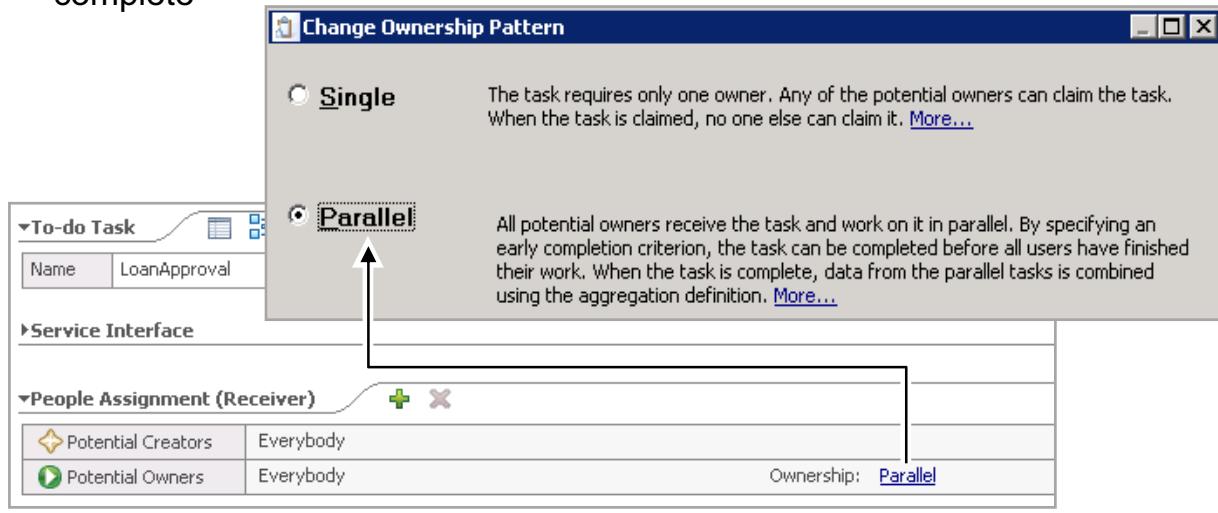
Figure D-8. Assignment criteria and work items

WB8601.2

Notes:

Task ownership patterns (1 of 2)

- When work items are assigned, they are owned by a single user or by users in parallel
- In parallel ownership, a subtask is created for every potential owner
- Owners work simultaneously, and when they are finished, criteria that you specify are used to aggregate the results and determine when the task is complete



© Copyright IBM Corporation 2016

Figure D-9. Task ownership patterns (1 of 2)

WB8601.2

Notes:



Task ownership patterns (2 of 2)

- When ownership is parallel, completion conditions specify when the parallel subtasks should be terminated
 - Examples: When most of the subtasks are completed or after time elapses
 - Incomplete subtasks are terminated automatically
- If a completion condition is met, the result aggregation method is run
 - Majority rule, unanimity required

Staff role - Potential Owners (Parallel ownership)

Specify the conditions under which the task ends before all owners have completed their tasks.

End the task based on the time that elapsed since it was created
 Calendar type: WorkWeek {<http://FoundationLibrary>}
 Days Hours Minutes Seconds

End the task when the following condition is met
 Expression language: Simple condition
 Condition: Select a template
 Percentage of workers
 One disapproval
 One approval
 Percentage of approvals
 Majority reached

© Copyright IBM Corporation 2016

Figure D-10. Task ownership patterns (2 of 2)

WB8601.2

Notes:

Query resolution at run time

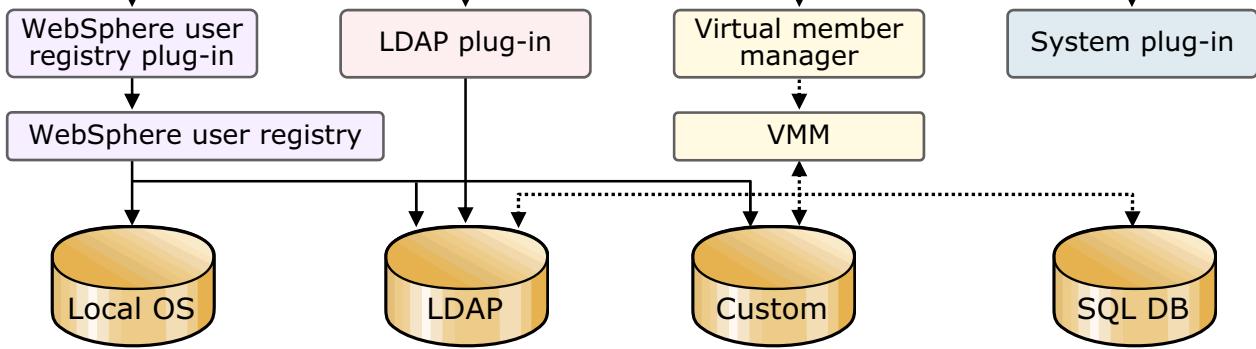
At run time, the people assignment service provides access to one of the people directory plug-ins to query a particular user registry

- **LDAP:** Uses an LDAP registry
- **User registry:** Supports the local operating system directory, LDAP, or custom registry by using the user registry API

- **System:** Uses static group values, user values, or context variables (for testing)
- **“Everybody”:** Based on the system plug-in, but always returns “everybody” as the query result (testing only)

Human Task Manager

People assignment service



© Copyright IBM Corporation 2016

Figure D-11. Query resolution at run time

WB8601.2

Notes:

Support for virtual member manager (VMM)

- VMM is supported to integrate organizational directories
 - Federates several supported user repositories into a single IBM Process Server-facing user directory

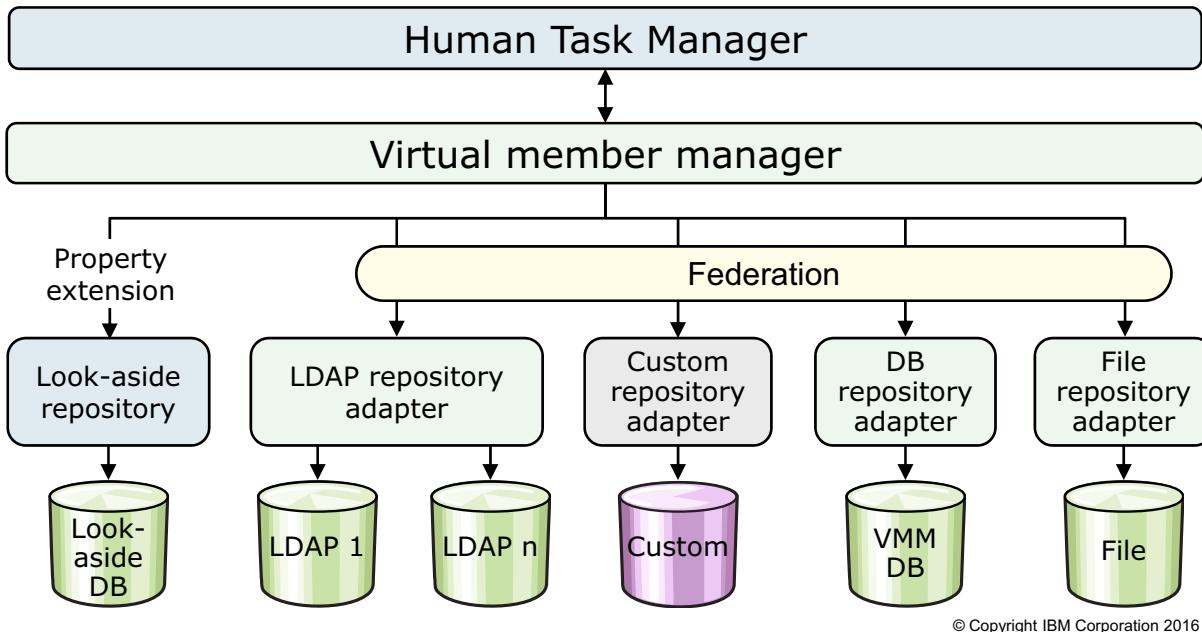


Figure D-12. Support for virtual member manager (VMM)

WB8601.2

Notes:

Participant substitution

- VMM support allows (and is required for) participant substitution
 - How tasks and escalations are reassigned for absent users
 - Same policy is applied for all of task roles that are associated with a template
- At run time, task ownership is reassigned
 - If people notify the system of absence, work is assigned to substitutes
 - Every user manages a list of substitutes and personal absences
 - Tasks that are assigned to absent users get automatically reassigned to specified substitutes

Define Substitutes

Use this page to manage absence settings for other users. Enter the user's name, select the absence checkbox, and specify the user's substitutes. The substitutes receive the work assignments for the absent user. [\[?\]](#)

Save	Cancel	
User name	<input type="text" value="admin"/> Lookup	Fill in a user name and click Lookup to load and display the settings for this user.
User is absent	<input type="checkbox"/>	To add a substitute fill in a user name and click Add .
User's substitutes	<input type="text"/> Add	

© Copyright IBM Corporation 2016

Figure D-13. Participant substitution

WB8601.2

Notes:



People directory provider selection

- Select the appropriate people directory provider in IBM Integration Designer
 - Provider configuration is done in the IBM Process Server administrative console
- The people directory provider registers as a server resource
 - The resource is accessible through a JNDI name
- When the task is deployed, the people assignment service establishes which people directory provider to use
 - It transforms the people assignment criteria into a query for the people directory provider, which is stored internally

To-do Task - RequestMoreDocumentation		
Description	People directory:	<input type="text" value="Everyone"/>
Details	Task priority:	<input type="text" value="Virtual Member Manager"/>
Propagation	Work basket identifier:	<input type="text" value="Lightweight Directory Access Protocol"/>
Interface	Business category:	<input type="text" value="User registry"/>
Duration	Default language: *	<input type="text" value="System"/>
Event Monitor	Event handler name:	<input type="text" value="Everyone"/>
Global Event Settings	Substitution policy:	<input type="text" value="English - United States"/>
Environment		<input type="text" value="No substitution"/>

© Copyright IBM Corporation 2016

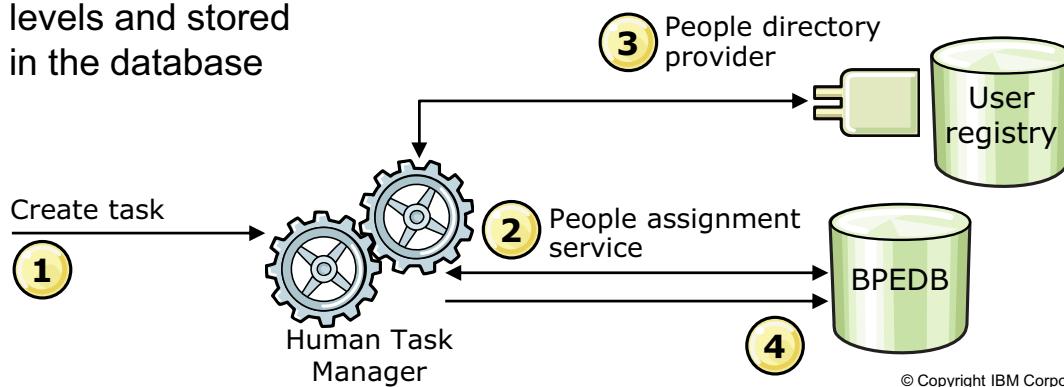
Figure D-14. People directory provider selection

WB8601.2

Notes:

Human task and query creation

1. The Human Task Manager (HTM) receives a task request
2. The HTM retrieves the people query from the BPEDB database
3. The people assignment service passes the people query to the people directory provider where the query is run against the user registry to retrieve a user, a group of users, or a group
4. The query results are returned to the Human Task Manager, and tasks are created for selected users with different permission levels and stored in the database



© Copyright IBM Corporation 2016

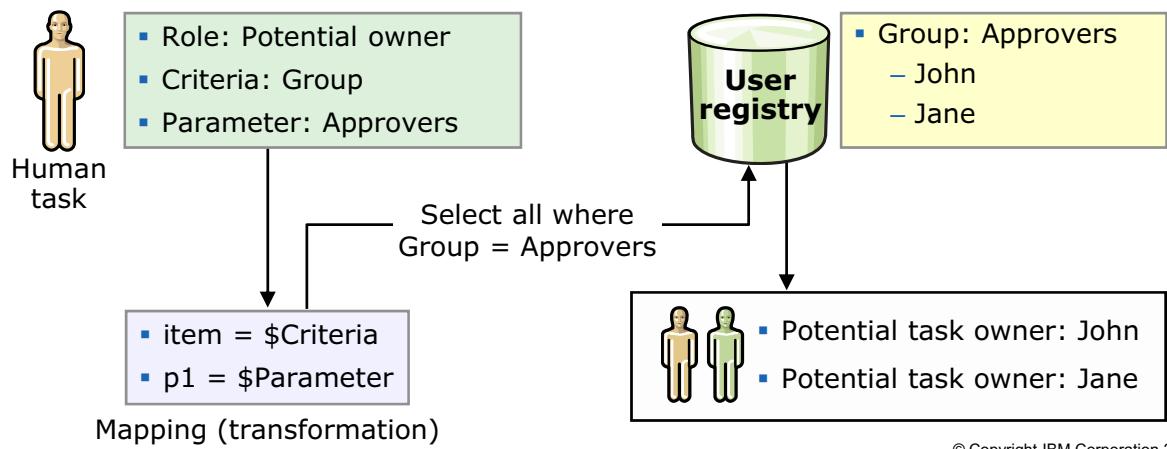
Figure D-15. Human task and query creation

WB8601.2

Notes:

People query example

- In this example, the “Potential owner” role is assigned to the “Approvers” group
- Both John and Jane, as members of the approvers group, can claim the task
- A people assignment post-processor plug-in is available to further modify the query result after it is returned to the Human Task Manager



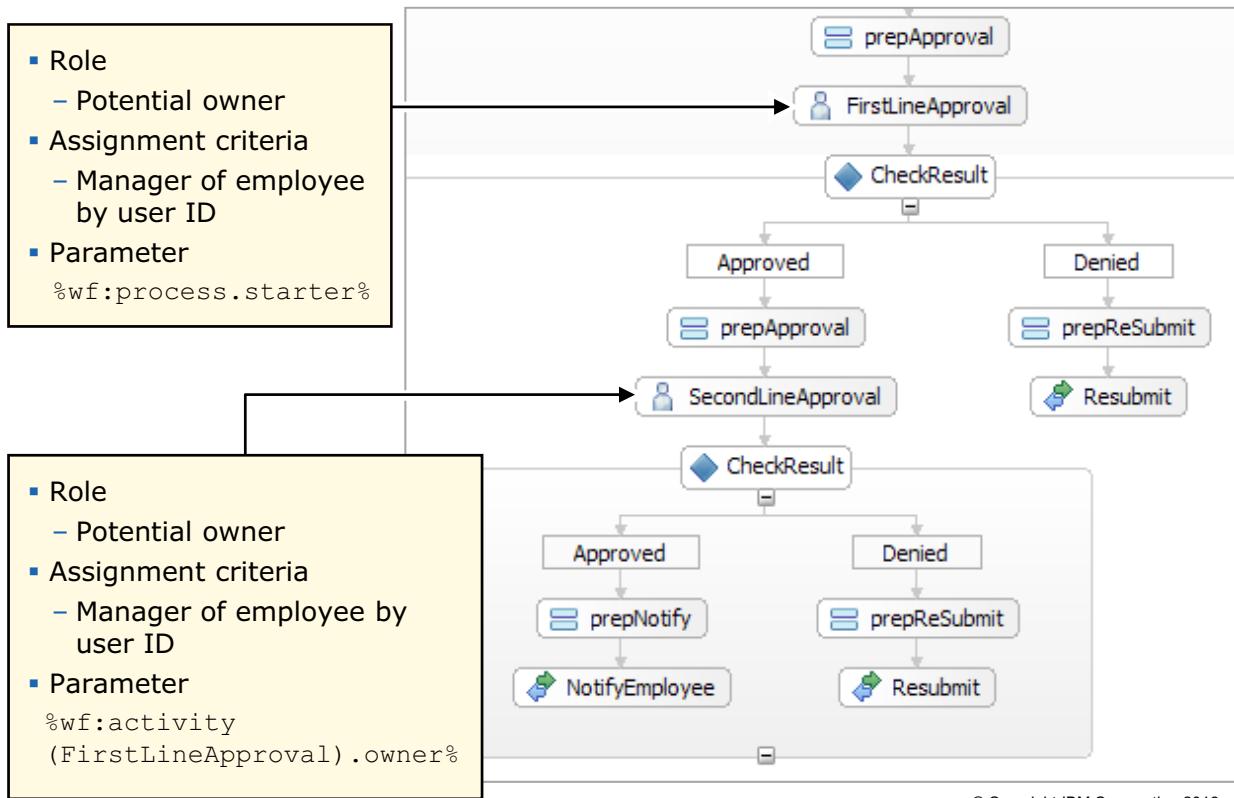
© Copyright IBM Corporation 2016

Figure D-16. People query example

WB8601.2

Notes:

Example: People query in BPEL for multiple reviews



© Copyright IBM Corporation 2016

Figure D-17. Example: People query in BPEL for multiple reviews

WB8601.2

Notes:

D.2. Escalation and notification

Escalation and notification



© Copyright IBM Corporation 2016

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

10.1

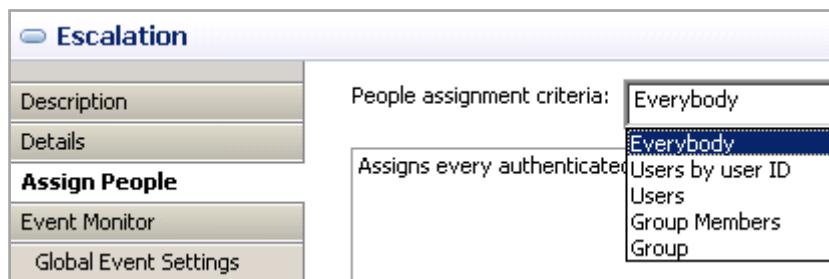
Figure D-18. Escalation and notification

WB8601.2

Notes:

Escalation support (1 of 2)

- Escalations are triggered when action is not taken on a task within a certain time period
 - Typically used to create notifications for overdue tasks
 - Can be used to increase the task priority
 - Must be defined; escalations are not automatic
- Escalations can be defined in escalation templates in IBM Integration Designer, or they can be defined in ad hoc tasks at run time
 - Escalations are assigned by using people assignment criteria



© Copyright IBM Corporation 2016

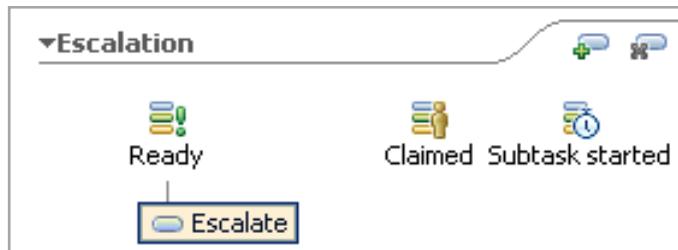
Figure D-19. Escalation support (1 of 2)

WB8601.2

Notes:

Escalation support (2 of 2)

- Escalations are activated based on the state of the task:
 - **Ready:** Task is waiting to be claimed, but is not claimed
 - **Claimed:** Task is claimed but not complete
 - **Subtask started:** A relevant subtask is started but not complete (the parent task is escalated, not the subtask)
- Duration can be a fixed value or dynamic (based on variable)
 - Fixed values can be days, hours, minutes, and seconds
 - Fixed value can also be calendar type: Simple, WebSphere CRON, and user-defined



© Copyright IBM Corporation 2016

Figure D-20. Escalation support (2 of 2)

WB8601.2

Notes:

Setting notifications

- When an escalation is activated, the notification type can be:
 - A work item
 - An email notification
 - An event
- People assignment criteria are also used to send email notifications
 - Person records must have an email address that is defined in the user registry
 - Use a default email message or customize it to include task details in subject or body
 - Default mail server preconfigured (WebSphere Application Server JavaMail)
- Notifications can repeat, based on a fixed duration or calendar type

Notification type: Work item

E-mail message:

Repeat settings → Repeat notification every: 0 Days 0 Hours 0 Minutes 0 Seconds

Increase task priority: No

© Copyright IBM Corporation 2016

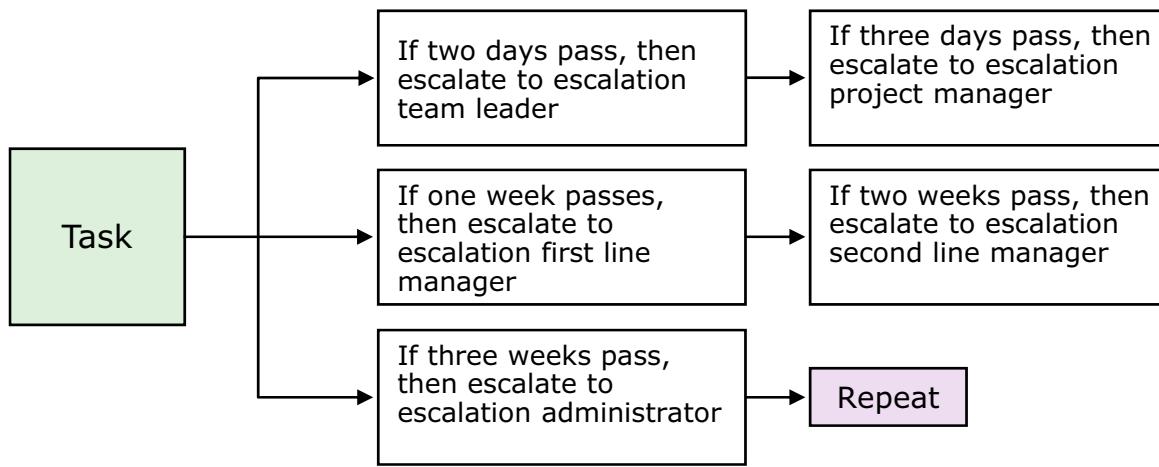
Figure D-21. Setting notifications

WB8601.2

Notes:

Escalation paths

- Escalations can be chained or run in parallel
- In a chain, the escalations are run sequentially
 - Single or multiple chains of escalations can be defined
 - Time and task state are checked before escalation
- Parallel escalations are in separate paths but are run at the same time



© Copyright IBM Corporation 2016

Figure D-22. Escalation paths

WB8601.2

Notes:



Checkpoint questions

1. True or False: Early binding of staff query parameters uses values from the BPEL process or task instance context.
2. True or False: The two task ownership patterns are called single and parallel patterns.
3. True or False: VMM is a people directory provider that is required for participant substitution.
4. True or False: A notification can repeat based on a fixed duration or calendar type.

© Copyright IBM Corporation 2016

Figure D-23. Checkpoint questions

WB8601.2

Notes:

Write your answers here:

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



Checkpoint answers

1. False. Early binding hardcodes the parameters for staff queries in the task definition and cannot be changed unless you update them in the development environment. Late binding of staff query parameters uses values from the business process or task instance context (the deployed query is parameterized to accept values at execution).
2. True.
3. True.
4. True.

© Copyright IBM Corporation 2016

Figure D-24. Checkpoint answers

WB8601.2

Notes:



References

- developerWorks: Authorization and staff resolution in Business Process Choreographer, Part 1
 - http://www.ibm.com/developerworks/websphere/techjournal/0710_lind/0710_lind.html
- WebSphere Application Server Enterprise Process Choreographer: Staff Resolution Parameter Reference
 - http://www.ibm.com/developerworks/websphere/library/techarticles/wasid/WPC_StaffReference.html

© Copyright IBM Corporation 2016

Figure D-25. References

WB8601.2

Notes:

Appendix E. List of abbreviations

A

ACL	access control list
ACID	atomicity, consistency, isolation, and durability
ACORD	Association for Cooperative Operations Research and Development
AFC	Adapter Foundation Classes
AIS	Advanced Integration service
AIX	Advanced IBM UNIX
Ajax	Asynchronous JavaScript and XML
Ant	Another Neat Tool
API	application programming interface
AQ	Advanced Queuing
ASBO	application-specific business object
AVS	Account Verification Skeleton

B

B2B	business-to-business
BAL	Business Action Language
BAM	business activity monitoring
BFM	Business Flow Management
BFM	Business Flow Manager
BG	business graph
BI	business integration
BO	business object
BPD	business process definition
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPELJ	Business Process Execution Language for Java
BPM	business process management
BPM	Business Process Manager
BPMN	Business Process Modeling Notation
BRM	business rules manager
BSS	business support systems

C

C++	C object-oriented programming language
CCI	common client interface
CD	compact disc
CEI	Common Event Infrastructure
CICS	Customer Information Control System
CLI	command-line interface
COBOL	Common Business Oriented Language
CRM	customer relationship management
CRON	Chronograph
CSV	comma-separated values
CVS	Concurrent Versions System

D

DAAPI	data access API
DAS	data access service
DB	database
DB2	Database 2
DESPI	Data Exchange Service Provider Interface
DoS	denial-of-service
DOS	Disk Operating System

E

EAI	Enterprise Application Integration
EAR	enterprise archive
EBS	E-Business Suite
ECI	External Call Interface
ECS	event correlation sphere
EE	Enterprise Edition
EIS	enterprise information system
EJB	Enterprise JavaBeans
EMD	Enterprise Metadata Discovery
EMF	Eclipse Modeling Framework
ERC	edition revision code

ERP enterprise resource planning

ESB enterprise service bus

F

FAQ frequently asked questions

FFDC first-failure data capture

FIFO first-in first-out

FTP File Transfer Protocol

G

GB gigabyte

GBO generic business object

GM general manager

GMT Greenwich mean time

GUI graphical user interface

H

HATS Host Access Transformation Service

HL7 Health Level 7

HR human resources

HTM Human Task Manager

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

I

ID identification

IDE integration development environment

IETF Internet Engineering Task Force

IID instance ID

IOP Internet Inter-ORB Protocol

IMAP Internet Message Access Protocol

IMS Information Management System

IMS TM Information Management System Transaction Manager

I/O input/output

IP Internet Protocol

IT information technology

J

J2C J2EE Connector architecture

J2CA J2EE Connector Architecture

J2EE Java 2 Platform, Enterprise Edition

Jacl Java Command Language

JAR Java archive

JAX Java API for XML

JAX-RPC Java API for XML-based RPC

JAX-WS Java API for XML Web Services

JCA Java EE Connector Architecture

JDBC Java Database Connectivity

JDT Java development tools

JMS Java Message Service

JNDI Java Naming and Directory Interface

JPEG Joint Photographic Experts Group

JSF JavaServer Faces

JSON JavaScript Object Notation

JSP JavaServer Pages

JSR Java Specification Request

JVM Java virtual machine

K

KPI key performance indicator

L

LDAP Lightweight Directory Access Protocol

LIFO last-in first-out

M

MBean managed bean

MDB message-driven bean

MIME Multipurpose Internet Mail Extensions

MQ	Message Queue
MQMD	MQ Message Descriptor
MTOM	Message Transmission Optimization Mechanism

N

ND	Network Deployment
-----------	--------------------

O

OAGIS	Open Applications Group Integration Specification
OASIS	Organization for the Advancement of Structured Information Standards
ODA	Object Discovery Agent
ODM	Object Data Manager
OLTP	online transaction processing
ORB	Object Request Broker
OS	operating system
OSOA	open service-oriented architecture
OSS	operations support systems

P

PC	personal computer
PDF	Portable Document Format
PI	project interchange
PNG	Portable Network Graphics
POJO	plain old Java object
POP3	Post Office Protocol

Q

QA	quality assurance
QoS	quality of service

R

RAR	resource adapter archive
RAS	Reusable Asset Specification
REST	Representational State Transfer
RFC	request for comments

RMI	Remote Method Invocation
RMI/IOP	Remote Method Invocation over Internet InterORB Protocol
RPC	Remote Procedure Call
RPG	Report Program Generator
RSS	Really Simple Syndication
RUP	Rational Unified Process
 S	
SACL	State Adaptive Choreography Language
SAP	Systems Applications and Products (data processing)
SCA	Service Component Architecture
SCDL	Service Component Definition Language
SDK	software development kit
SDO	Service Data Object
SIB	service integration bus
SIBus	service integration bus
SLA	service level agreement
SMB	small and medium-sized business
SMO	service message object
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
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
SPI	service provider interface
SQL	Structured Query Language
SSL	Secure Sockets Layer
SSO	single sign-on
STP	straight-through processing
SVG	Scalable Vector Graphics
SwaRef	SOAP with attachments, referenced

T

TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TIBCO	The Information Bus Company
TIFF	Tag Image File Format
TLS	Transport Layer Security
TM	Transaction Manager

U

UDB	Universal Database
UDDI	Universal Description, Discovery, and Integration
UDP	User Datagram Protocol
UI	user interface
UML	Unified Modeling Language
UNIX	Uniplexed Information and Computing System
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTC	Universal Test Client
UTE	unit test environment
UTF	Unicode Transformation Format
UTF-8	Unicode Transformation Format (8 bit)

V

VM	virtual machine
VMM	virtual member manager

W

W3C	World Wide Web Consortium
WAN	wide area network
WAR	web archive
WebDAV	Web-based Distributed Authoring and Versioning
WLM	workload management
WS	web services
WS-BPEL	Web Services Business Process Execution Language
WSDL	Web Services Description Language

WS-I	Web Services Interoperability
WSIL	Web Service Inspection Language
WS-N	Web Services Notification
WS-Security	Web Services Security
WWW	World Wide Web

X

XA	Extended Architecture
XCT	cross-component tracing
XMI	XML metadata interchange
XML	Extensible Markup Language
XML4J	XML Parser for Java
XOP	XML-binary Optimized Packaging
XPath	XML Path Language
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation

Z

z/OS	zSeries operating system
-------------	--------------------------

Appendix F. Resource guide

Completing this WebSphere Education course is a great first step in building your WebSphere, CICS, and SOA skills. Beyond this course, IBM offers several resources to keep your WebSphere skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
 - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
 - For more information, see: <http://www.ibm.com/training>
- **IBM Training News**
 - Review or subscribe to updates from IBM and its training partners.
 - For more information, see: <http://bit.ly/IBMTTrainEN>
- **IBM Certification**
 - You can demonstrate to your employer or clients your new WebSphere, CICS, or SOA mastery through achieving IBM Professional Certification. WebSphere certifications are available for developers, administrators, and business analysts.
 - For more information, see: <http://www.ibm.com/certify>
- **Training paths**
 - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many WebSphere products and roles, including developers and administrators.
 - For more information, see:
<http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&c=a0003096>

Social media links

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

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

- **Facebook:**

- Become a fan of IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
- Find IBM Training at: facebook.com/ibmtraining

- **YouTube:**

- Visit the IBM Training YouTube channel to learn about IBM training programs and courses.
- Find IBM Training at: youtube.com/IBMTraining

Support

- **WebSphere Support portal**

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

- **IBM Support Assistant**

- The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
- IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
- For more information, see: <http://www.ibm.com/software/support/isa>

- **WebSphere Education Assistant**

- IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your WebSphere Education course.
- For more information, see: <http://www.ibm.com/software/info/education/assistant/>

WebSphere documentation and tips

- **IBM Redbooks**
 - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many WebSphere products. Often, Redbooks include sample code and other support materials available as downloads from the site.
 - For more information, see: <http://www.ibm.com/redbooks>
- **IBM documentation and libraries**
 - Information centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The information centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain WebSphere products. The WebSphere information center and library are located conveniently in the left navigation on WebSphere product web pages.
- **developerWorks**
 - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the WebSphere zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
 - For more information, see: <http://www.ibm.com/developerworks>

WebSphere Services

- IBM Software Services for WebSphere are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The WebSphere Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for WebSphere makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see: <http://www.ibm.com/developerworks/websphere/services/>

IBM
®