

## Course Guide

# Developing with Integration Designer in IBM Business Process Manager V8.6 - I

Course code WB872G ERC 1.0



## 2018 Date of publication edition

### NOTICES

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:  
**INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.** Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

### TRADEMARKS

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

Adobe, and the Adobe logo, are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

**© Copyright International Business Machines Corporation 2018.**

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

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

---

# Contents

---

<b>Preface .....</b>	<b>P-1</b>
Contents .....	P-3
Course overview .....	P-21
Document conventions .....	P-23
Exercises .....	P-24
Additional training resources .....	P-25
IBM product help.....	P-26
<b>Unit 1   Introduction to IBM Business Process Manager V8.6 .....</b>	<b>1-1</b>
How to check online for course material updates .....	1-3
Unit objectives .....	1-4
Topics .....	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
IBM Business Process Manage .....	1-17
IBM Business Process Manager.....	1-18
Business integration roles in IBM BPM development .....	1-19
IBM Business Process Manager.....	1-20
Major tools in IBM Business Process Manager .....	1-21
IBM Process Designer .....	1-22
IBM Integration Designer .....	1-23
IBM Process Center.....	1-24
Process Portal .....	1-25
IBM Process Server (1 of 2) .....	1-27
IBM Process Server (2 of 2) .....	1-28
IBM Process Server components .....	1-29
User roles and tools in a typical deployment scenario.....	1-30
Introduction to IBM BPM on Cloud .....	1-31
BPM on Cloud customer focus: Manage and automate decisions .....	1-32
IBM BPM on Cloud: Three runtime environments .....	1-33
IBM BPM on Cloud user portal .....	1-34

Unit summary.....	1-35
Checkpoint questions .....	1-36
Checkpoint answers .....	1-37
References .....	1-38
<b>Unit 2 Overview of IBM Business Process Manager.....</b>	<b>2-1</b>
Unit objectives .....	2-3
Topics .....	2-4
BPD and BPEL .....	2-5
Two kinds of business processes: BPD and BPEL .....	2-6
BPMN .....	2-7
BPMN process in Process Designer (desktop interface) .....	2-9
BPMN process in Process Designer (web-based interface) .....	2-11
BPEL.....	2-12
BPEL process in IBM Integration Designer .....	2-13
When to use BPMN? .....	2-14
When to use BPEL? .....	2-15
Key concepts and capabilities .....	2-17
Service Component Architecture (SCA) .....	2-18
Business value of Service Component Architecture (SCA) .....	2-19
Business rules .....	2-20
Business value of rules.....	2-21
Adapters .....	2-22
Business value of adapters.....	2-23
Mediations .....	2-24
Business value of mediations .....	2-25
Unit summary.....	2-26
Checkpoint questions .....	2-27
Checkpoint answers .....	2-28
<b>Unit 3 Course business scenario: What you build.....</b>	<b>3-1</b>
Unit objectives .....	3-3
Some important points.....	3-4
WB872 lab environment .....	3-5
Exercise 1: Exploring IBM Integration Designer, part I .....	3-6
What are some key tasks that you do in Exercise 1? .....	3-7
Components that are required for Exercise 1 .....	3-8
Explore IBM Integration Designer in Exercise 1 .....	3-9
Course business scenario: What you work with in Ex 1 .....	3-10
Exercise 2: Exploring IBM Integration Designer, part II .....	3-11
What are some key tasks that you do in Exercise 2? .....	3-12
Components that are required for Exercise 2 .....	3-13

Use cross-component tracing in Exercise 2 .....	3-14
Test by using the Business Process Choreographer Explorer .....	3-15
Course business scenario: What you work with in Ex 2 .....	3-16
Exercise 3: Working with web services.....	3-17
What are some key tasks that you do in Exercise 3? .....	3-18
Components that are required for Exercise 3 (Section 1).....	3-19
Invoke an external web service in Exercise 3 (Section 1).....	3-20
Components that are required for Exercise 3 (Section 2).....	3-21
Expose an application as a web service that is running in IBM Process Server in Exercise 3 (Section 2).....	3-22
Course business scenario: What you work with in Ex 3 .....	3-23
Exercise 4: Creating business objects and shared interfaces .....	3-24
What are some key tasks that you do in Exercise 4? .....	3-25
Components that are required for Exercise 4 (1 of 2).....	3-26
Components that are required for Exercise 4 (2 of 2).....	3-27
Course business scenario: What you work with in Ex 4 .....	3-28
Exercise 5: Creating a business process, part I .....	3-29
What are some key tasks that you do in Exercise 5? .....	3-30
Account verification process: Overview .....	3-31
Components that are required for Exercise 5.....	3-33
Course business scenario: What you work with in Ex 5 .....	3-41
Exercise 6: Creating a business process, part II .....	3-42
What are some key tasks that you do in Exercise 6? .....	3-43
Components that are required for Exercise 6 .....	3-44
Code snippets that are created in the AccountVerification process in Exercise 6 .....	3-47
Complete Account Verification process .....	3-48
Course business scenario: What you work with in Ex 6 .....	3-52
Exercise 7: Creating a business process, part III .....	3-53
What are some key tasks that you do in Exercise 7? .....	3-54
Components that are required for Exercise 7 .....	3-55
Create data map in Exercise 7 .....	3-56
Assemble SCA application that contains the business process in Exercise 7 ...	3-58
Test the created application by deploying to a Process Server runtime .....	3-59
Use the test client in Exercise 7.....	3-60
Course business scenario: What you work with in Ex 7 .....	3-61
Exercise 8: Creating business rules .....	3-62
What are some key tasks that you do in Exercise 8? .....	3-63
Components that are required for Exercise 8 .....	3-64
Create rule group in Exercise 8 .....	3-65
Course business scenario: What you work with in Ex 8 .....	3-66
Exercise 9: Implementing WebSphere (JCA) adapters .....	3-67

What are some key tasks that you do in Exercise 9? .....	3-68
Components that are required for Exercise 9 .....	3-69
Assemble SCA components in Exercise 9 .....	3-71
Course business scenario: What you work with in Ex 9 .....	3-73
Exercise 10: Creating mediation services, part I .....	3-74
What are some key tasks that you do in Exercise 10? .....	3-75
Components that are required for Exercise 10 .....	3-76
Assemble SCA components in Exercise 10 .....	3-77
Course business scenario: What you work with in Ex 10 .....	3-79
Exercise 11: Creating mediation services, part II .....	3-80
What are some key tasks that you do in Exercise 11? .....	3-81
Components that are required for Exercise 11 .....	3-82
Create the RouteRequest mediation flow in Exercise 11 .....	3-84
Assemble SCA components in Exercise 11 .....	3-85
Course business scenario: What you work with in Ex 11 .....	3-87
Exercise 12: Exploring Business Space .....	3-88
What are some key tasks that you do in Exercise 12? .....	3-89
Account verification process: Overview .....	3-90
Create business space in Exercise 12.....	3-91
Course business scenario: What you work with in Ex 12 .....	3-92
Exercise 13: Using component tests .....	3-93
What are some key tasks that you do in Exercise 13? .....	3-94
Components that are required for Exercise 13 .....	3-95
Course business scenario: What you work with in Ex 13 .....	3-96
Exercise 14: Exploring IBM Process Center .....	3-97
What are some key tasks that you do in Exercise 14? .....	3-98
Components that are required for Exercise 14 .....	3-99
Course business scenario: What you work with in Ex 14 .....	3-100
Exercise 15: Implementing Advanced Integration services .....	3-101
What are some key tasks that you do in Exercise 15? .....	3-102
Course business scenario: What you work with in Ex 15 .....	3-103
Course business scenario: What you work with in Ex 15 .....	3-104
Unit summary.....	3-105

<b>Unit 4 IBM Integration Designer overview .....</b>	<b>4-1</b>
Unit objectives .....	4-3
Topics .....	4-4
Introduction to IBM Integration Designer .....	4-5
IBM Integration Designer: Create, assemble, and test applications .....	4-6
IBM Integration Designer roles .....	4-7
Installing IBM Business Process Manager V8.6 .....	4-9
IBM Installation Manager .....	4-11
IBM Integration Designer: Platform architecture .....	4-13
Installing IBM Integration Designer .....	4-15
Application composition and deployment .....	4-17
Getting started .....	4-18
Application composition and deployment .....	4-19
Business Integration perspective .....	4-20
Features overview: Visual tools .....	4-22
IBM Process Server unit test environment .....	4-26
Module .....	4-30
Module components .....	4-31
Mediation Module .....	4-32
Mediation Module components .....	4-33
Library components .....	4-35
Creating modules and libraries .....	4-36
Library deployment .....	4-37
Business Integration view .....	4-38
Dependency editor .....	4-39
Packaging modules .....	4-40
Module deployment .....	4-41
Application, module, and library relationship .....	4-43
Using the process center perspective .....	4-44
Building projects .....	4-45
Artifacts in IBM Integration Designer and Process Designer .....	4-46
Documentation and organization capabilities .....	4-47
Documentation and organization capabilities .....	4-48
Documentation: Sticky notes .....	4-49
Documentation: Java task tags .....	4-50
Organization: Integration solution .....	4-51

Unit summary.....	4-53
Checkpoint questions .....	4-54
Checkpoint answers .....	4-55
Exercise 1: Exploring IBM Integration Designer, part I .....	4-56
Course business scenario: What you work with in Ex 1 .....	4-57
Components that are required for Exercise 1 .....	4-58
Explore IBM Integration Designer in Exercise 1 .....	4-59
References .....	4-96
<b>Unit 5 Service Component Architecture overview.....</b>	<b>5-1</b>
Unit objectives .....	5-3
Topics .....	5-4
Overview of SCA modules.....	5-5
Service Component Architecture (SCA) .....	5-6
Overview and business value of SCA.....	5-7
SCA components.....	5-9
SCA component definition .....	5-11
SCA component interfaces .....	5-12
SCA component references.....	5-14
SCA module: Overview.....	5-16
References and wires .....	5-17
Exports.....	5-18
Imports.....	5-19
Example: Using imports and exports .....	5-21
SCA module components: Stand-alone references.....	5-22
Example: HelloWorld .....	5-24
Assembly Diagram: Components .....	5-25
Composing SCA modules: Assembly Diagram editor.....	5-26
Testing SCA applications.....	5-28
Testing SCA components in the integration test client .....	5-29
Test configurations .....	5-31
Emulators.....	5-32
Monitors .....	5-33
Entering data for testing.....	5-37
Using a test data table for testing .....	5-38
Using a stand-alone data pool for testing .....	5-39
Overview: Testing SCA components .....	5-40

Testing SCA modules in the integration test client .....	5-43
Overview: Testing SCA modules .....	5-44
Using the Servers view when testing modules .....	5-47
Server Logs view .....	5-48
Cross-component tracing.....	5-49
Unit summary.....	5-51
Checkpoint questions .....	5-52
Checkpoint answers .....	5-53
Exercise 2: Exploring IBM Integration Designer, part II .....	5-54
Course business scenario: What you work with in Ex 2 .....	5-55
Components that are required for Exercise 2 .....	5-56
Use cross-component tracing in Exercise 2 .....	5-57
Test by using the Business Process Choreographer Explorer .....	5-58
<b>Unit 6 SCA bindings.....</b>	<b>6-1</b>
Unit objectives .....	6-3
Topics .....	6-4
SCA bindings .....	6-5
Incoming and outgoing interactions with external components .....	6-6
SCA invocation styles .....	6-8
Import and export resources .....	6-9
Quality of service qualifiers.....	6-10
Export binding types .....	6-11
Import binding types .....	6-12
JCA adapter EIS bindings.....	6-13
Enterprise JavaBeans (EJB) binding .....	6-16
WSDL to Java interface mapping for EJB beans.....	6-18
SCA bindings .....	6-19
JMS bindings .....	6-21
Generic JMS bindings.....	6-23
WebSphere MQ bindings.....	6-24
HTTP bindings .....	6-26
Web services .....	6-29
Overview: Web services .....	6-30
Discovering web services .....	6-32
Importing a web service WSDL interface.....	6-34
Web service export binding .....	6-35
Testing web services .....	6-39
Unit summary.....	6-40
Checkpoint questions .....	6-41
Checkpoint answers .....	6-42
Exercise objectives .....	6-43

Course business scenario: What you work with in Ex 3 .....	6-44
Components that are required for Exercise 3 (Section 1).....	6-45
Invoke an external web service in Exercise 3 (Section 1).....	6-47
Components that are required for Exercise 3 (Section 2).....	6-48
Expose an application as a web service that is running in IBM Process Server in Exercise 3 (Section 2).....	6-49
References .....	6-80
<b>Unit 7 Business objects.....</b>	<b>7-1</b>
Unit objectives .....	7-3
Topics .....	7-4
Introduction to Service Data Objects .....	7-5
Introduction to Service Data Objects (SDO) .....	7-6
Data objects.....	7-8
Data graphs .....	7-9
Example: SDO data tree.....	7-10
Introduction to business objects .....	7-11
Business objects are an SOA core component .....	7-12
Introduction to the business object framework .....	7-13
Business objects that are represented as SDOs .....	7-14
Business objects: Data abstraction for SCA .....	7-15
Composition of business object framework .....	7-16
Business objects.....	7-19
Creating business objects.....	7-22
Business graphs .....	7-24
Business graph: Change summary.....	7-25
Data objects in IBM Business Process Manager.....	7-27
Unit summary.....	7-28
Checkpoint questions .....	7-29
Checkpoint answers .....	7-30
Supplemental note: Business object framework runtime environment .....	7-31
Exercise 4: Creating business objects and shared interfaces .....	7-32
Course business scenario: What you work with in Ex 4 .....	7-33
Components that are required for Exercise 4 .....	7-34
References .....	7-45

<b>Unit 8 Business process choreography overview .....</b>	<b>8-1</b>
Unit objectives .....	8-3
Topics .....	8-4
Introduction to Web Services Business Process Execution Language (WS-BPEL) .....	8-5
Business processes are service components.....	8-6
Business processes in IBM Business Process Manager.....	8-7
Overview: Business Process Execution Language .....	8-9
BPEL interoperates with SCA and SDO .....	8-12
Microflows versus long-running processes.....	8-13
Elements of BPEL processes .....	8-16
Main elements and concepts of a BPEL process .....	8-17
Elements of a business process defined .....	8-18
Process elements: Partners.....	8-19
Process elements: Variables .....	8-20
Process elements: Correlation sets .....	8-22
Process elements: Activities .....	8-25
Unit summary.....	8-26
Checkpoint questions .....	8-27
Checkpoint answers .....	8-28
Exercise 5: Creating a business process, part I .....	8-29
Course business scenario: What you work with in Ex 5 .....	8-30
Account verification process: Overview .....	8-31
Components that are required for Exercise 5 .....	8-32
<b>Unit 9 Business process basic and structured activities.....</b>	<b>9-1</b>
Unit objectives .....	9-3
Topics .....	9-4
WS-BPEL basic activities .....	9-5
Overview of WS-BPEL activities.....	9-6
Basic activities: Receive, reply, and invoke .....	9-7
Basic activities: Assign .....	9-8
Basic activities: Receive choice .....	9-9
Basic activities: Wait .....	9-11
Basic activities: Human task .....	9-12
Basic activities: Snippets .....	9-13
Basic activities: Empty action .....	9-14
Basic activities: Data map.....	9-15
Creating a data map .....	9-16
Common XPath editor .....	9-17
Activity exit conditions.....	9-18

WS-BPEL structured activities .....	9-19
Structured activities: Sequences and parallel activities .....	9-20
Links and link conditions .....	9-21
Structured activities: Choice .....	9-22
Structured activities: While loop .....	9-23
Structured activities: Repeat until loop .....	9-24
Structured activities: For each .....	9-25
Structured activities: Generalized flow .....	9-26
Structured activities: Scope .....	9-29
Structured activities: Collaboration scope .....	9-30
Unit summary .....	9-32
Checkpoint questions .....	9-33
Checkpoint answers .....	9-34
Exercise 6: Creating a business process, part II .....	9-35
Course business scenario: What you work with in Ex 6 .....	9-36
Complete Account verification process .....	9-37
Components that are required for Exercise 6 .....	9-41
Code snippets that are created in the AccountVerification process in Exercise 6 .....	9-44

## **Unit 10 Business process handlers, runtime behavior, and clients..... 10-1**

Unit objectives .....	10-3
Topics .....	10-4
WS-BPEL handlers and error processing activities .....	10-5
Handlers and error processing activities: Throw and rethrow .....	10-6
Handlers and error processing activities: Fault handler .....	10-8
When a fault is thrown .....	10-9
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
Runtime aspects of BPEL processes .....	10-19
Business process runtime support .....	10-20
Business process templates and instances .....	10-22
Microflow and long-running processes at run time .....	10-23
Clients for business processes .....	10-25
Business Process Choreographer Explorer client .....	10-27
Runtime process dynamicity (1 of 2) .....	10-29
Runtime process dynamicity (2 of 2) .....	10-30
Installing and uninstalling business process applications .....	10-31
Deletion of completed instances .....	10-33

Unit summary.....	10-34
Checkpoint questions .....	10-35
Checkpoint answers .....	10-36
Exercise 7: Creating a business process, part III .....	10-37
Course business scenario: What you work with in Ex 7 .....	10-38
Components that are required for Exercise 7 .....	10-39
Create data map in Exercise 7 (1 of 2) .....	10-40
Create data map in Exercise 7 (2 of 2) .....	10-41
Assemble the SCA application that contains the business process in Exercise 7 .....	10-42
Test the created application by deploying to a Process Server runtime .....	10-43
Use the test client in Exercise 7.....	10-44
References .....	10-84
<b>Unit 11 Business rules.....</b>	<b>11-1</b>
Unit objectives .....	11-3
Topics .....	11-4
Overview of business rules.....	11-5
Business rules are a service component .....	11-6
What is a business rule?.....	11-7
Business rule actions.....	11-8
Business value of rules.....	11-9
Rule sets defined .....	11-11
Rule sets in IBM Integration Designer .....	11-12
Decision table .....	11-15
Decision tables in IBM Integration Designer.....	11-16
Rule sets versus decision tables .....	11-19
Rule groups .....	11-20
Rule group components.....	11-22
Business rule group properties .....	11-23
Display names and description fields for rule groups .....	11-24
Business rules: Tools.....	11-25
Business rules in IBM Process Designer.....	11-27
Runtime aspects of rule groups .....	11-28
Business rules at run time .....	11-29
More runtime aspects of business rules .....	11-30
Designer and developer roles in business rules .....	11-31
Business rules manager web client .....	11-32
Templates in the business rules manager .....	11-33
Exporting and importing rule changes .....	11-34
Business rule auditing.....	11-35
Public business rules API .....	11-36

Unit summary.....	11-37
Checkpoint questions .....	11-38
Checkpoint answers .....	11-39
Exercise 8: Creating business rules .....	11-40
Course business scenario: What you work with in Ex 8 .....	11-41
Components that are required for Exercise 8 .....	11-42
Create rule group in Exercise 8 .....	11-43
References .....	11-75
<b>Unit 12 Adapters.....</b>	<b>12-1</b>
Unit objectives .....	12-3
Topics .....	12-4
Introduction to adapters .....	12-5
Adapters are supporting services .....	12-6
Introduction to adapters .....	12-7
Adapter components.....	12-8
WebSphere (JCA) Adapters .....	12-10
Available WebSphere (JCA) application adapters.....	12-11
Architecture.....	12-12
Available WebSphere (JCA) technology adapters.....	12-13
Advantages of Java EE Connector Architecture.....	12-14
Using WebSphere (JCA) Adapters .....	12-17
Using WebSphere Adapters .....	12-18
Enterprise Metadata Discovery.....	12-19
Discovered interactions .....	12-22
Data Exchange Service Provider Interface.....	12-23
Application adapters and event polling .....	12-24
Unit summary.....	12-25
Checkpoint questions .....	12-26
Checkpoint answers .....	12-27
Exercise 9: Implementing WebSphere (JCA) adapters .....	12-28
Course business scenario: What you work with in Ex 9 .....	12-29
Components that are required for Exercise 9 .....	12-30
Assemble SCA components in Exercise 9 .....	12-32
References .....	12-61

<b>Unit 13 Developing mediation services.....</b>	<b>13-1</b>
Unit objectives .....	13-3
Topics .....	13-4
Mediation services .....	13-5
Mediation flows are supporting services.....	13-6
What does an enterprise service bus do? .....	13-7
Key concepts .....	13-9
Example: Mediation module and mediation flow .....	13-10
Typical task to implement a mediation flow .....	13-11
Mediation module and mediation flow: Tools.....	13-12
Mediation flow editor: Overview tab .....	13-14
Mediation flow editor: Request flow view .....	13-15
Mediation flow editor: Response flow view .....	13-16
Mediation flow editor: Correspondence between Overview and Flow views ...	13-17
Terminals in the mediation flow editor .....	13-18
Wiring of mediation primitive terminals .....	13-19
Example of wiring of mediation primitive terminals.....	13-20
Mediation flow editor: Properties view .....	13-21
Mediations subflow: Overview .....	13-22
Mediations subflow: Implementation.....	13-24
Promoted properties: Overview .....	13-25
Promoted properties: Administrative management.....	13-26
Service message objects.....	13-27
Overview: SDO, SMO, business objects .....	13-28
Service message object (SMO) .....	13-29
SMO structure: Headers .....	13-30
SMO structure: Correlation and transient contexts .....	13-31
SMO structure: FailInfo context .....	13-32
SMO structure: Body .....	13-33
SMO in a mediation flow.....	13-34
Manipulating SMOs .....	13-36
Unit summary.....	13-38
Checkpoint questions .....	13-39
Checkpoint answers .....	13-40
Exercise 10: Creating mediation services, part I .....	13-41
Course business scenario: What you work with in Ex 10 .....	13-42
Components that are required for Exercise 10 .....	13-43
Assemble SCA components in Exercise 10 .....	13-44

<b>Unit 14 Mediation primitives.....</b>	<b>14-1</b>
Unit objectives .....	14-3
Topics .....	14-4
Message transformation and enrichment primitives .....	14-5
What is a mediation primitive?.....	14-6
Common features of mediation primitives .....	14-8
Mediation primitives: Transformation.....	14-9
Mapping mediation primitive .....	14-11
Message Element Setter primitive .....	14-13
Message Header Setter primitives.....	14-14
Custom Mediation primitive .....	14-15
Flow control primitives .....	14-16
Mediation primitives: Routing.....	14-17
Message Filter primitive .....	14-19
Message Filter primitive: Configuration .....	14-20
Service Invoke primitive: Overview .....	14-22
Service Invoke primitive compared to callout node .....	14-23
Type Filter primitive .....	14-24
Faults, tracing, and error handling primitives .....	14-25
Faults .....	14-26
WSDL faults: Modeled .....	14-27
WSDL faults: Unmodeled .....	14-28
Handling WSDL faults: Request flow .....	14-29
Handling WSDL faults: Response flow .....	14-30
Mediation primitives: Error handling, debugging, and event recording .....	14-31
Message Logger primitive.....	14-32
Trace primitive .....	14-34
Stop primitive .....	14-35
Fail primitive.....	14-36
Unit summary.....	14-37
Checkpoint questions .....	14-38
Checkpoint answers .....	14-39
Exercise 11: Creating mediation services, part II .....	14-40
Course business scenario: What you work with in Ex 11 .....	14-41
Components that are required for Exercise 11 .....	14-42
Create the RouteRequest mediation flow in Exercise 11 .....	14-44
Assemble SCA components in Exercise 11 .....	14-45

<b>Unit 15 Business Space.....</b>	<b>15-1</b>
Unit objectives .....	15-3
Topics .....	15-4
Introduction to Business Space .....	15-5
Problem statement that Business Space addresses .....	15-6
What is Business Space? .....	15-7
Business Space: A unified environment .....	15-8
Key roles (participants) in business process management.....	15-10
Integration with the BPM portfolio .....	15-11
Business Space terminology .....	15-12
Business Space hierarchy .....	15-13
Pages.....	15-14
Templates .....	15-16
Templates available in IBM Process Manager .....	15-18
Business space.....	15-20
Widget.....	15-22
Problem determination widgets .....	15-24
Human Task Management widget .....	15-26
Tasks widget.....	15-27
WebSphere Service Registry and Repository widgets .....	15-28
IBM Business Monitor widgets.....	15-29
Configure a new space .....	15-30
Create a space .....	15-31
Customizing a business space .....	15-35
Working with business spaces.....	15-36
Unit summary.....	15-38
Checkpoint questions .....	15-39
Checkpoint answers .....	15-40
Exercise 12: Exploring Business Space .....	15-41
Course business scenario: What you work with in Ex 12 .....	15-42
Account verification process: Overview .....	15-43
Create business space in Exercise 12.....	15-45

<b>Unit 16 Advanced testing .....</b>	<b>16-1</b>
Unit objectives .....	16-3
Topics .....	16-4
Advanced testing .....	16-5
Testing modules .....	16-6
Unit testing: Integration test client.....	16-7
Integration test client: Events tab.....	16-8
Integration test client: Configurations tab .....	16-10
Component testing.....	16-12
Component test project.....	16-14
Test suite: Overview tab .....	16-16
Test suite: Test Cases tab .....	16-17
Test suite: Configurations tab .....	16-19
Running a test case .....	16-21
Cross-component trace .....	16-22
Fine-grained trace.....	16-26
Command-line test invocation .....	16-27
Stand-alone data pool.....	16-28
Component Test Explorer.....	16-29
Server Logs view .....	16-31
The IBM Guided Activity Assistant.....	16-33
Deploying integration applications with serviceDeploy .....	16-34
Overview of serviceDeploy .....	16-35
serviceDeploy: Single developer .....	16-39
Multiple developers and serviceDeploy .....	16-40
Unit summary.....	16-41
Checkpoint questions .....	16-42
Checkpoint answers .....	16-43
Exercise 13: Using component tests .....	16-44
Course business scenario: What you work with in Ex 13 .....	16-45
Components that are required for Exercise 13 .....	16-46

<b>Unit 17 IBM Process Center .....</b>	<b>17-1</b>
Unit objectives .....	17-3
Topics .....	17-4
Introduction to IBM Process Center.....	17-5
IBM Business Process Manager.....	17-6
IBM Process Center capabilities.....	17-7
Components of IBM Process Center .....	17-8
IBM Process Center Console .....	17-9
Accessing IBM Process Center .....	17-10
IBM Process Center: Process Apps .....	17-11
IBM Process Center: Toolkits .....	17-12
IBM Process Center: Snapshots.....	17-13
IBM Process Center: Servers .....	17-14
IBM Process Center: Admin .....	17-15
Manage, test, and deploy IBM BPM applications .....	17-16
Importing from the IBM Process Center repository.....	17-17
Associating a module or library.....	17-18
Publish and synchronize.....	17-19
IBM Process Designer artifacts in IBM Integration Designer.....	17-20
IBM Integration Designer artifacts in IBM Process Designer.....	17-21
Manage.....	17-22
Cleanup snapshots .....	17-23
Deploy.....	17-24
Typical development and deployment scenario.....	17-25
Unit summary.....	17-26
Checkpoint questions .....	17-27
Checkpoint answers .....	17-28
Exercise 14: Exploring the IBM Process Center .....	17-29
Course business scenario: What you work with in Ex 14 .....	17-30
Components that are required for Exercise 14 .....	17-31

<b>Unit 18 Advanced Integration services .....</b>	<b>18-1</b>
Unit objectives .....	18-3
Topics .....	18-4
AIS introduction .....	18-5
What is an Advanced Integration service? .....	18-6
Business value of an Advanced Integration service .....	18-7
Approaches in creation of AIS .....	18-8
Two approaches for working with AIS .....	18-9
Top-down approach.....	18-10
Step 1: Create process application in Process Designer.....	18-11
Step 2: Define inputs and outputs in AIS .....	18-12
Step 3: Implement the AIS.....	18-13
Step 4: Refresh and publish changes to Process Center .....	18-14
Bottom-up approach .....	18-15
Bottom-up approach, Step 1: Create export in Integration Designer .....	18-16
Bottom-up approach: Steps 2, 3, 4 .....	18-17
Unit summary.....	18-18
Checkpoint questions .....	18-19
Checkpoint answers .....	18-20
Exercise 15: Implementing Advanced Integration services .....	18-21
Course business scenario: What you work with in Ex 15 .....	18-22
<b>Appendix A IBM BPM on Cloud .....</b>	<b>A-1</b>
Introduction to IBM BPM on Cloud .....	A-3
Accelerate decision management solution deployment with IBM BPM on Cloud .....	A-4
BPM on Cloud customer focus: Manage and automate decisions .....	A-5
IBM BPM on Cloud: Three runtime environments .....	A-6
IBM BPM on Cloud free trial .....	A-7
Activating access and logging in to IBM BPM on Cloud .....	A-8
IBM BPM on Cloud user portal .....	A-9
Using the IBM Process Designer.....	A-12
Using the IBM Process Center .....	A-15
Using the IBM Process Portal.....	A-18
Using the IBM Integration Designer.....	A-19
Finding help for IBM BPM on Cloud .....	A-20

---

# Course overview

---

## Preface overview

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.

## Intended audience

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

## Topics covered

Topics covered in this course include:

- Introduction to IBM Business Process Manager
- Overview of IBM Business Process Manager
- Course business scenario: What you build
- IBM Integration Designer overview
- Service Component Architecture overview
- SCA bindings
- Business objects
- Business process choreography overview
- Business process basic and structured activities
- Business process handlers, runtime behavior, and clients
- Business rules
- Adapters
- Developing mediation services
- Mediation primitives
- Business Space
- Advanced testing
- Process Center
- Advanced Integration services
- Course summary

## Course prerequisites

Participants should have:

- 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 WB827, Process Implementing with IBM Business Process Manager V8.6 - I, is highly recommended, but not required

---

## Document conventions

---

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

- **Bold:** Bold style is used in demonstration and exercise step-by-step solutions to indicate a user interface element that is actively selected or text that must be typed by the participant.
- *Italic:* Used to reference book titles.
- **CAPITALIZATION:** All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application.  
To keep capitalization consistent with this guide, type text exactly as shown.

---

# Exercises

---

## Exercise format

Exercises are designed to allow you to work according to your own pace. Content contained in an exercise is not fully scripted out to provide an additional challenge. Refer back to demonstrations if you need assistance with a particular task. The exercises are structured as follows:

### The business question section

This section presents a business-type question followed by a series of tasks. These tasks provide additional information to help guide you through the exercise. Within each task, there may be numbered questions relating to the task. Complete the tasks by using the skills you learned in the unit. If you need more assistance, you can refer to the Task and Results section for more detailed instruction.

### The task and results section

This section provides a task based set of instructions that presents the question as a series of numbered tasks to be accomplished. The information in the tasks expands on the business case, providing more details on how to accomplish a task. Screen captures are also provided at the end of some tasks and at the end of the exercise to show the expected results.

---

## Additional training resources

---

Visit the [IBM Skills Gateway](http://www.ibm.com/training/) ([www.ibm.com/training/](http://www.ibm.com/training/)) for details on:

- Instructor-led training in a classroom or online
- Self-paced training that fits your needs and schedule
- Comprehensive curricula, learning journeys, and training paths that help you identify the courses that are right for you
- [IBM Professional Certification Program](http://www-03.ibm.com/certify/) (<http://www-03.ibm.com/certify/>)
- For other resources that will enhance your success, bookmark the [IBM Cloud Skills Gateway](https://www-03.ibm.com/services/learning/ites.wss/zz-en?pageType=page&c=O602980X82373O75) <https://www-03.ibm.com/services/learning/ites.wss/zz-en?pageType=page&c=O602980X82373O75>

# IBM product help

Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	<i>IBM Product - Help link</i>
Books for Printing (.pdf)	<p>You want to use search engines to find information. You can then print out selected pages, a section, or the whole book.</p> <p>Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book.</p> <p>The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.</p>	Start/Programs/ <i>IBM Product/Documentation</i>
IBM on the Web	<p>You want to access any of the following:</p> <ul style="list-style-type: none"> <li>• IBM Skills Gateway</li> <li>• Online support</li> <li>• IBM Web site</li> </ul>	<ul style="list-style-type: none"> <li>• <a href="https://www-03.ibm.com/services/learning/sites.wss/zz-en?pageType=page&amp;c=a0011023">https://www-03.ibm.com/services/learning/sites.wss/zz-en?pageType=page&amp;c=a0011023</a></li> <li>• <a href="https://www.ibm.com/support/home/">https://www.ibm.com/support/home/</a></li> <li>• <a href="http://www.ibm.com">http://www.ibm.com</a></li> </ul>

**Unit 1** Introduction to IBM Business Process Manager V8.6

The slide features a blue header bar with 'IBM Training' on the left and the 'IBM' logo on the right. The main content area has a light gray diagonal striped background. The title 'Introduction to IBM Business Process Manager V8.6' is centered in large blue text. Below it, the subtitle 'IBM Business Process Manager V8.6' is also in blue. At the bottom, a copyright notice reads: '© Copyright IBM Corporation 2018' and 'Course materials may not be reproduced in whole or in part without the written permission of IBM.'

**Introduction to IBM Business  
Process Manager V8.6**

IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## How to check online for course material updates



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

### Instructions

Enter this URL in your browser:  
<http://ibm.biz/CloudEduCourses>

On the wiki page, locate and click the Course Information category.

Find your course in the list and then click the link.

The wiki page displays information for the course. If an errata document is available, this page is where it is found.

If you want to download an attachment, such as an errata document, click the Attachments tab at the bottom of the page.

Comments (0) Versions (1) **Attachments (1)** About

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

## Unit objectives

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

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Unit objectives*

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.

## Topics

- Introduction to business process management
- IBM Business Process Manager

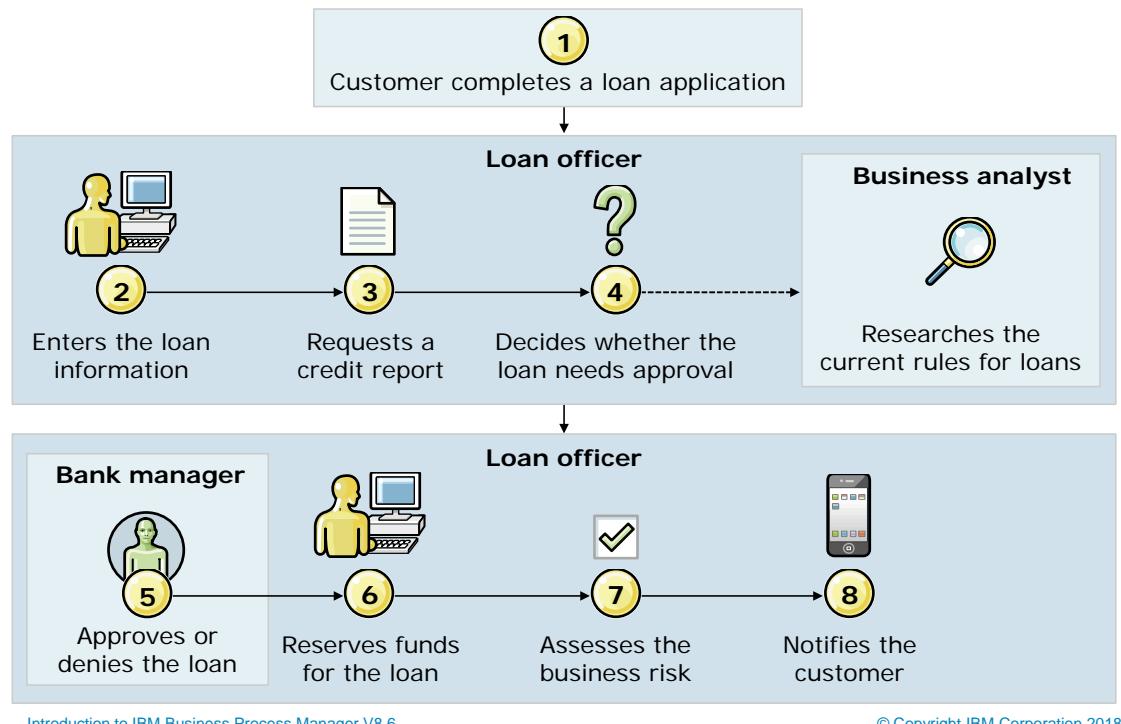
# Introduction to business process management

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

*Introduction to business process management*

## A typical manual system: Loan application



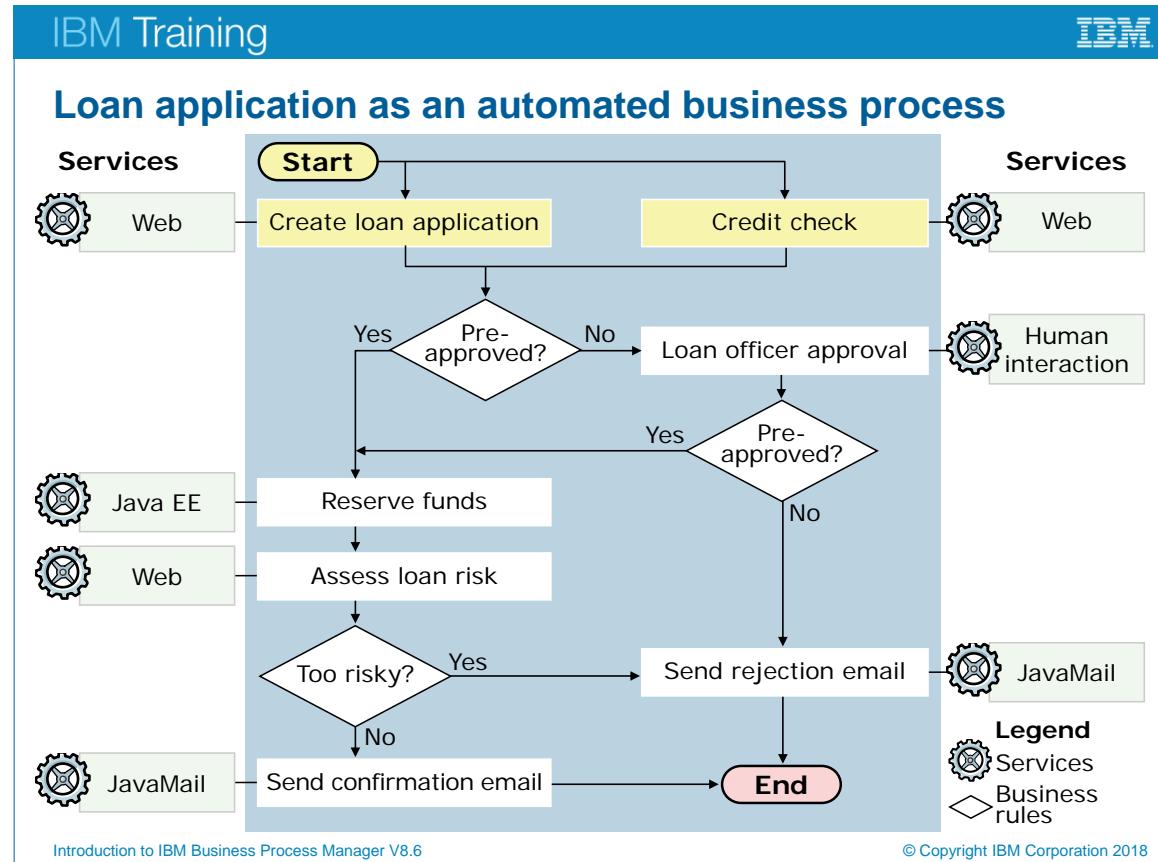
### A typical manual system: Loan application

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

## 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

*Challenges with the old system*



*Loan application as an automated business process*

## Services and service-oriented architecture

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

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Services and service-oriented architecture*

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

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

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

## Evolution of enterprise integration toward SOA

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

Point-to-point connectivity



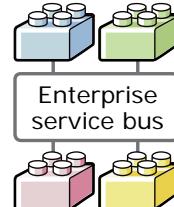
- Custom point-to-point integration between technology-aware bridges and application components
- **Point-to-point systems are brittle**

EAI hub-and-spoke



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

SOI with ESB



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

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Evolution of enterprise integration toward SOA*

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

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

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

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

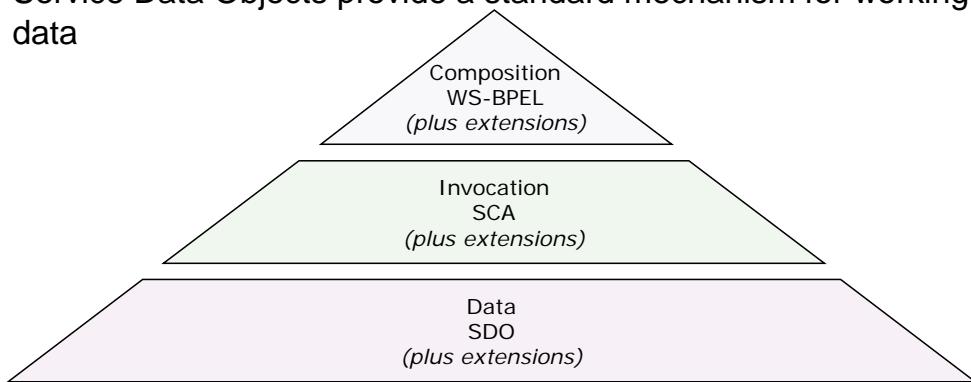
SOA blends the best of all these concepts into one new architecture, giving users capabilities that they never had before. Service-oriented integration (SOI) is the least brittle. SOI, if implemented by using web services, has standard interfaces and is much cheaper for new business requirements. If 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



Introduction to IBM Business Process Manager V8.6

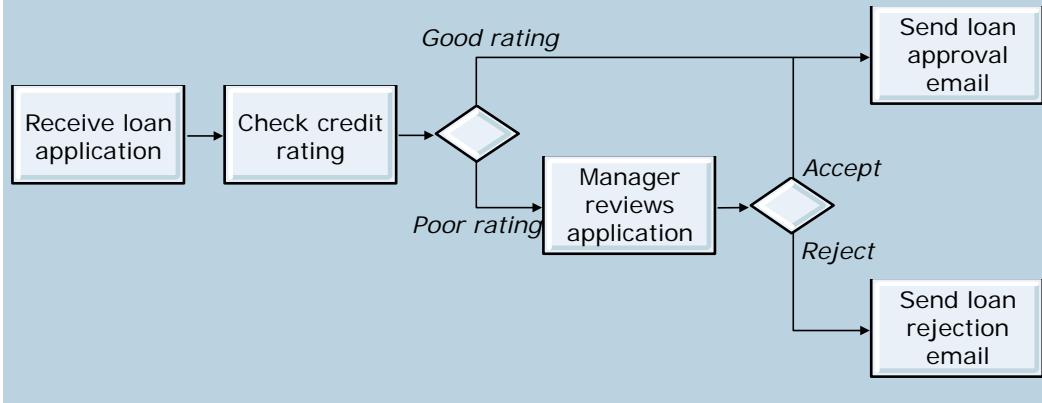
© Copyright IBM Corporation 2018

*SOA model: Open standards with IBM extensions*

## Business processes support business goals

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

Example:



Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Business processes support business goals*

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

A business process defines the potential execution order of services.

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

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

## 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

## 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

# IBM Business Process Manager

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

*IBM Business Process Manager*

## 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: An authoring environment for developing services and self-contained enterprise applications
- IBM Process Designer: An authoring environment for developing process models

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *IBM Business Process Manager*

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

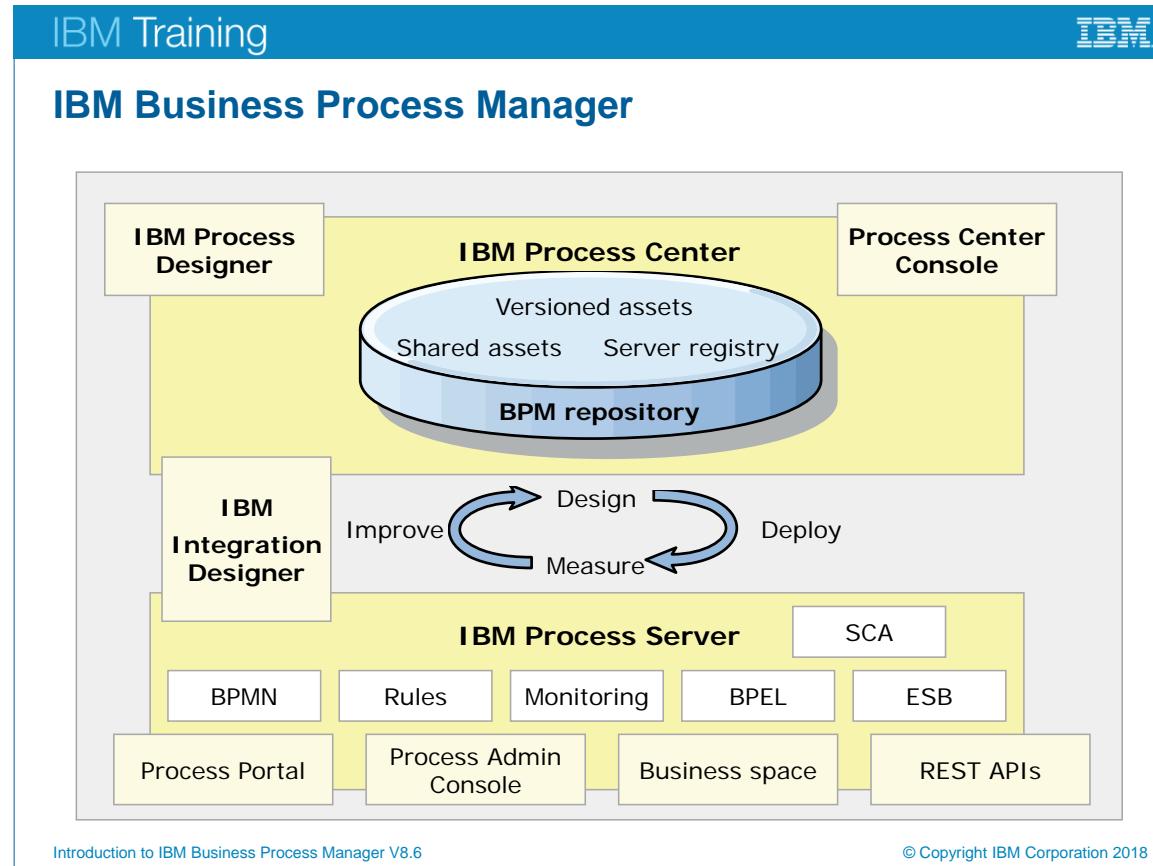
User role	Role definition
Integration developer	<ul style="list-style-type: none"> <li>Some basic programming experience (loops, conditionals, and string manipulation)</li> <li>Expects tools to simplify and abstract advanced IT implementation details</li> </ul>
Application developer (IT developer)	<ul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

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Business integration roles in IBM BPM development*

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



### IBM Business Process Manager

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

IBM Business Process Manager 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

- IBM Process Designer: An authoring environment that is used for creating process models
- 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

### *Major tools in IBM Business Process Manager*

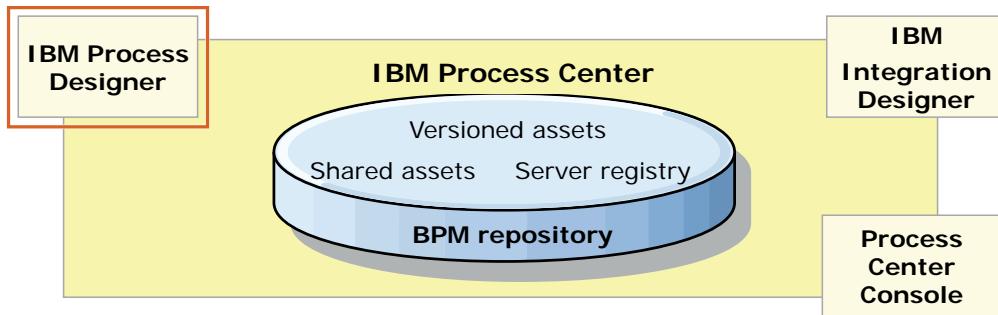
IBM Process Designer is an authoring environment that is used to create process models that contain automated and human tasks that are developed with the Business Process Model and Notation (BPMN) and Business Process Definition (BPD) formats.

IBM Integration Designer is an authoring environment that is used to create process models and 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)



### *IBM Process Designer*

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

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

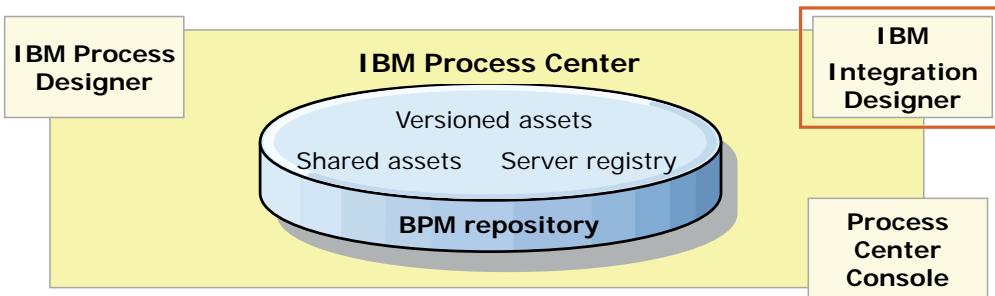
All Process Designer projects are contained in process applications. You store those process applications and associated artifacts in the Process Center repository.

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

- Development tool for building SCA-based integration applications
- Provides a visual development environment for developing, assembling, testing, deploying, and managing integration units and mediation units
- Uses Service Component Architecture (SCA) units and libraries, and Advanced Integration services (AIS)



Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *IBM Integration Designer*

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

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

The artifacts that are produced in IBM Integration Designer include:

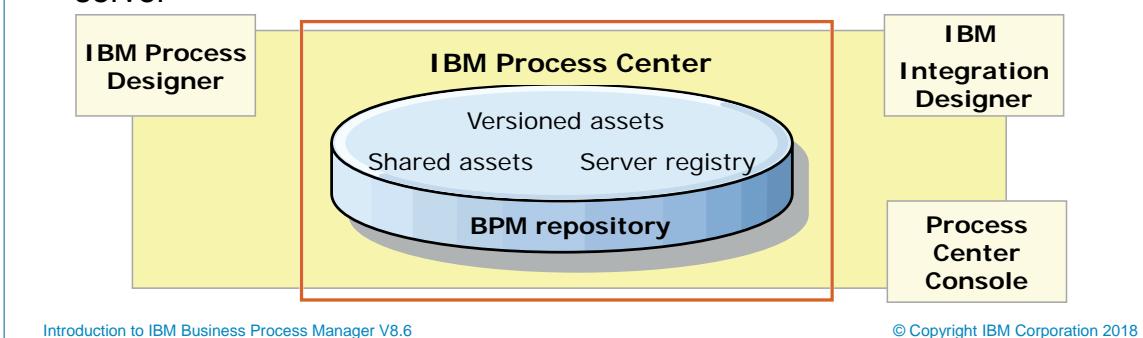
- SCA components that contain one or more modules and libraries and are deployable as EAR files
- Advanced Integration services that are traditional BPEL processes

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

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

## IBM Process Center

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



Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *IBM Process Center*

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

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

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

**Process Portal**

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

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### Process Portal

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

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

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

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

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

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
- Built on top of WebSphere Application Server
- WebSphere Application Server offers a fast, flexible, and secure Java application server runtime environment

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### IBM Process Server (1 of 2)

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

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

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

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

## 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

### IBM Process Server

Process layer builds on WebSphere Application Server to deliver robust business process management

### Choreography and mediation

### WebSphere Application Server

Network Deployment  
Highly available transaction engine

### Clustering

### WebSphere Application Server

### Application server

### *IBM Process Server (2 of 2)*

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

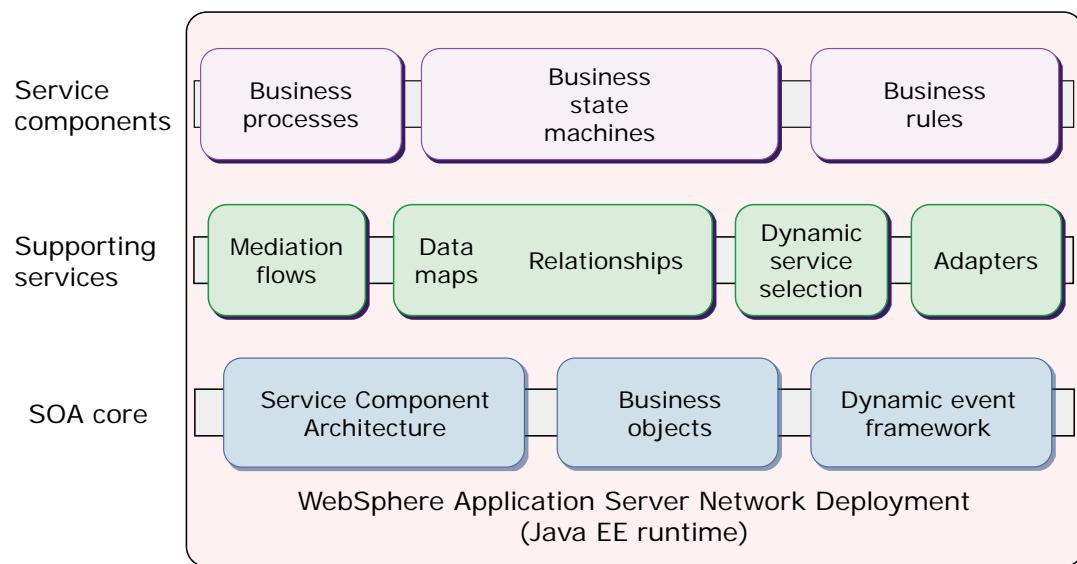
Although IBM Process Server is implemented on top of WebSphere Application Server, Java EE skills are not required for designing and implementing BPM solutions in IBM Process Server. IBM Process Server provides a higher level of abstraction, hiding its own implementation details.

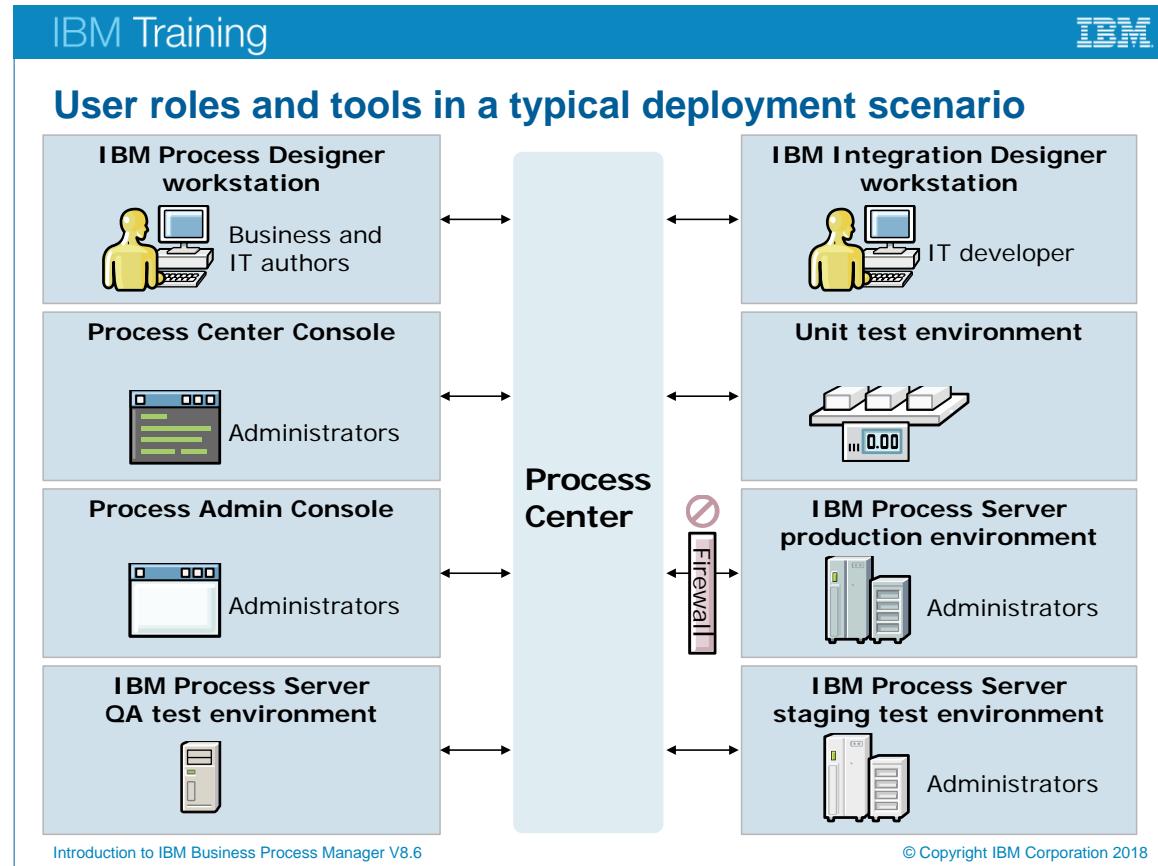
Each layer encapsulates and builds on the lower layer. Everything begins with the application server. The higher abstraction layers indirectly use the application server or WebSphere Application Server Network Deployment for security, user registry, transactions, scalability, clustering, high availability,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

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

## IBM Process Server components

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





### *User roles and tools in a typical deployment scenario*

The diagram displays a typical development and deployment topology.

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

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

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

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

## Introduction to IBM BPM on Cloud

- Enterprise-grade IBM BPM cloud service for development, testing, and production
- Cloud-based, collaborative, and role-based environment
- Capture, automate, and manage frequently occurring, repeatable rules-based business decisions
- Ready-to-use development, test, and production environments are available
- Monthly subscription plans
- Available exclusively on IBM Cloud infrastructure
- As of 2015, over 25 data centers are available worldwide
- Managed by IBM
- Artifacts that are created with IBM BPM on Cloud are compatible with IBM BPM on-premises product
- Free 30-day trial available (see Appendix A)

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

### *Introduction to IBM BPM on Cloud*

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

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

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:

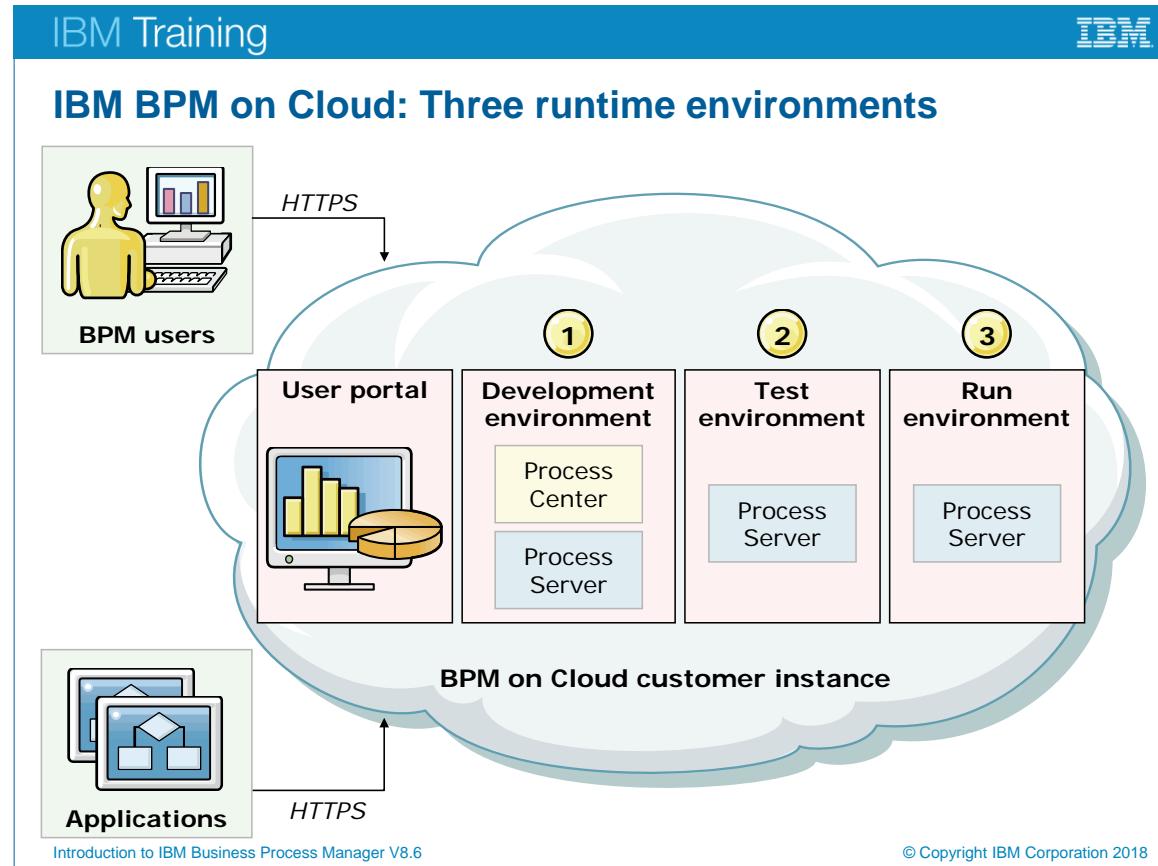
Uptime  
Monitoring  
Backup  
High availability  
Disaster recovery  
Updates  
Maintenance



### Customers manage:

Application development  
Application integration  
Application support





### IBM BPM on Cloud: Three runtime environments

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**

<b>Process Center</b> 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	<b>REST UI</b> Prototype IBM BPM REST resources and their associated parameters.  Launch More info	<b>Process Portal</b> Collaborate on tasks and view the performance of individuals, teams, and processes on dashboards.  Launch More info	<b>Process Admin Console</b> Manage the Process Center server and the process servers in your runtime environments.  Launch More info
<b>Tech Preview: Responsive Federated Portal</b> Technical demonstrations of Responsive Federated Portal and Responsive Coach Toolkit as a sample  Launch More info	<b>Business Process Choreographer Explorer</b> Monitor and manage BPEL processes. The BPC Explorer provides a number of views that show process and task metadata.  Launch More info	<b>Business Rules Manager</b> Manage business rules  Launch More info	

Introduction to IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018

*IBM BPM on Cloud user portal*

## Unit summary

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

## 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++.

*Checkpoint questions*

## 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).

## References

- IBM Business Process Manager product documentation:  
[https://www.ibm.com/support/knowledgecenter/SSFPJS\\_8.6.0/com.ibm.wbpm.main.doc/kc-homepage-bpm.html](https://www.ibm.com/support/knowledgecenter/SSFPJS_8.6.0/com.ibm.wbpm.main.doc/kc-homepage-bpm.html)
- BPM – Business Process Management:  
<http://www.ibm.com/software/info/bpm/>
- IBM Business Process Manager on developerWorks:  
<http://www.ibm.com/developerworks/websphere/zones/bpm/>

## **Unit 2** Overview of IBM Business Process Manager

The slide has a blue header bar with 'IBM Training' on the left and the IBM logo on the right. The main title 'Overview of IBM Business Process Manager' is centered in large blue font. Below it, the subtitle 'IBM Business Process Manager V8.6' is also in blue. At the bottom, there is a copyright notice: '© Copyright IBM Corporation 2018' and 'Course materials may not be reproduced in whole or in part without the written permission of IBM.'

IBM Training

**Overview of IBM Business Process Manager**

IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## Unit objectives

- Describe the differences between BPMN and BPEL
- Describe the capabilities of IBM Business Process Manager

### *Unit objectives*

This unit is an overview of IBM BPM.

## Topics

- BPD and BPEL
- Key concepts and capabilities

*Topics*

## BPD and BPEL

Overview of IBM Business Process Manager

© Copyright IBM Corporation 2018

*BPD and BPEL*

## 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

*Two kinds of business processes: BPD and BPEL*

## 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



### BPMN

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.

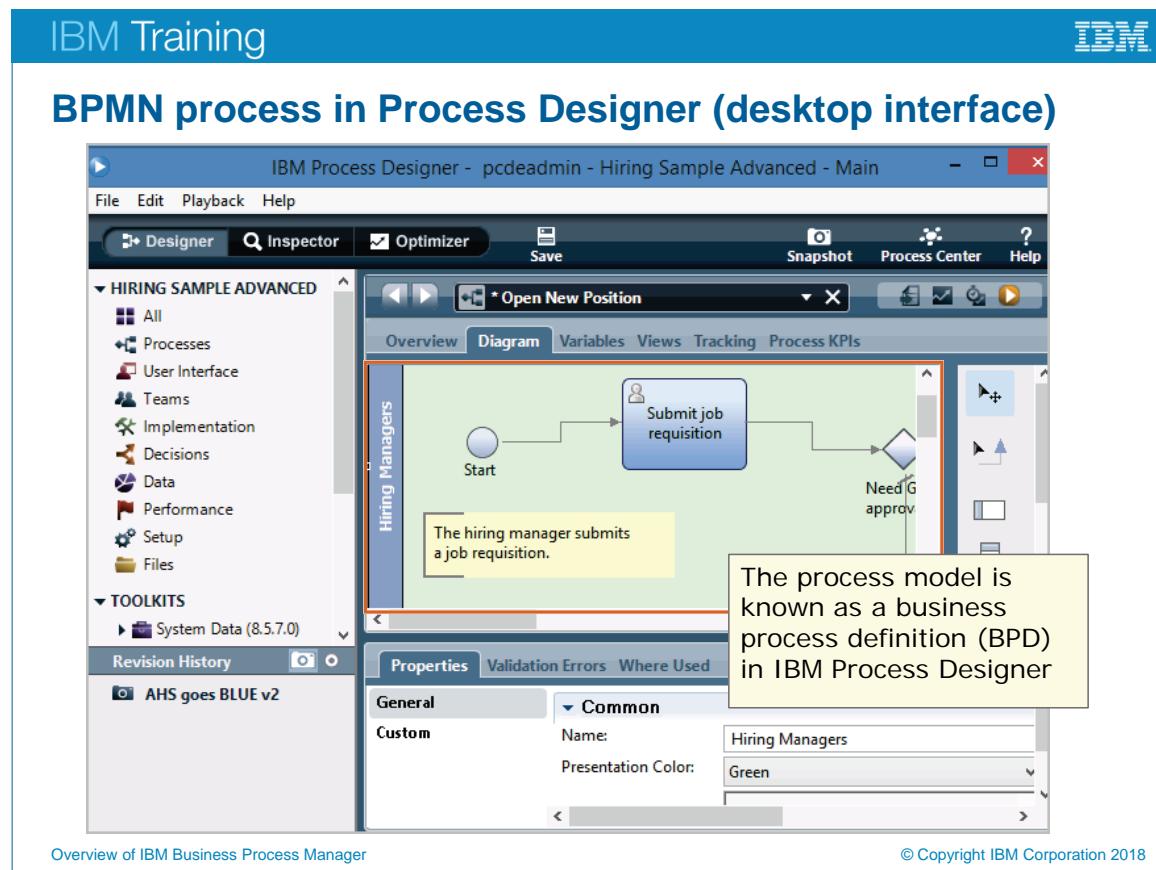
BPMN also allows a way to compact your process definition. Many of the symbols represent ideas, so symbols allow for a more concise and smaller model than drawing a diagram without BPMN.

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.



*BPMN process in Process Designer (desktop interface)*

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.

The screenshot shows the IBM BPMN process in Process Designer (web-based interface). The interface has a top navigation bar with 'DESIGNER' and 'INSPECTOR' tabs, and a search bar for 'Standard HR Open New Position'. Below the navigation is a toolbar with 'CREATE NEW' and other icons. On the left is a sidebar with categories: Processes, User Interface, Teams, Data, Performance, Files, and Toolkits. The main workspace displays a BPMN diagram for 'Hiring Sample' under 'Hiring Manager'. The diagram includes a 'Start' node, a task 'Submit position...', a decision diamond 'New position', and a connector 'Yes - GM approval'. A tooltip says: 'See the Documentation field for each activity and event for information about the artifact.' The bottom right of the workspace shows a 'Properties' panel with tabs for General, Implementation, and Data Mapping, currently displaying the 'Common' tab.

### BPMN process in Process Designer (web-based interface)

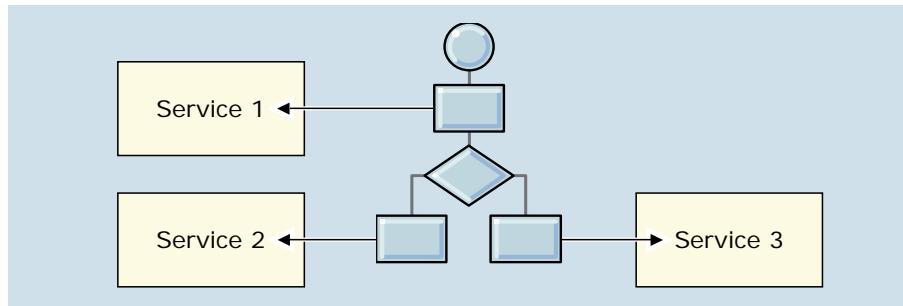
When you click the Open in Designer link in Process Center, the web editor opens by default. If you are working in the desktop Process Designer and you open an artifact that is created in the web Process Designer – for example, a process – the web Process Designer is launched. The artifact is opened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



Overview of IBM Business Process Manager

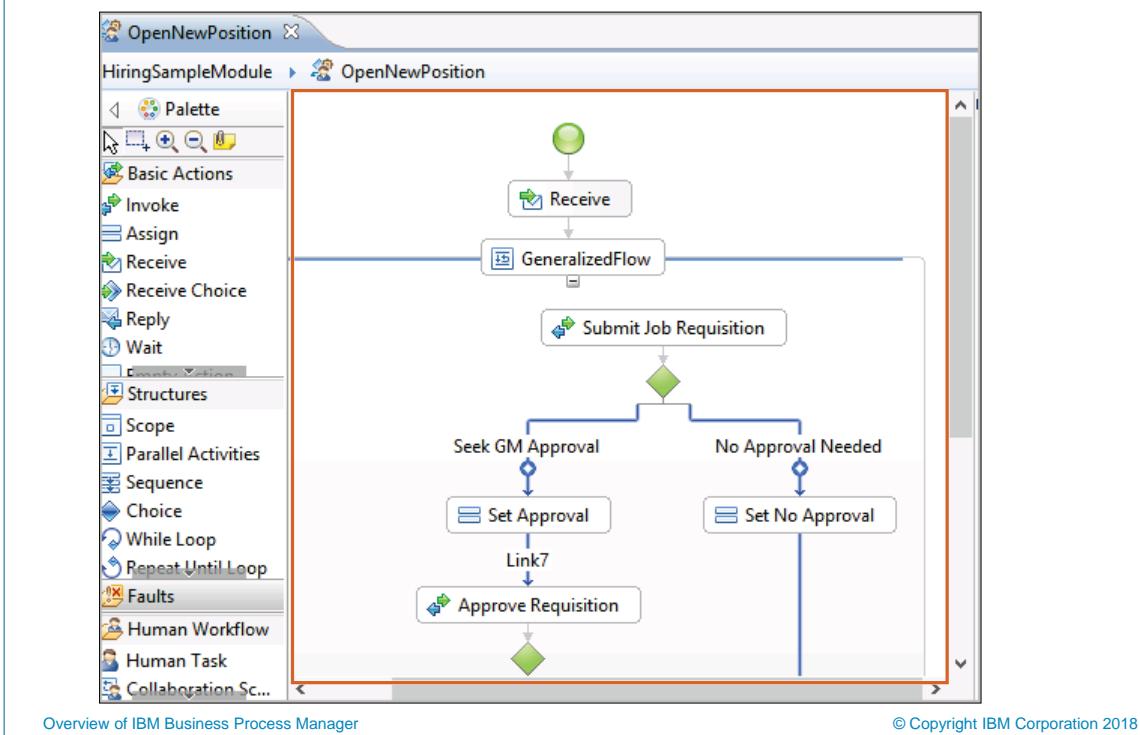
© Copyright IBM Corporation 2018

## BPEL

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



*BPEL process in IBM Integration Designer*

## 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

*When to use BPMN?*

## 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

### *When to use BPEL?*

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.

## Key concepts and capabilities

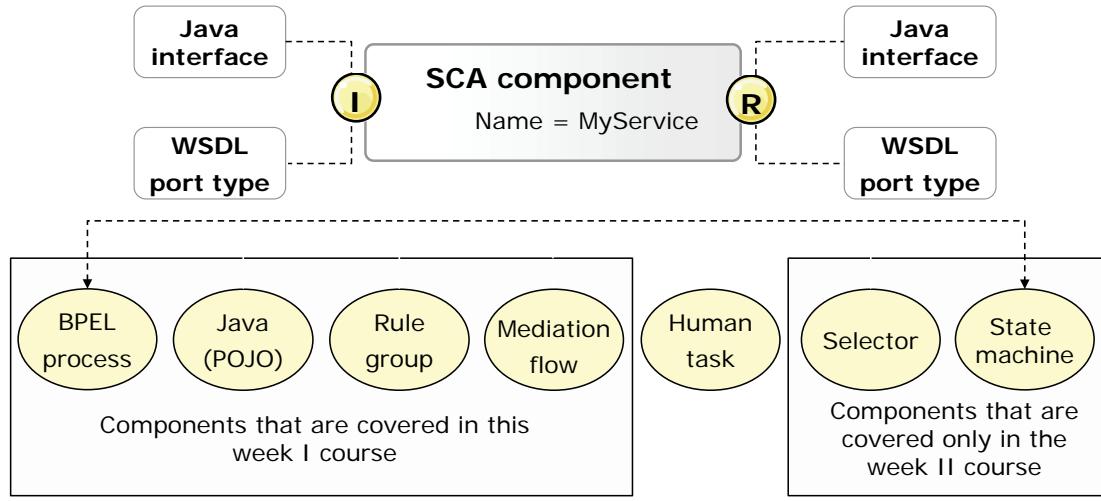
Overview of IBM Business Process Manager

© Copyright IBM Corporation 2018

*Key concepts and capabilities*

## 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



Overview of IBM Business Process Manager

© Copyright IBM Corporation 2018

### Service Component Architecture (SCA)

Components that are covered in this course 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, you are not encouraged to create new human task components for your IBM BPM solution. Coach user interface is the right solution.

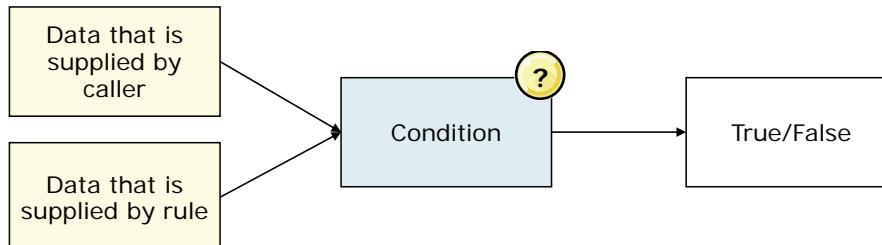
## Business value of Service Component Architecture (SCA)

- 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

*Business value of Service Component Architecture (SCA)*

## Business rules

- 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



Overview of IBM Business Process Manager

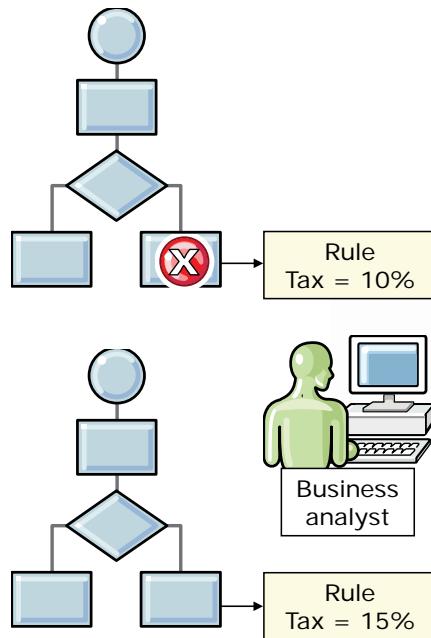
© Copyright IBM Corporation 2016

© Copyright IBM Corporation 2018

### *Business rules*

## 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



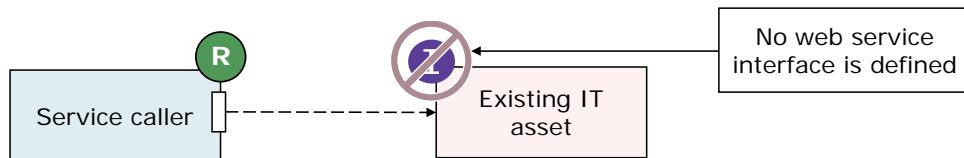
Overview of IBM Business Process Manager

© Copyright IBM Corporation 2018

*Business value of rules*

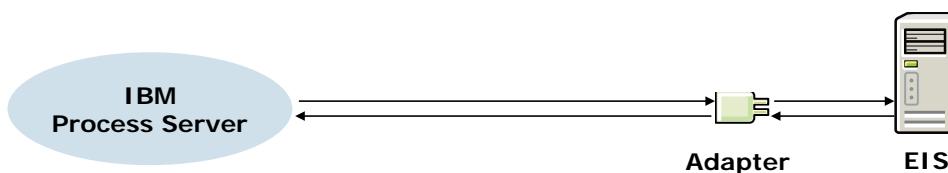
## Adapters

- 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)



## 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



## 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

### *Mediations*

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



Overview of IBM Business Process Manager

© Copyright IBM Corporation 2018

*Business value of mediations*

## Unit summary

- Describe the differences between BPMN and BPEL
- Describe the capabilities of IBM Business Process Manager

## Checkpoint questions

- True or False: A BPD is a business process that is created by using IBM Integration Designer.
- True or False: An organization that purchased IBM BPM can develop its processes only in Integration Designer that uses BPEL.

*Checkpoint questions*

## 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 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.

**Unit 3** Course business scenario: What you build

IBM Training



## **Course business scenario: What you build**

**IBM Business Process Manager V8.6**

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## Unit objectives

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

Course business scenario: What you build

© Copyright IBM Corporation 2018

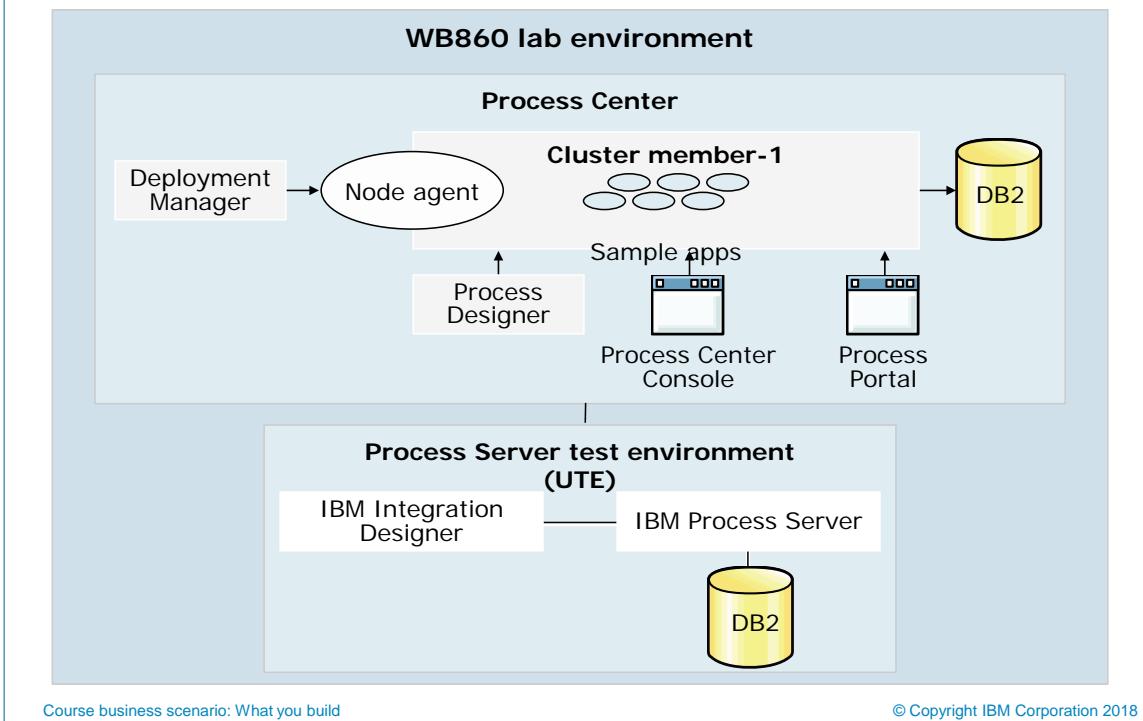
*Unit objectives*

## 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

*Some important points*

## WB872 lab environment



WB872 lab environment

## Exercise 1: Exploring IBM Integration Designer, part I

Exploring IBM Integration Designer, part I

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 1: Exploring IBM Integration Designer, part I*

## What are some key tasks that you do in Exercise 1?

- Explore the IBM Integration Designer tool and its capabilities

Course business scenario: What you build

© Copyright IBM Corporation 2018

### *What are some key tasks that you do in Exercise 1?*

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.

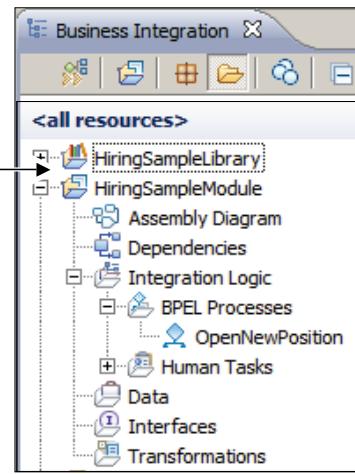
## Components that are required for Exercise 1

Prebuilt components that are imported in the lab:

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

New components that you create in the lab:

**None**



### Components that are required for Exercise 1

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.

IBM Training IBM

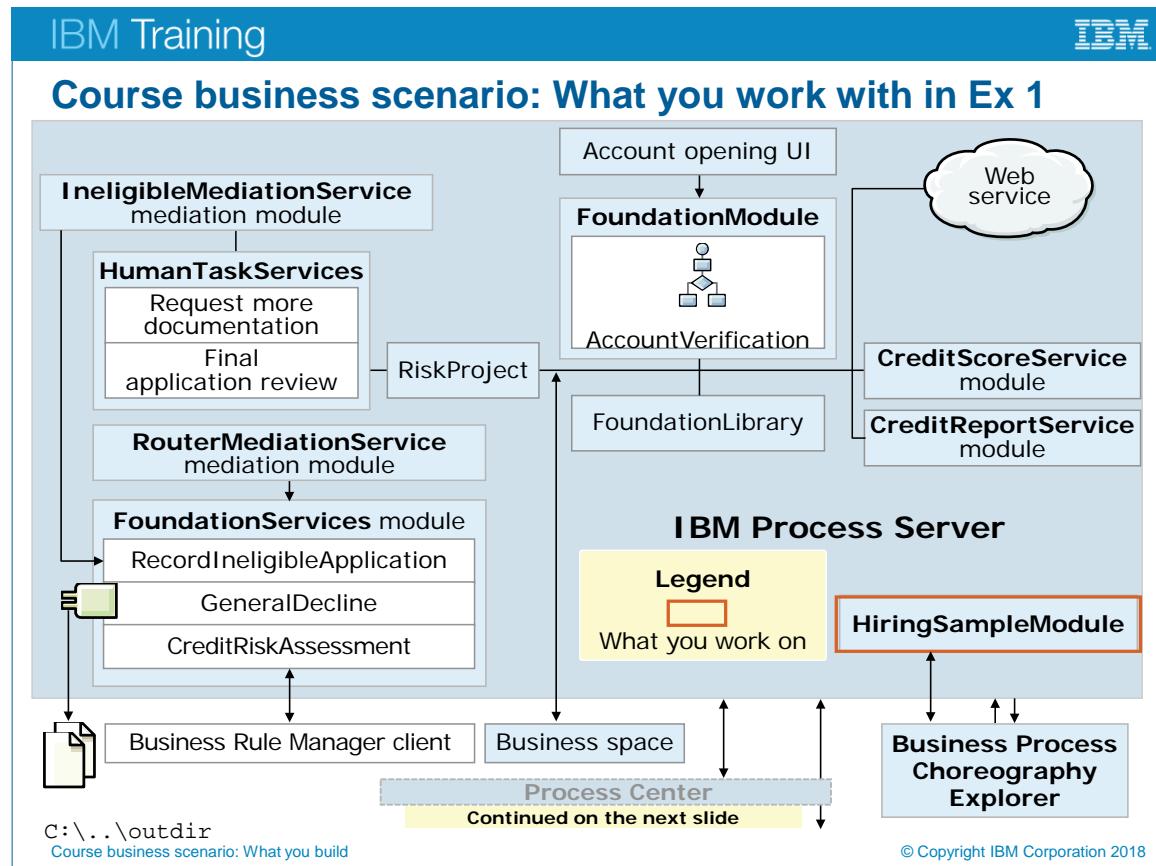
## Explore IBM Integration Designer in Exercise 1

```
graph TD; Start[Submit Job Requisition] --> Decision{Decision Diamond}; Decision -- Seek GM Approval --> GMApprove[GM Approves]; Decision -- No Approval Needed --> ManagerApprove[Hiring Manager Approves]; GMApprove --> Converge(( )); ManagerApprove --> Converge;
```

Course business scenario: What you build © Copyright IBM Corporation 2018

### Explore IBM Integration Designer in Exercise 1

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 1*

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

## Exercise 2: Exploring IBM Integration Designer, part II

Exploring IBM Integration Designer, part II

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 2: Exploring IBM Integration Designer, part II*

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

- 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

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

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.

IBM Training IBM

## Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

1. **HiringSampleLibrary**
2. **HiringSampleModule**  
Previously imported in Exercise 2

New components that you create in the lab:

**None**

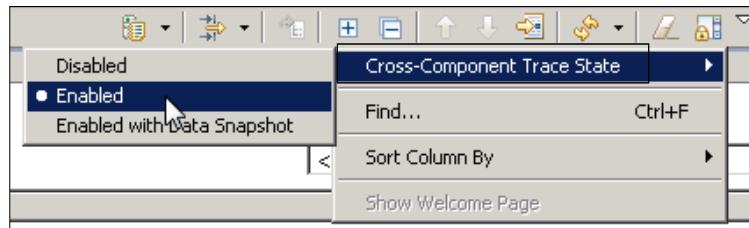
```
graph TD; Root[<call resources>] --> HSL[HiringSampleLibrary]; Root --> HSM[HiringSampleModule]; HSM --> AD[Assembly Diagram]; HSM --> D[Dependencies]; HSM --> IL[Integration Logic]; IL --> BP[BPEL Processes]; IL --> HT[Human Tasks]; IL --> D[Data]; IL --> I[Interfaces]; IL --> T[Transformations]; BP --> OP[OpenNewPosition]
```

Course business scenario: What you build © Copyright IBM Corporation 2018

### Components that are required for Exercise 2

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 2



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-be64-5434E
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

Course business scenario: What you build

© Copyright IBM Corporation 2018

### Use cross-component tracing in Exercise 2

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.

IBM Training IBM

## Test by using the Business Process Choreographer Explorer

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 Business Category**

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

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	Tony
<input type="checkbox"/> 5	GMAApproval	Ready	To-do Task	Tony

Items found: 2 Items selected: 0

Page 1 of 1

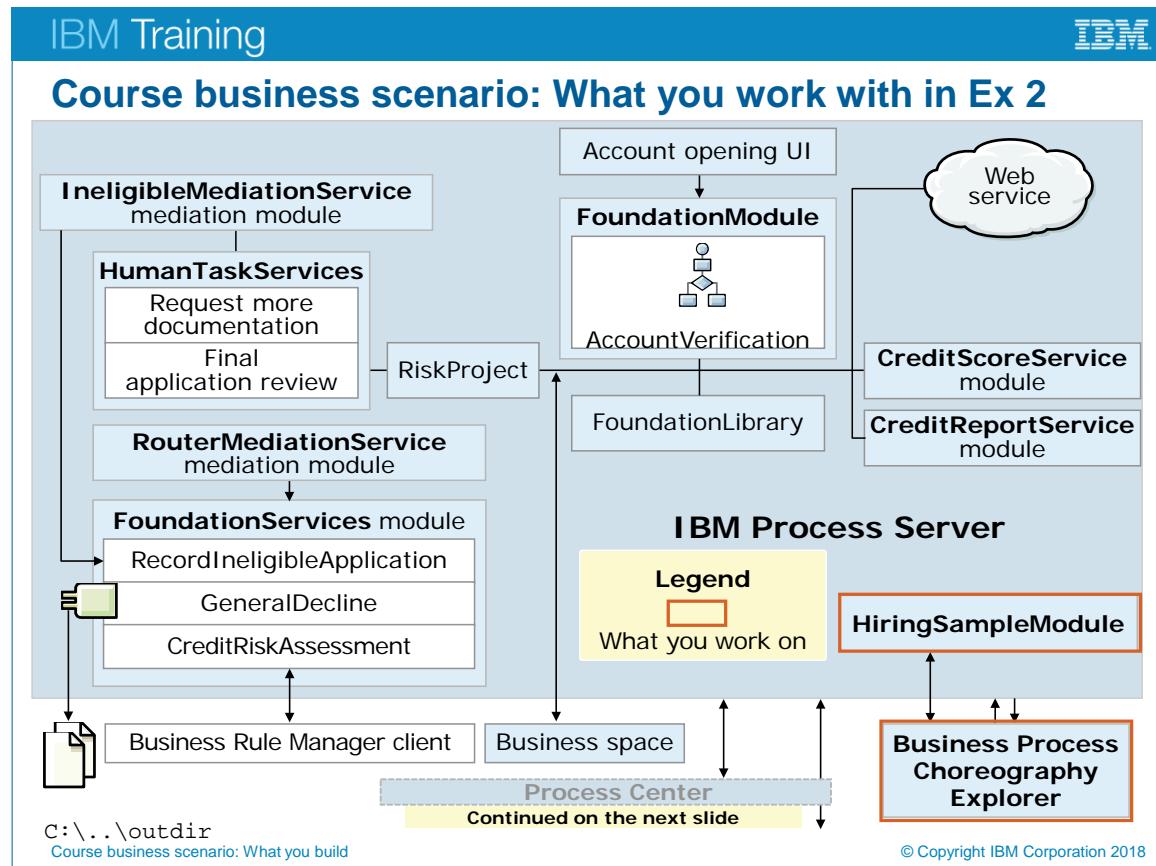
Course business scenario: What you build © Copyright IBM Corporation 2018

### Test by using the Business Process Choreographer Explorer

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 2

## Exercise 3: Working with web services

Working with web services

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 3: Working with web services*

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

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

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

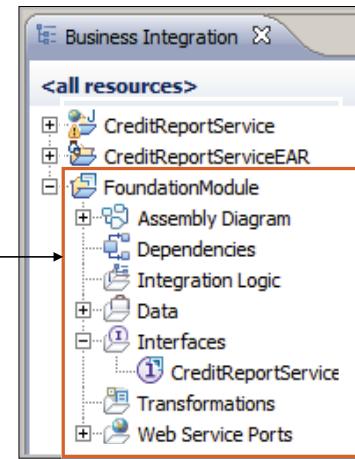
## Components that are required for Exercise 3 (Section 1)

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

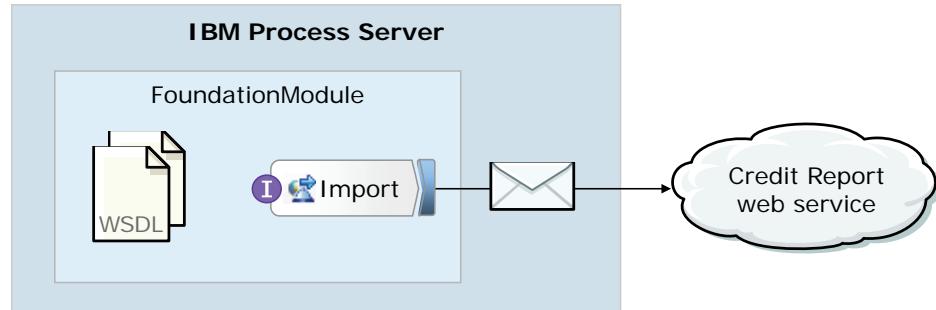


Course business scenario: What you build

© Copyright IBM Corporation 2018

*Components that are required for Exercise 3 (Section 1)*

## Invoke an external web service in Exercise 3 (Section 1)



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Invoke an external web service in Exercise 3 (Section 1)*

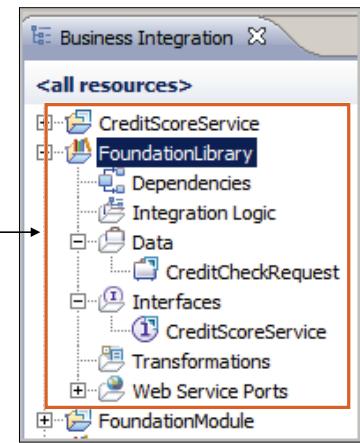
## Components that are required for Exercise 3 (Section 2)

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:

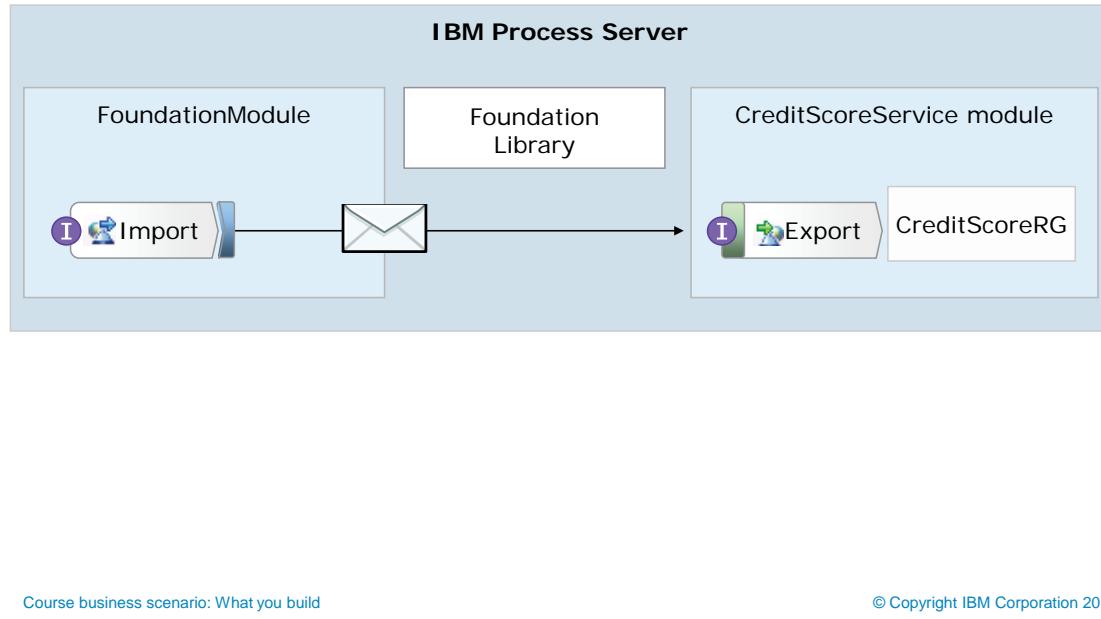
1. **FoundationLibrary**
  - Container for shared business objects and interfaces:  
**CreditScoreService.wsdl** interface and  
**CreditCheckRequest** business object
2. **CreditScoreService**
  - Export SCA component that exposes the **CreditScoreService** web service
3. **CreditScoreService**
  - Import SCA component that calls the service through the SCA export



© Copyright IBM Corporation 2018

*Components that are required for Exercise 3 (Section 2)*

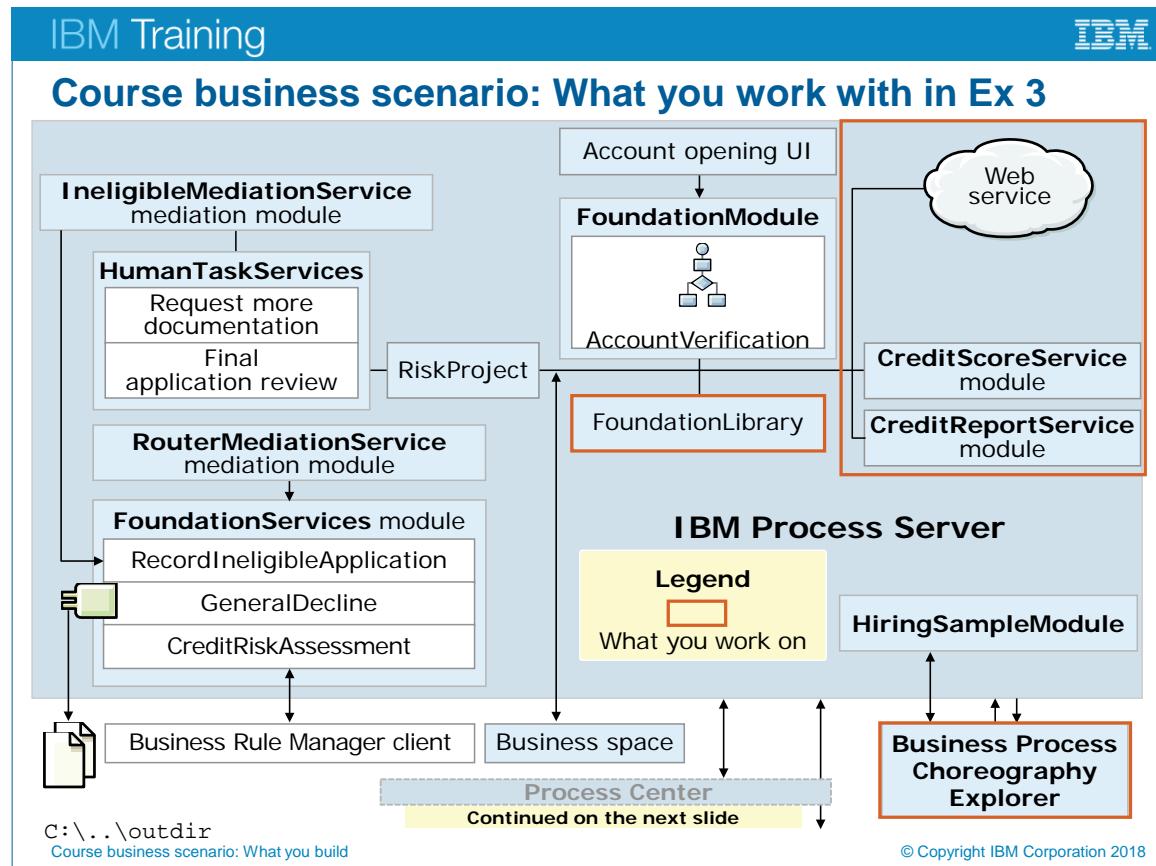
## Expose an application as a web service that is running in IBM Process Server in Exercise 3 (Section 2)



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Expose an application as a web service that is running in IBM Process Server in Exercise 3 (Section 2)*



Course business scenario: What you work with in Ex 3

## Exercise 4: Creating business objects and shared interfaces

Creating business objects and shared interfaces

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 4: Creating business objects and shared interfaces*

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

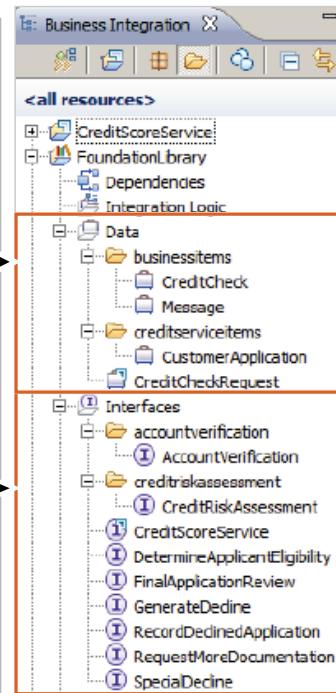
- Implement business objects and define their elements
- Implement interfaces that use business objects as inputs and outputs

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

## Components that are required for Exercise 4 (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



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Components that are required for Exercise 4 (1 of 2)*

## Components that are required for Exercise 4 (2 of 2)

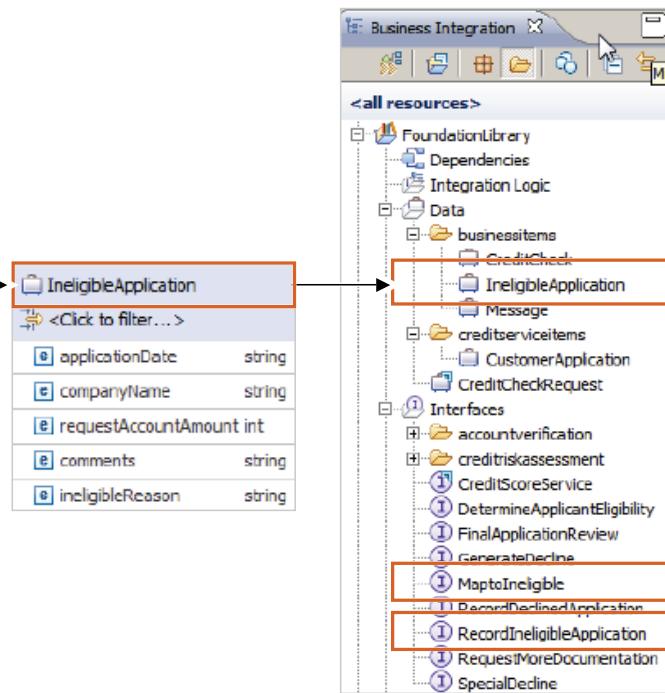
New components that you create in the lab:

**1. Business object:**

- IneligibleApplication

**2. Interfaces:**

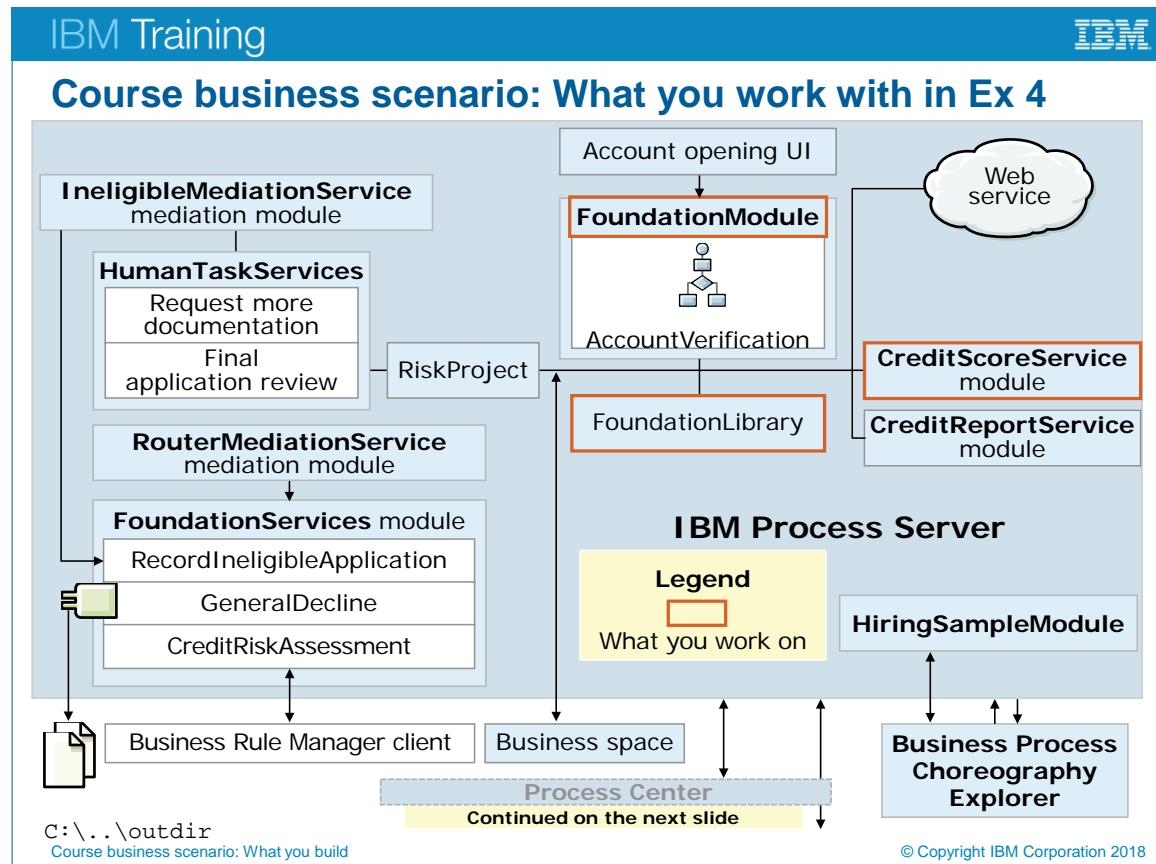
- MptoIneligible
- RecordIneligibleApplication



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Components that are required for Exercise 4 (2 of 2)*



Course business scenario: What you work with in Ex 4

## Exercise 5: Creating a business process, part I

Creating a business process, part I

Course business scenario: What you build

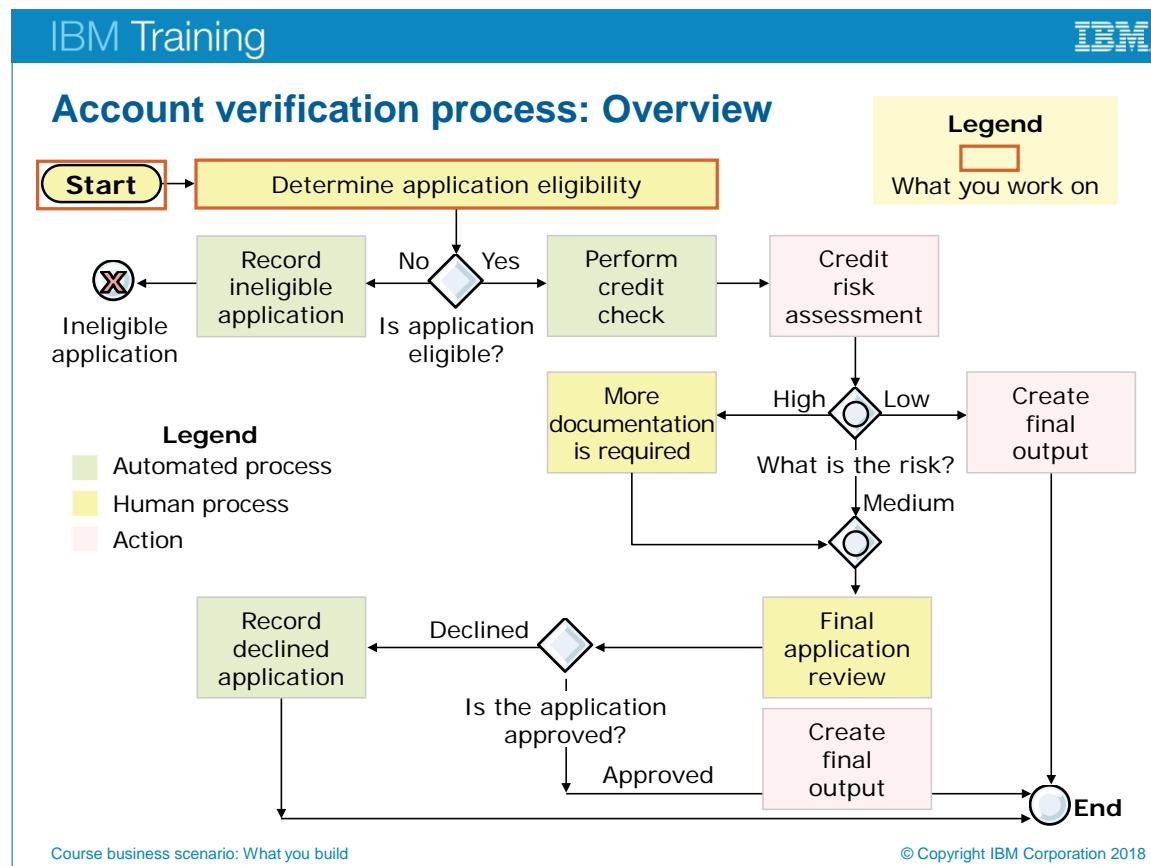
© Copyright IBM Corporation 2018

*Exercise 5: Creating a business process, part I*

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

- 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

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



### Account verification process: Overview

The following scenario illustrates this concept.

### Account Verification Process scenario

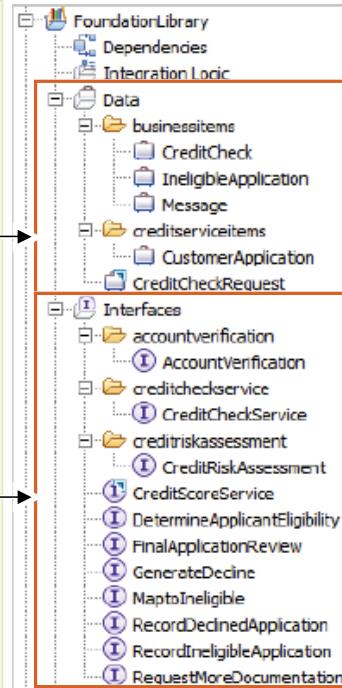
- When the customer submits an application, the application must be tested for eligibility.
- If the application is ineligible, record the ineligible application in the database and end the process.
- If the application is eligible, the system calls an external service to do a credit check.
- A credit 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.

## Components that are required for Exercise 5 (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



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Components that are required for Exercise 5*

IBM Training IBM

## Components that are required for Exercise 5 (2 of 7)

New components that you create in the lab:

**1. AccountVerification BPEL process:**

- Uses the **AccountVerification** interface

Operation - InputCriterion	
Description	
	Interface <b>AccountVerification</b>
	Operation <b>InputCriterion</b>
Input	<b>Input</b>
Output	<b>Output</b>
	<b>Customer Application</b>
	<b>Message</b>

Course business scenario: What you build © Copyright IBM Corporation 2018

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.

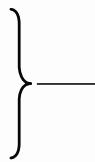
The interface partner is a direct link to the interface where the partner is configured. An interface partner is the process interface and exposes operations that external users or services call.

## Components that are required for Exercise 5 (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

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:

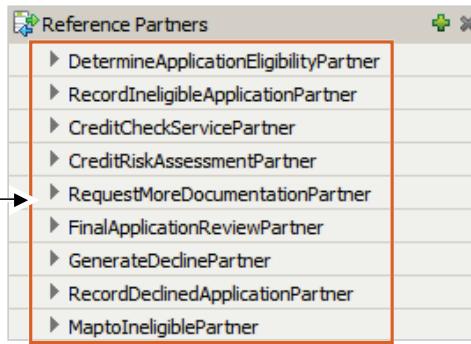
- The global variable that is named CreditCheckVariable stores a CreditCheckRequest business object.
- The global variable that is named IneligibleApplicationVariable stores an IneligibleApplication business object.
- The global variable that is named CustomerApplicationVariable2 stores a CustomerApplication 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 5 (4 of 7)

New components that you create in the lab:

### 3. Reference partners:

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



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.

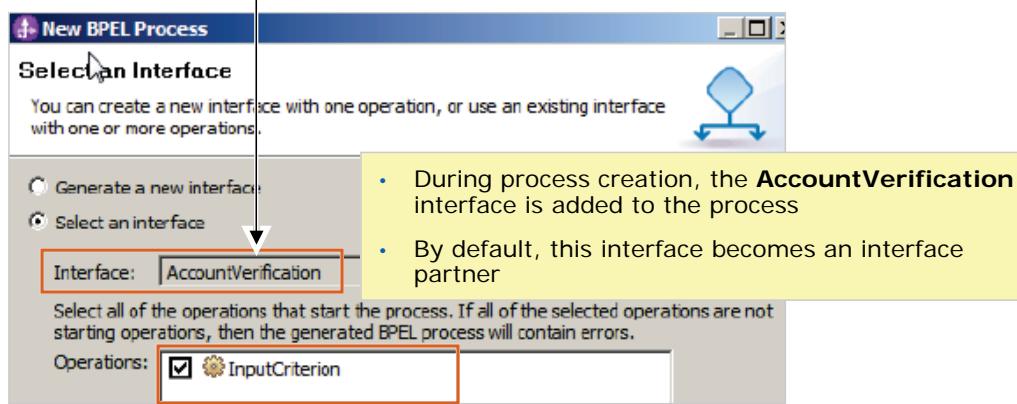
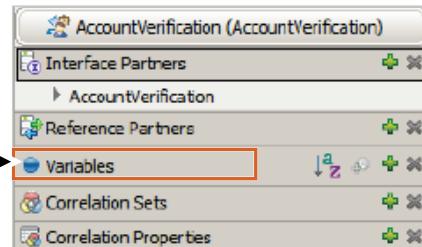
7. Add the GenerateDeclinePartner reference partner with the GenerateDecline interface.
8. Add the RecordDeclinedApplicationPartner reference partner with the RecordDeclinedApplication interface.
9. Add the MaptoIneligiblePartner reference partner with the MaptoIneligible interface.

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

New components that you create in the lab:

### 4. Interface partner

- **AccountVerification** interface partner



Course business scenario: What you build

© Copyright IBM Corporation 2018

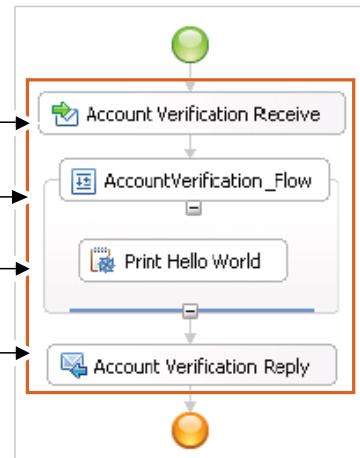
An interface provides the input and output of a component. It is created independent of the internal implementation of the component. The AccountVerification interface dictates how to use the AccountVerification process. It specifies the operations that can be called and the data that is passed, such 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 5 (6 of 7)

New components that you create in the lab:

5. Receive activity
  - **Account Verification Receive**
6. Generalized flow activity
  - **AccountVerification\_Flow**
7. Snippet action
  - **Print Hello World**
8. Reply activity
  - **Account Verification Reply**

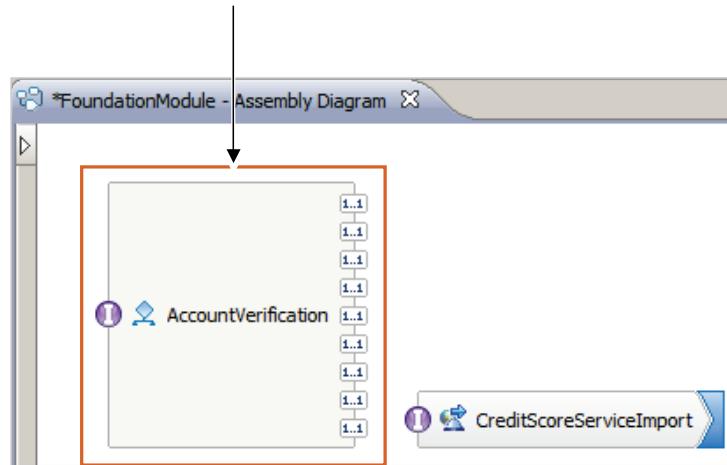


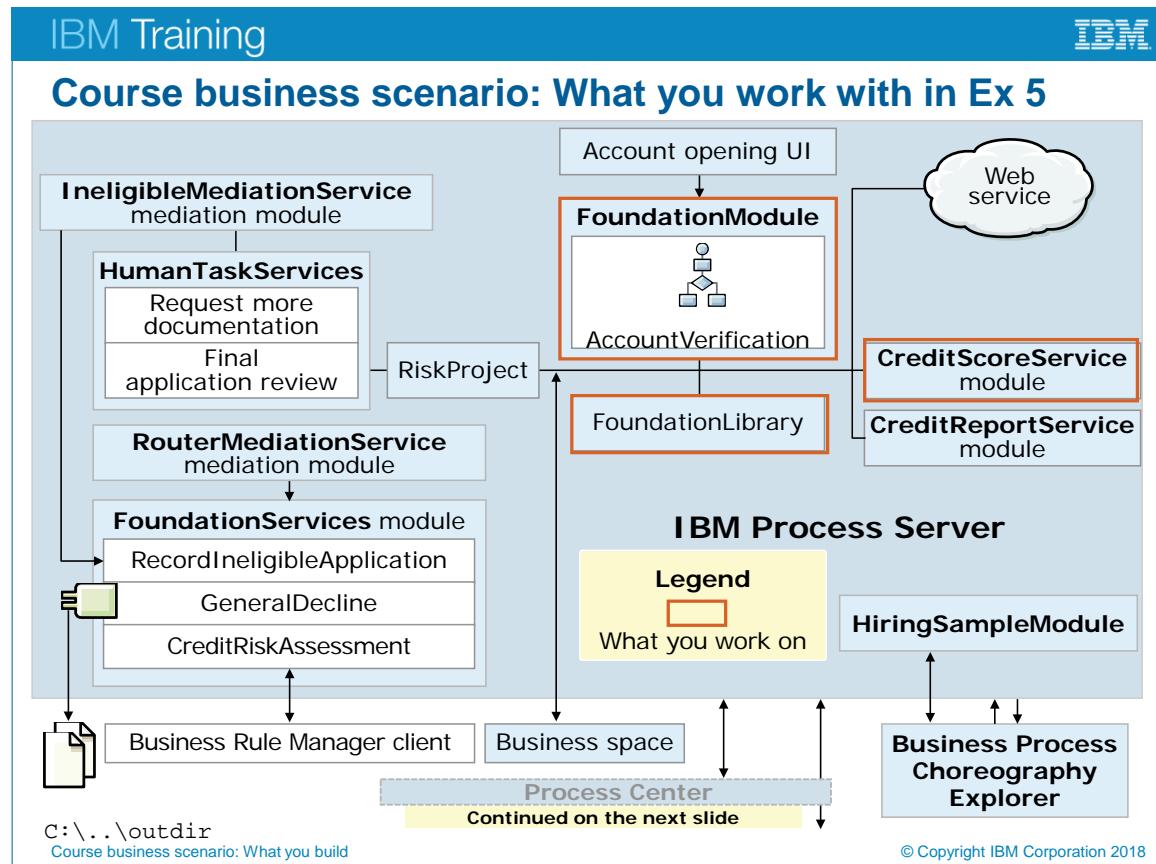
In this exercise, you add a generalized flow that is a structured activity that acts as the container for the simple and complex BPEL activities.

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

New components that you create in the lab:

9. SCA component
  - **AccountVerification** component





Course business scenario: What you work with in Ex 5

## Exercise 6: Creating a business process, part II

Creating a business process, part II

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 6: Creating a business process, part II*

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

- Implement basic and structured BPEL activities in a business process

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

## Components that are required for Exercise 6 (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:**

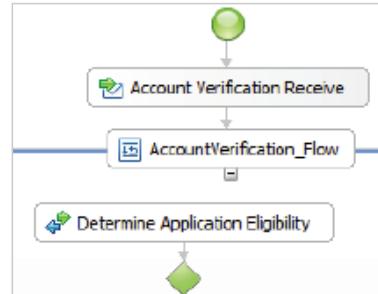
The next slide lists all of the subcomponents of the **AccountVerification** process that are built in this lab

*Components that are required for Exercise 6*

## Components that are required for Exercise 6 (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



Course business scenario: What you build

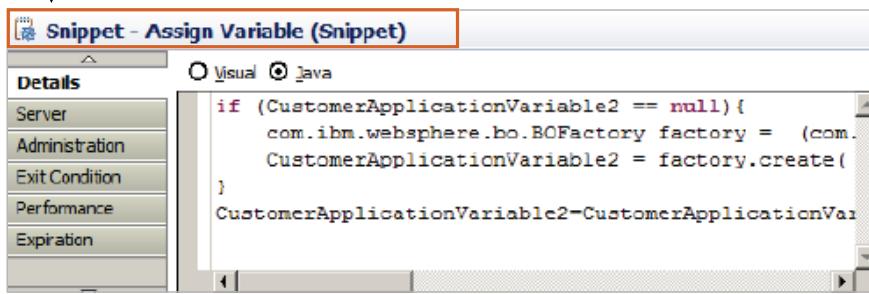
© Copyright IBM Corporation 2018

- 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.
- The Map to Ineligible invoke activity invokes the InputCriterion operation of the Maptolineligible interface of MaptolineligiblePartner.
- 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.
- 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.
- The Credit Check Service activity invokes the calculateCreditScore operation of the CreditScoreService interface of CreditCheckServicePartner.
- 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 6

- 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

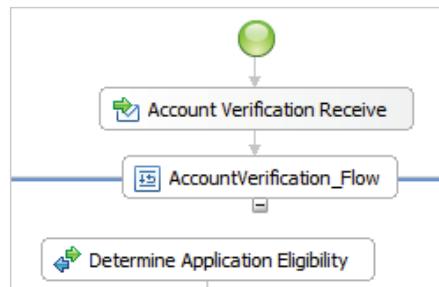


Course business scenario: What you build

© Copyright IBM Corporation 2018

*Code snippets that are created in the AccountVerification process in Exercise 6*

## Complete Account Verification process (1 of 4)

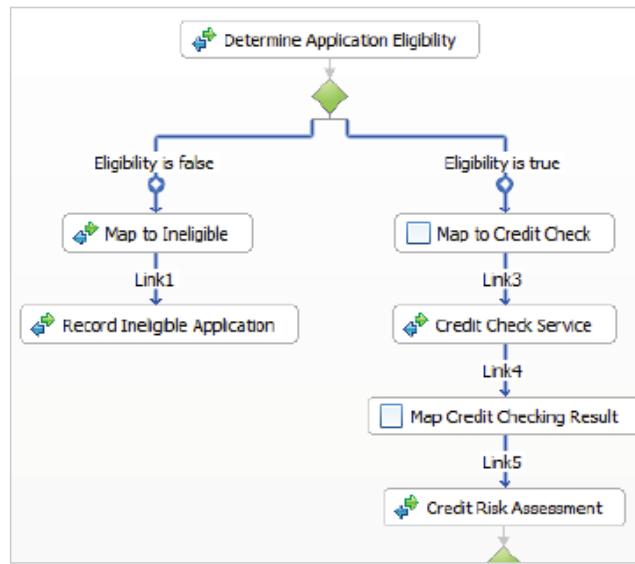


Course business scenario: What you build

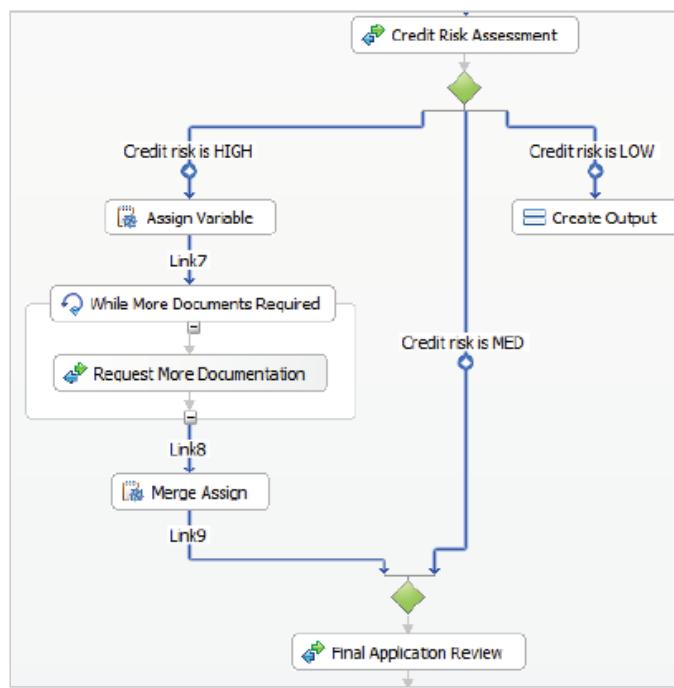
© Copyright IBM Corporation 2018

*Complete Account Verification process*

## Complete Account Verification process (2 of 4)



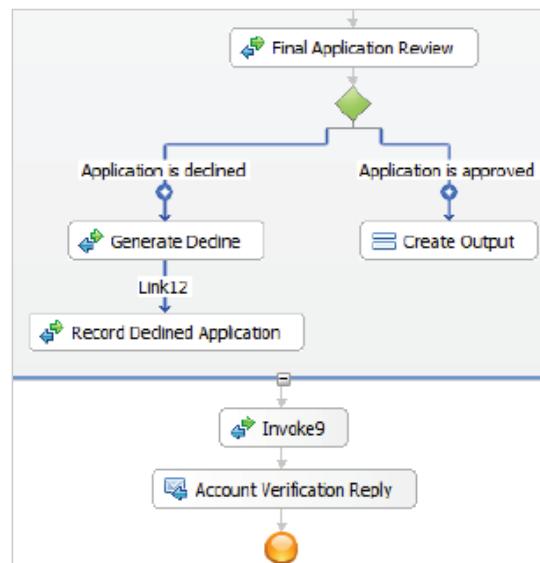
## Complete Account Verification process (3 of 4)

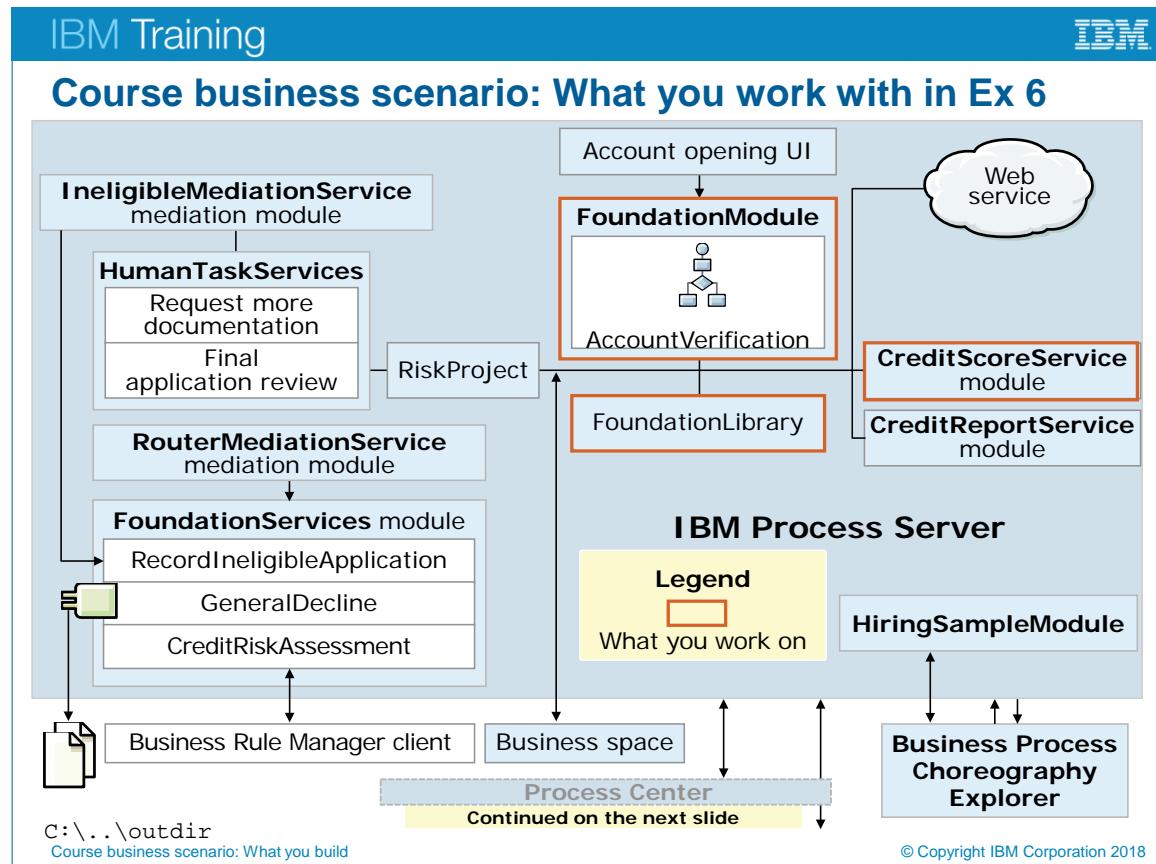


Course business scenario: What you build

© Copyright IBM Corporation 2018

## Complete Account Verification process (4 of 4)





Course business scenario: What you work with in Ex 6

## Exercise 7: Creating a business process, part III

Creating a business process, part III

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 7: Creating a business process, part III*

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

- 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

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

## Components that are required for Exercise 7

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**

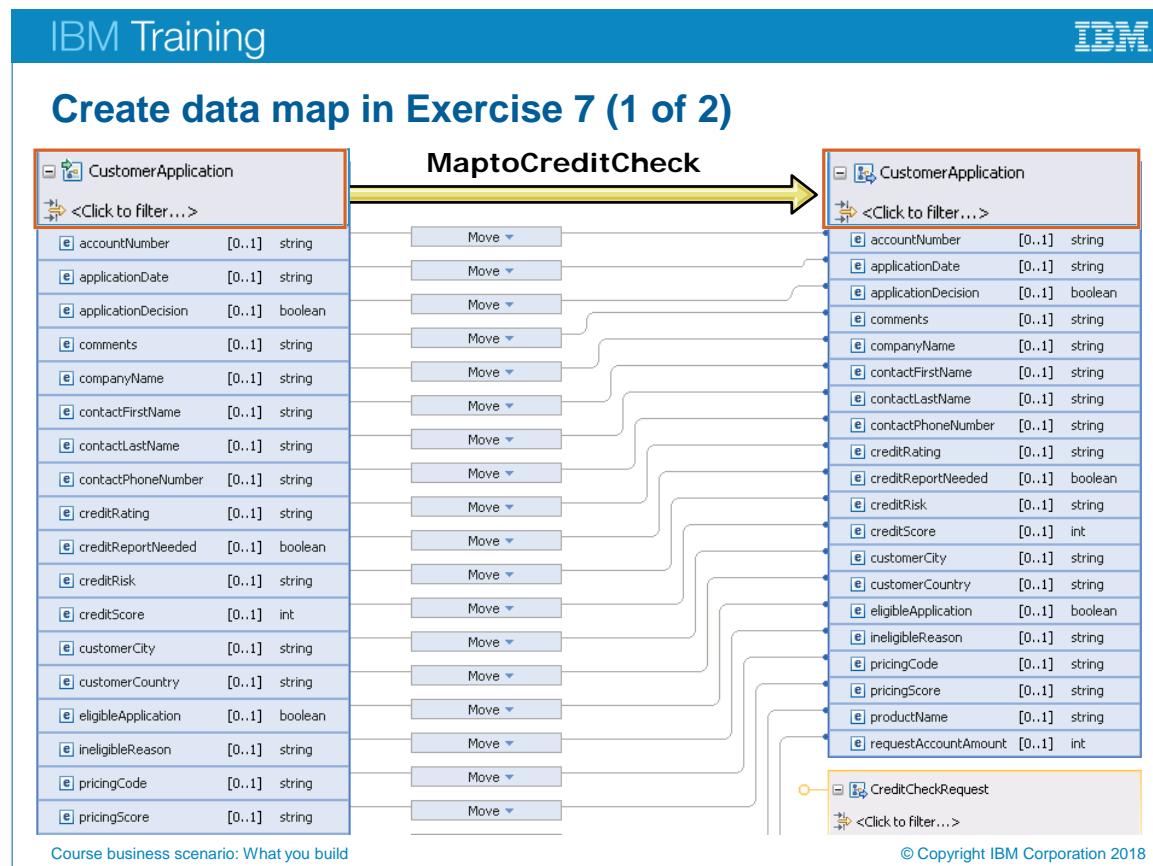
— BPEL process that you completed in Exercise 7  
— You complete implementation of the process in this lab

**Note:**

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

### *Components that are required for Exercise 7*

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 7

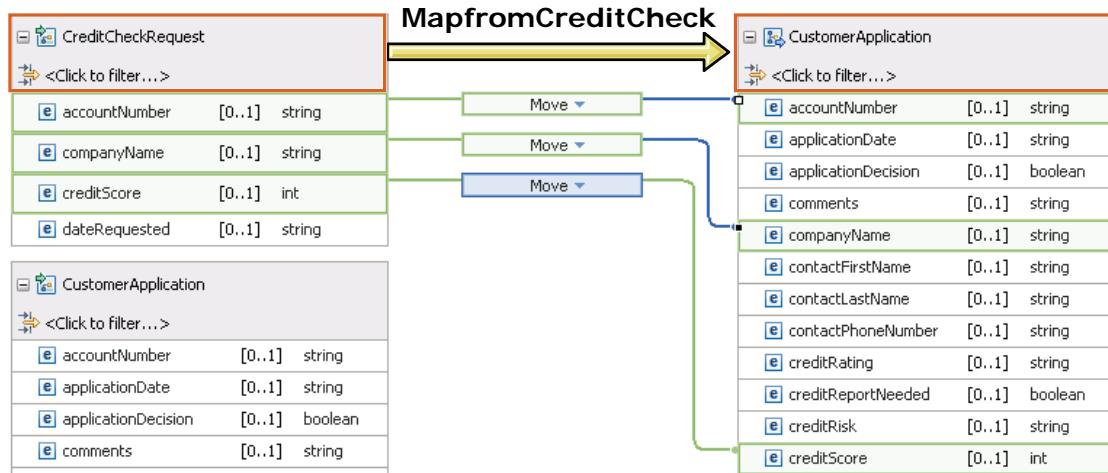
The AccountVerification business process that you implement receives and operates on a CustomerApplication business object input. During process execution, your business process calls an external web service, CreditScoreService, which provides the credit score that is needed to determine the customer's 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 7 (2 of 2)



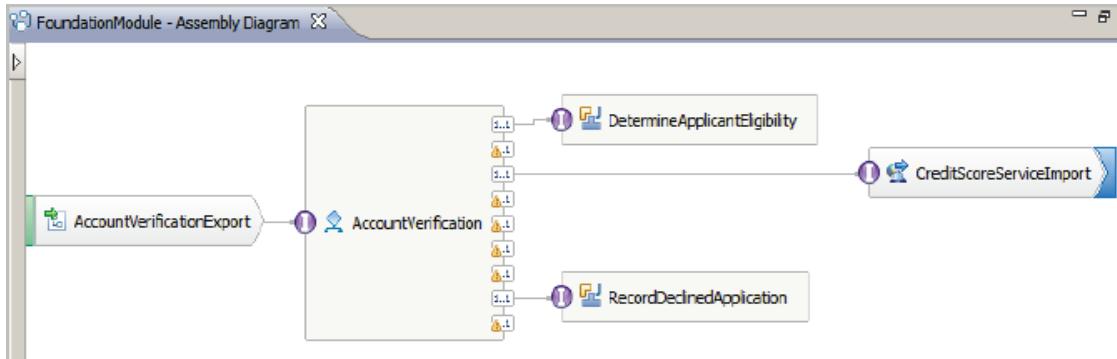
Course business scenario: What you build

© Copyright IBM Corporation 2018

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

IBM Training 

## Assemble SCA application that contains the business process in Exercise 7

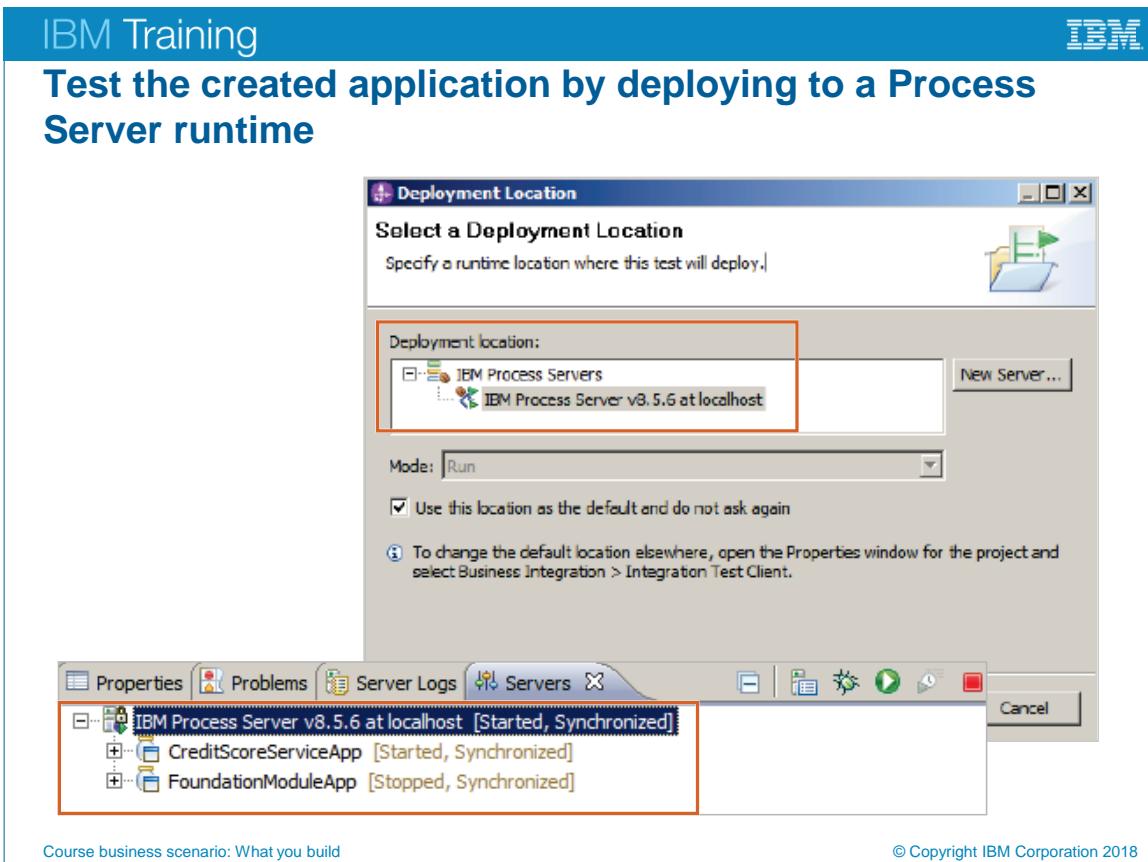


The diagram illustrates the assembly of SCA components within a FoundationModule. It features a central composite component labeled 'AccountVerification' which contains several internal components: 'DetermineApplicantEligibility', 'RecordDeclinedApplication', and five smaller orange components. External interfaces include 'AccountVerificationExport' on the left and 'CreditScoreServiceImport' on the right, both connected to the central component via purple circular ports.

Course business scenario: What you build © Copyright IBM Corporation 2018

### Assemble SCA application that contains the business process in Exercise 7

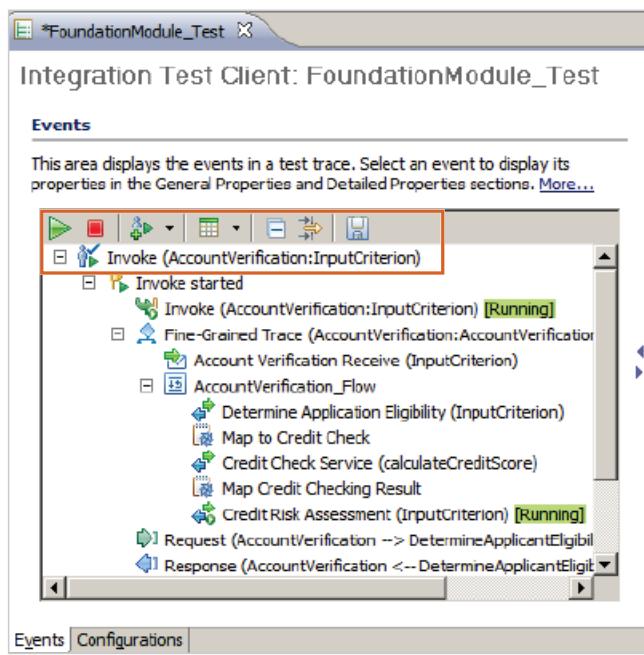
In the assembly diagram, you wire the SCA components into an integrated application that is deployed to the runtime environment.



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

## Use the test client in Exercise 7

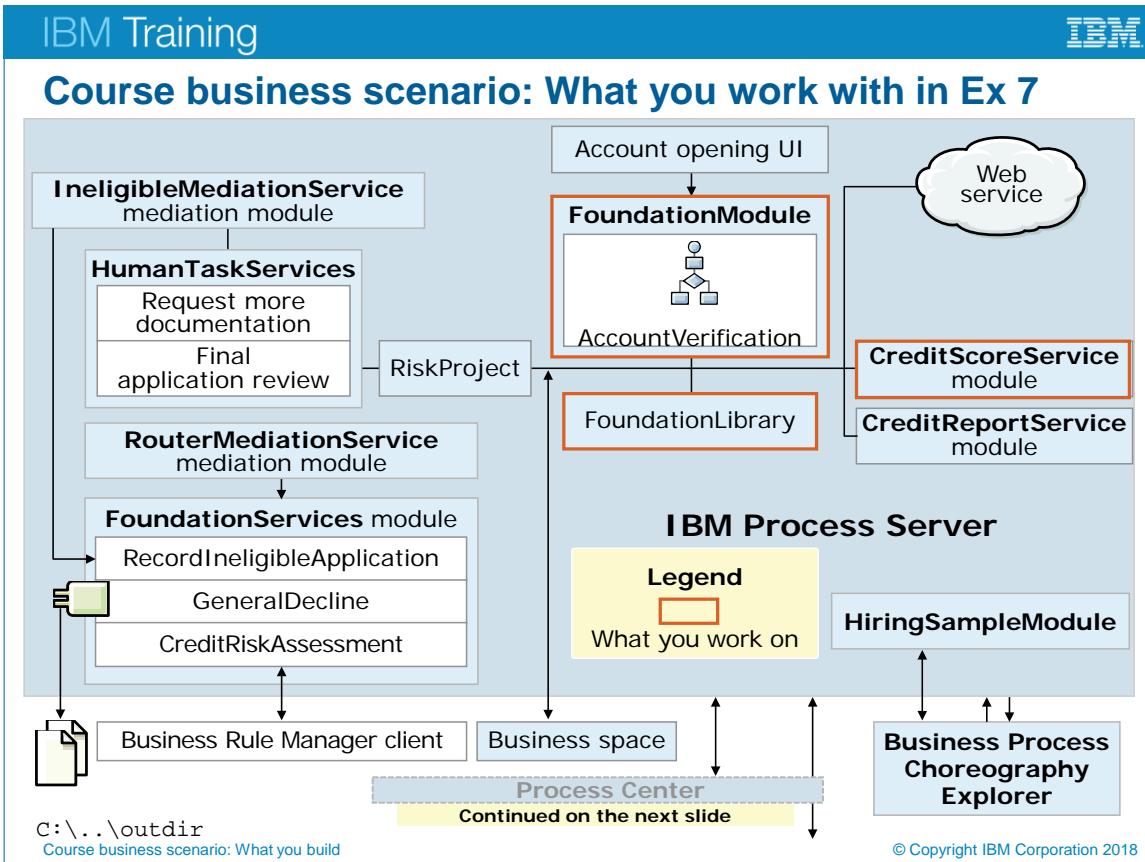
- Use the Integration Test client to run and test the **AccountVerification** process



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Use the test client in Exercise 7*



Course business scenario: What you work with in Ex 7

## Exercise 8: Creating business rules

Creating business rules

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 8: Creating business rules*

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

- 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

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

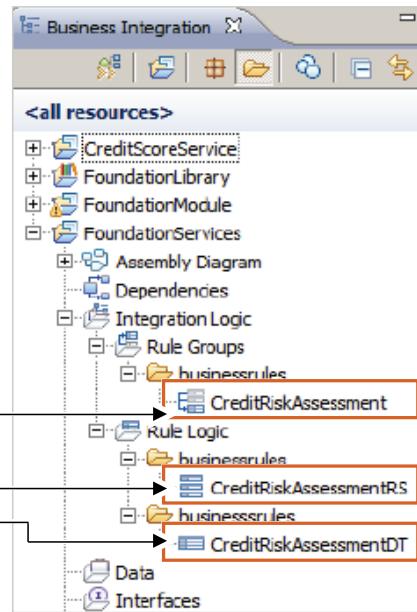
## Components that are required for Exercise 8

Prebuilt components that are imported in the lab:

1. **FoundationModule**
2. **CreditScoreService**
3. **FoundationLibrary**
4. **AccountVerification**
  - BPEL process that you 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



Course business scenario: What you build

© Copyright IBM Corporation 2018

### *Components that are required for Exercise 8*

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 8

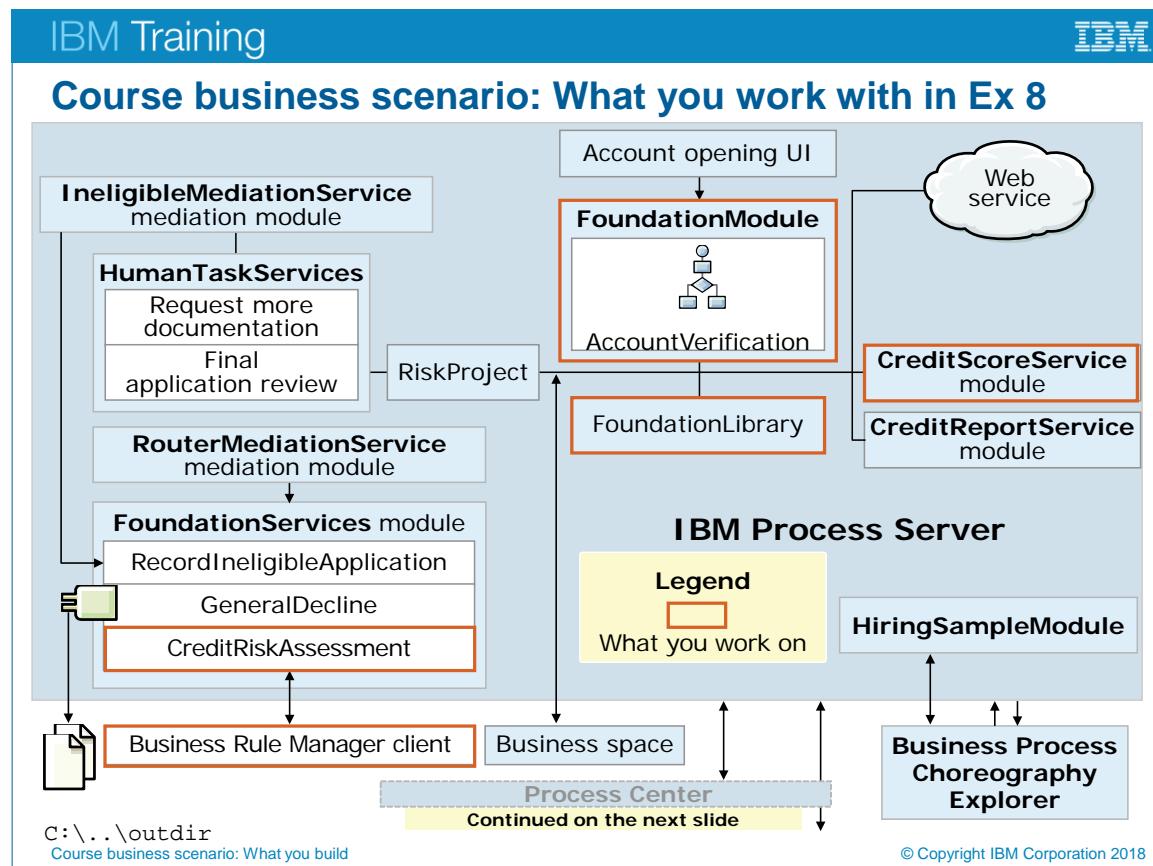
The screenshot shows the IBM Business Rules interface. On the left, there's a navigation tree with nodes like 'Rule Set', 'Interface', 'Variables', and 'Rules'. The 'Rules' node is expanded, showing a table with three rows: 'MapOutput', 'RiskHIGH', and 'RiskMED'. The 'RiskLOW' row is partially visible. The 'RiskHIGH' row is highlighted with a red border. The 'MapOutput' row contains the action 'Output =copyBO( Input)'. The 'RiskHIGH' row contains the template 'CreditRiskTemplate' and the presentation 'If the customer credit score is greater than [0] and less than [4] then the credit risk is [HIGH]'. The 'RiskMED' row contains the template 'CreditRiskTemplate' and the presentation 'If the customer credit score is greater than [3] and less than [8] then the credit risk is [MED]'. The 'RiskLOW' row contains the template 'CreditRiskTemplate' and the presentation 'If the customer credit score is greater than [7] and less than [12] then the credit risk is [LOW]'. A 'Scheduled Rule Logic' panel is open at the top, showing a single entry for 'CreditRiskAssessmentRS'.

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

Course business scenario: What you build

© Copyright IBM Corporation 2018

### Create rule group in Exercise 8



Course business scenario: What you work with in Ex 8

## Exercise 9: Implementing WebSphere (JCA) adapters

Implementing WebSphere (JCA) adapters

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 9: Implementing WebSphere (JCA) adapters*

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

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

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

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 9 (1 of 2)

Prebuilt components that are imported in the lab:

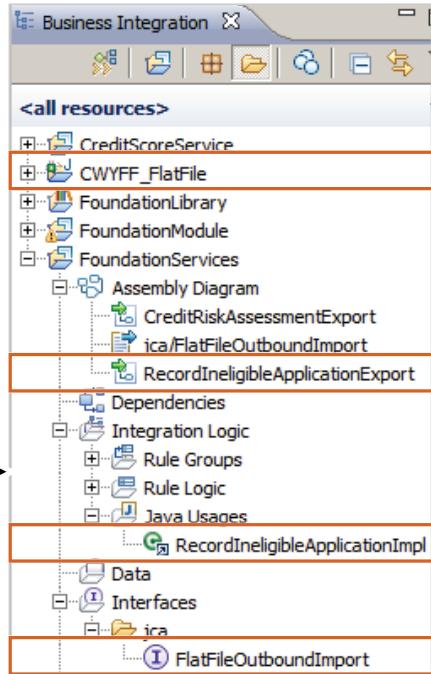
- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. FoundationServices** that you built in Exercise 9

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

Course business scenario: What you build

© Copyright IBM Corporation 2018



### Components that are required for Exercise 9

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.

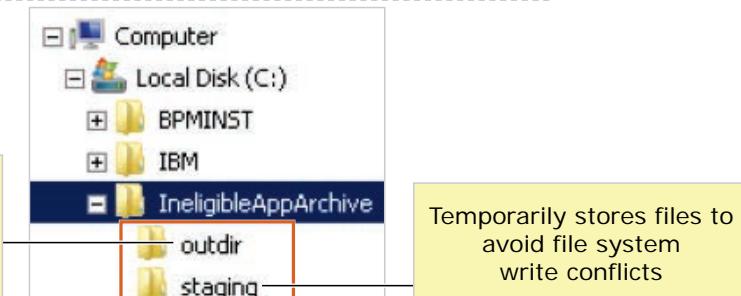
In this exercise, you implement a service to record ineligible applications as XML files on the file system. This service uses the WebSphere Adapter for Flat Files. The AccountVerification process invokes the service when the eligibleApplication attribute is set to false.

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.

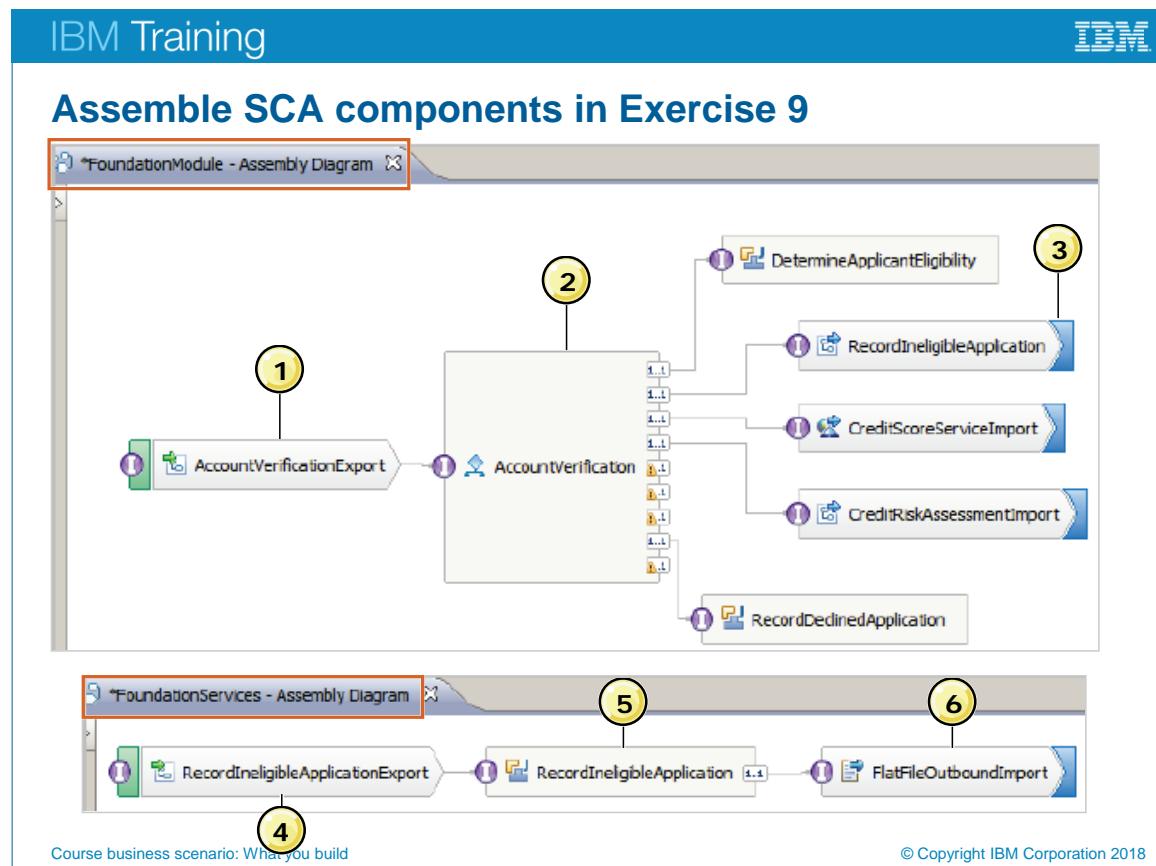
## Components that are required for Exercise 9 (2 of 2)

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



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`.



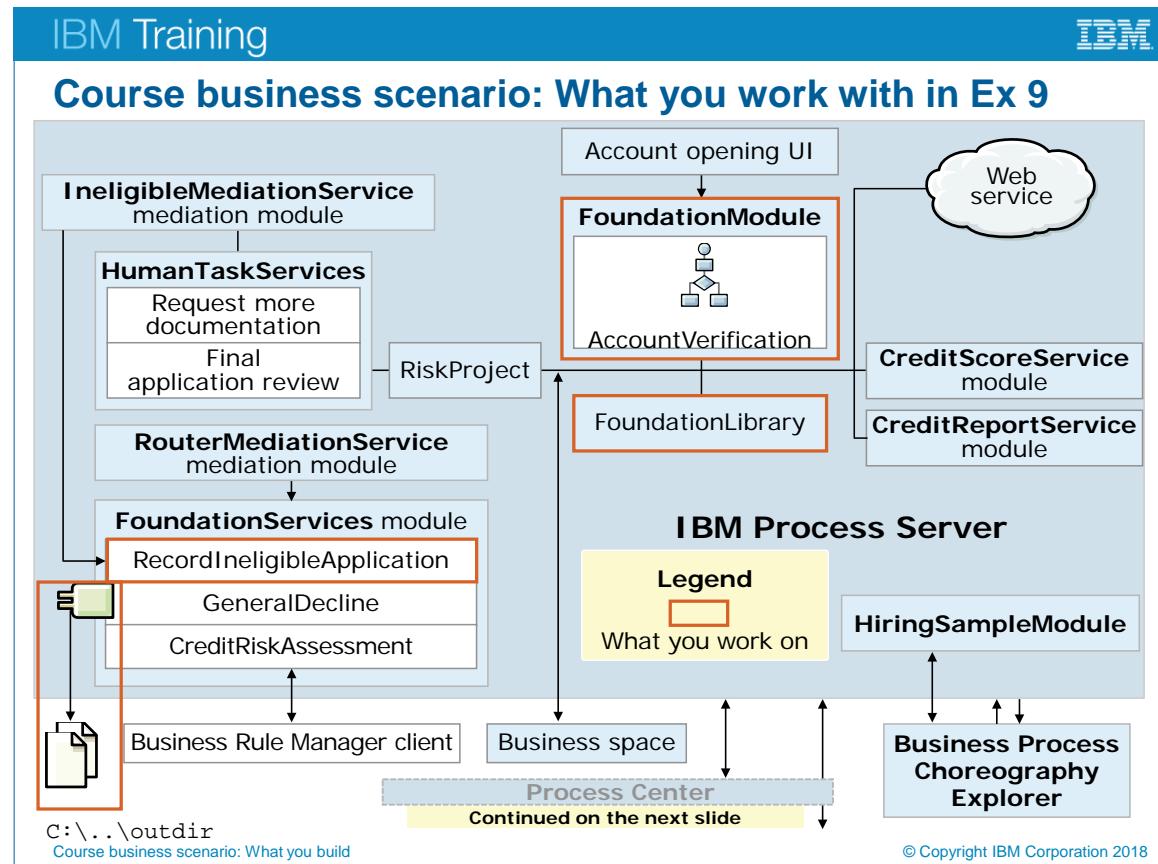
### Assemble SCA components in Exercise 9

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 9

## Exercise 10: Creating mediation services, part I

Creating mediation services, part I

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 10: Creating mediation services, part I*

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

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

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

IBM Training IBM

## Components that are required for Exercise 10

Prebuilt components that are imported in this lab:

- 1. FoundationModule**
- 2. CreditScoreService**
- 3. FoundationLibrary**
- 4. FoundationServices**
- 5. CWYFF\_FlatFile**

```

<call resources>
  + CreditScoreService
  + CWYFF_FlatFile
  + FoundationLibrary
  + FoundationModule
  + FoundationServices
  - IneligibleMediationService
    + Assembly Diagram
      - IneligibleMediationExport
    + Dependencies
    + Integration Logic
      + Mediation Flows
        - IneligibleMediation
    + Data
    + Interfaces
    + Transformations
      + Data Maps
      + xslt
    - InputToOutput
  
```

New components that you create in this lab:

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

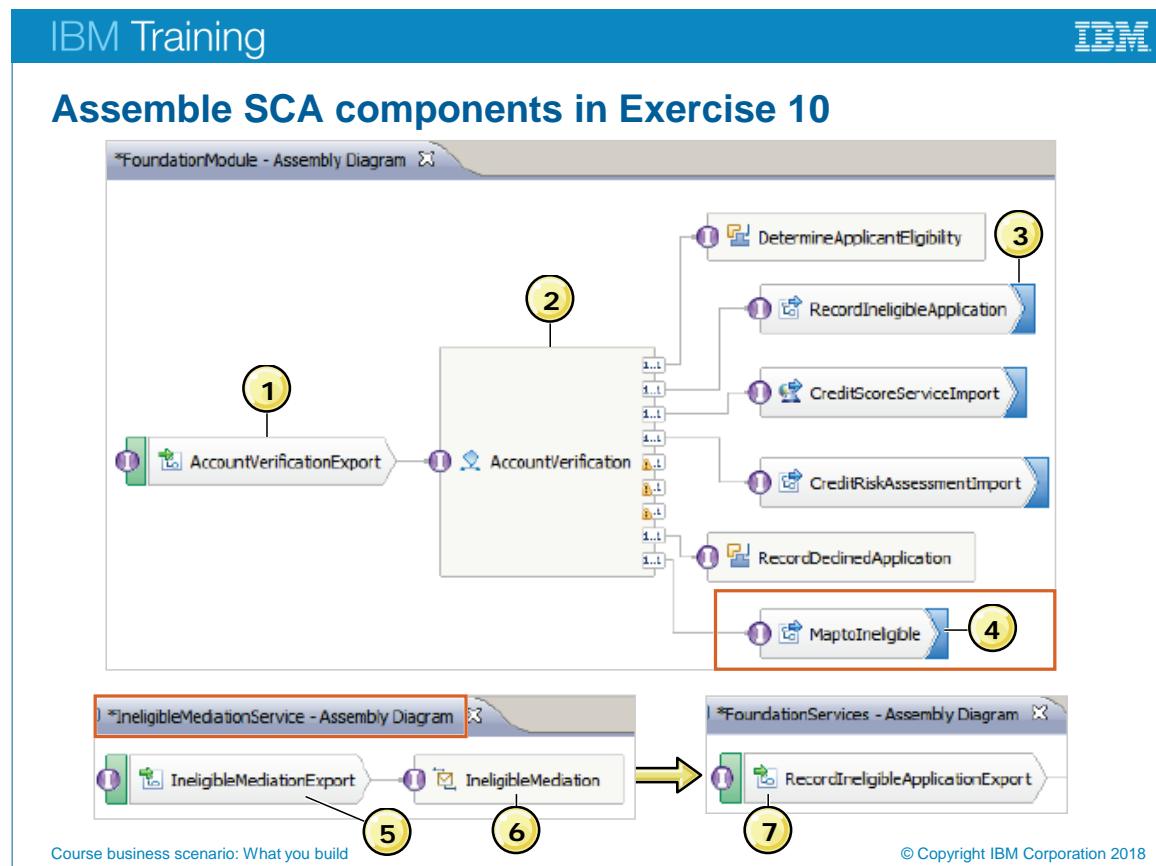
Course business scenario: What you build © Copyright IBM Corporation 2018

### Components that are required for Exercise 10

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.

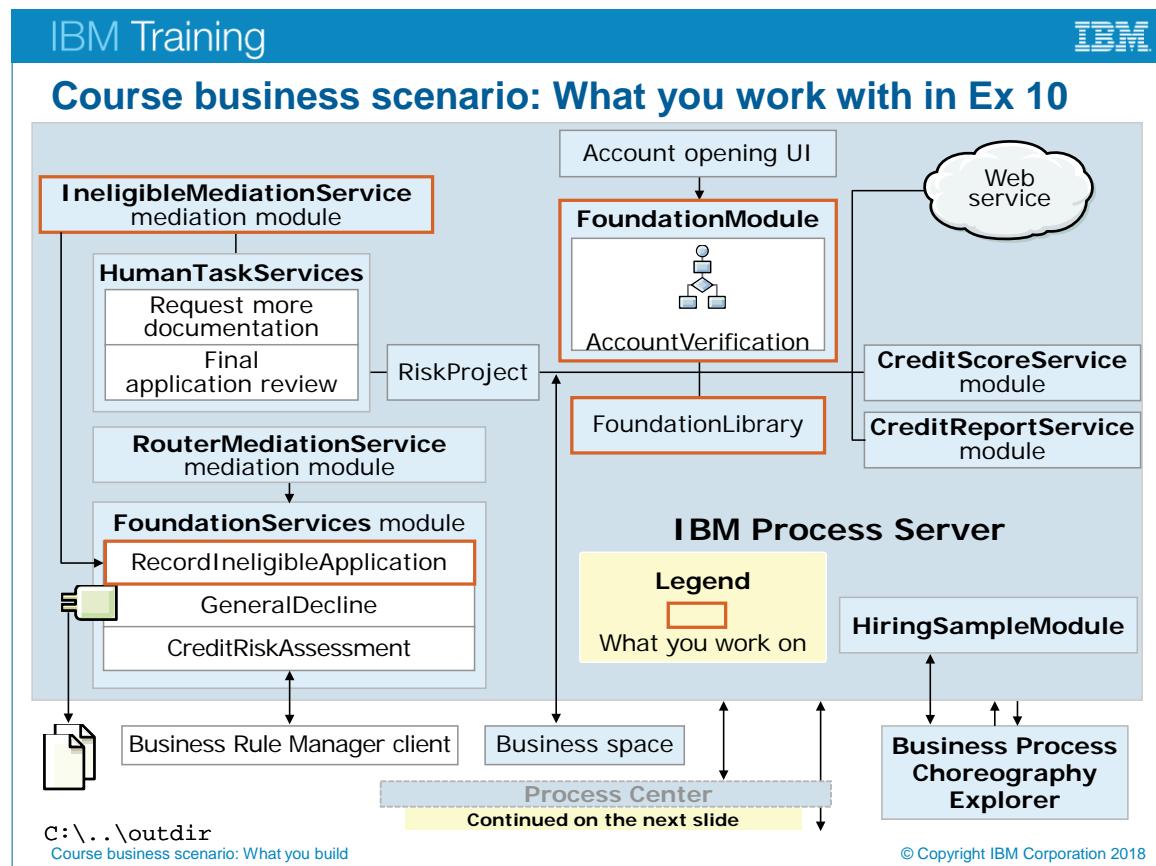


### Assemble SCA components in Exercise 10

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 10

## Exercise 11: Creating mediation services, part II

Creating mediation services, part II

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 11: Creating mediation services, part II*

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

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

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

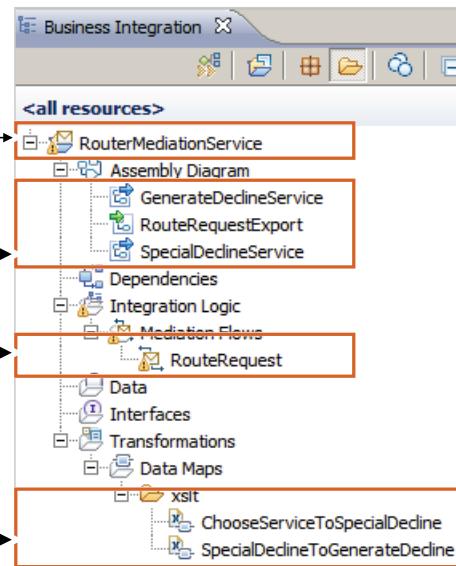
## Components that are required for Exercise 11 (1 of 2)

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



Course business scenario: What you build

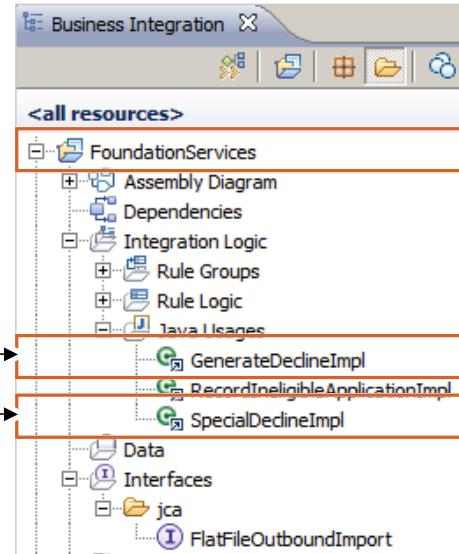
© Copyright IBM Corporation 2018

*Components that are required for Exercise 11*

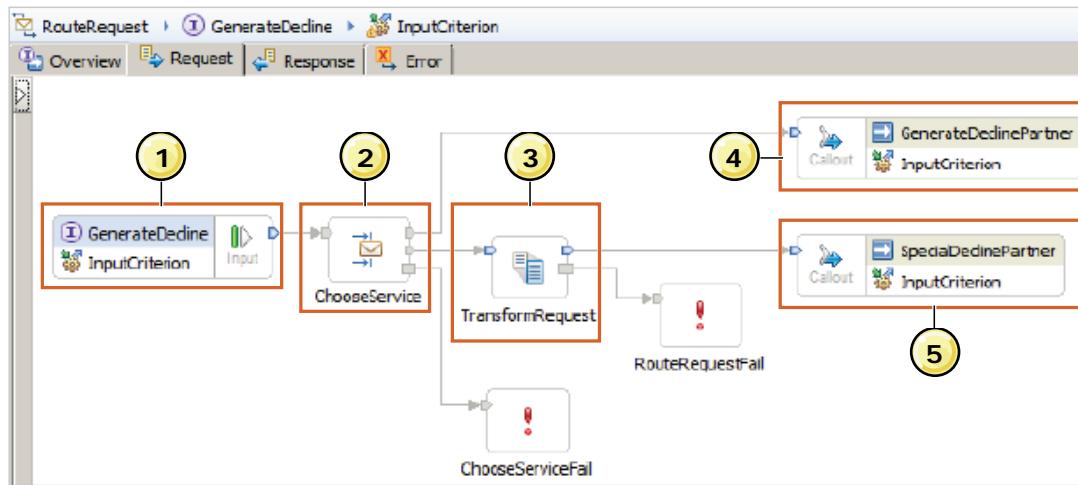
## Components that are required for Exercise 11 (2 of 2)

Java components that you add in this lab:

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



## Create the RouteRequest mediation flow in Exercise 11

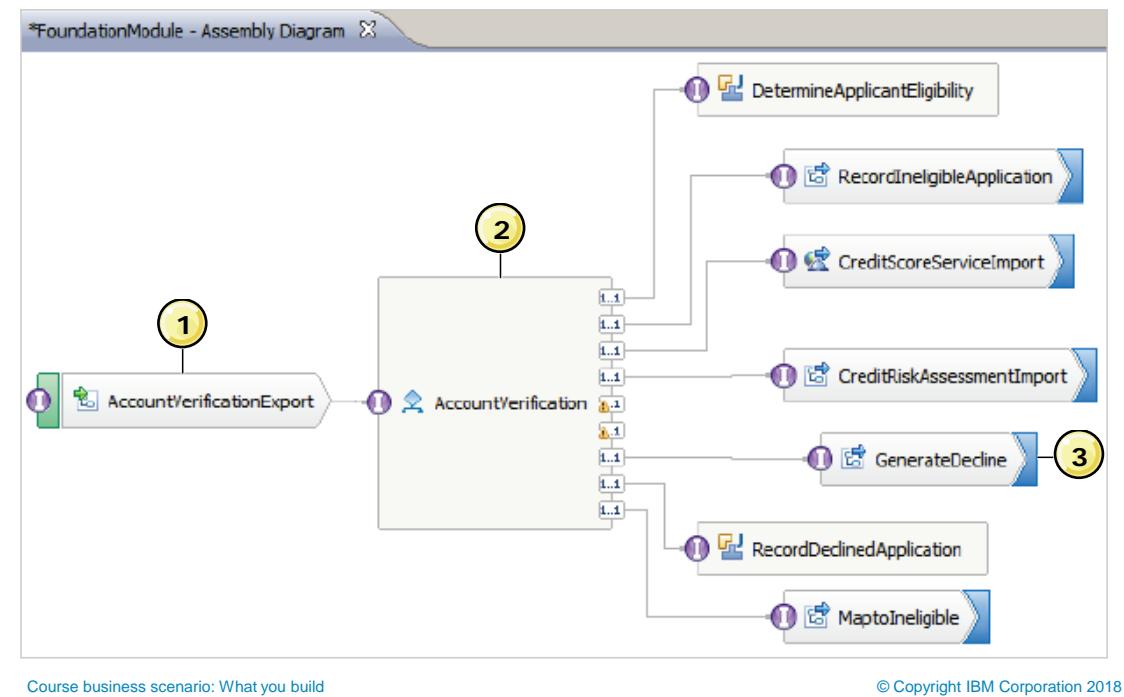


Course business scenario: What you build

© Copyright IBM Corporation 2018

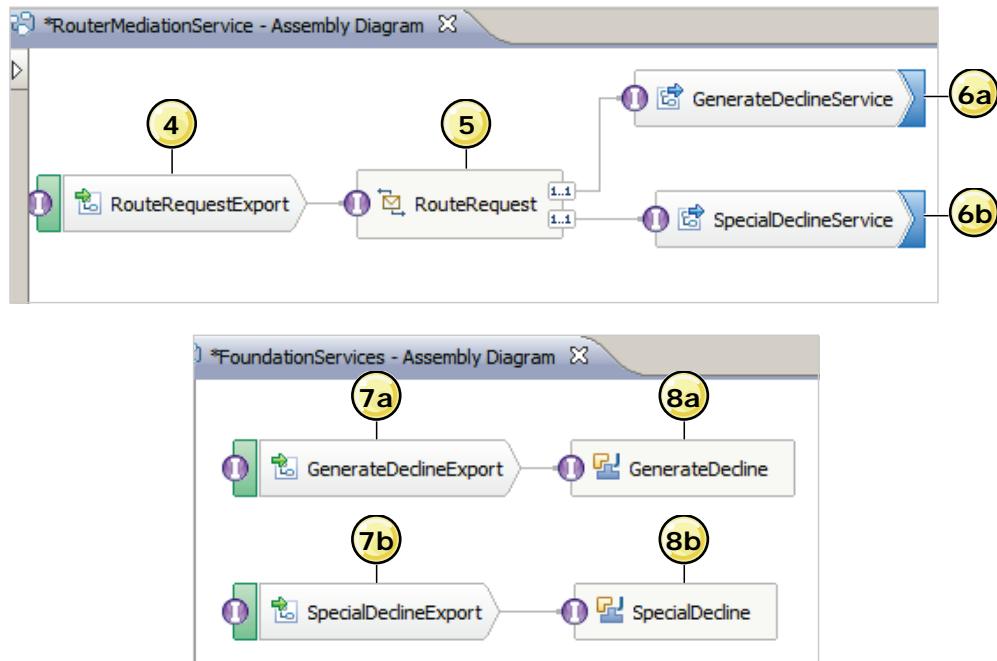
*Create the RouteRequest mediation flow in Exercise 11*

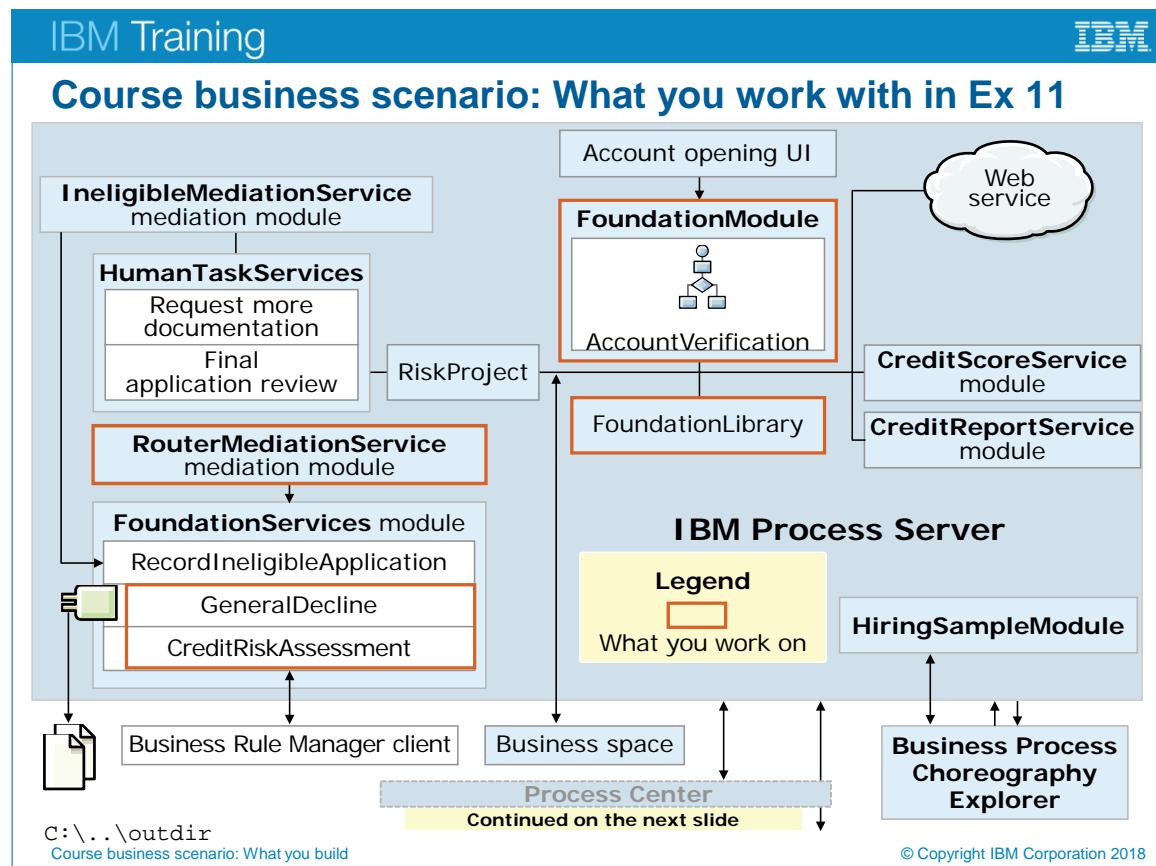
## Assemble SCA components in Exercise 11 (1 of 2)



Assemble SCA components in Exercise 11

## Assemble SCA components in Exercise 11 (2 of 2)





Course business scenario: What you work with in Ex 11

## Exercise 12: Exploring Business Space

Exploring Business Space

Course business scenario: What you build

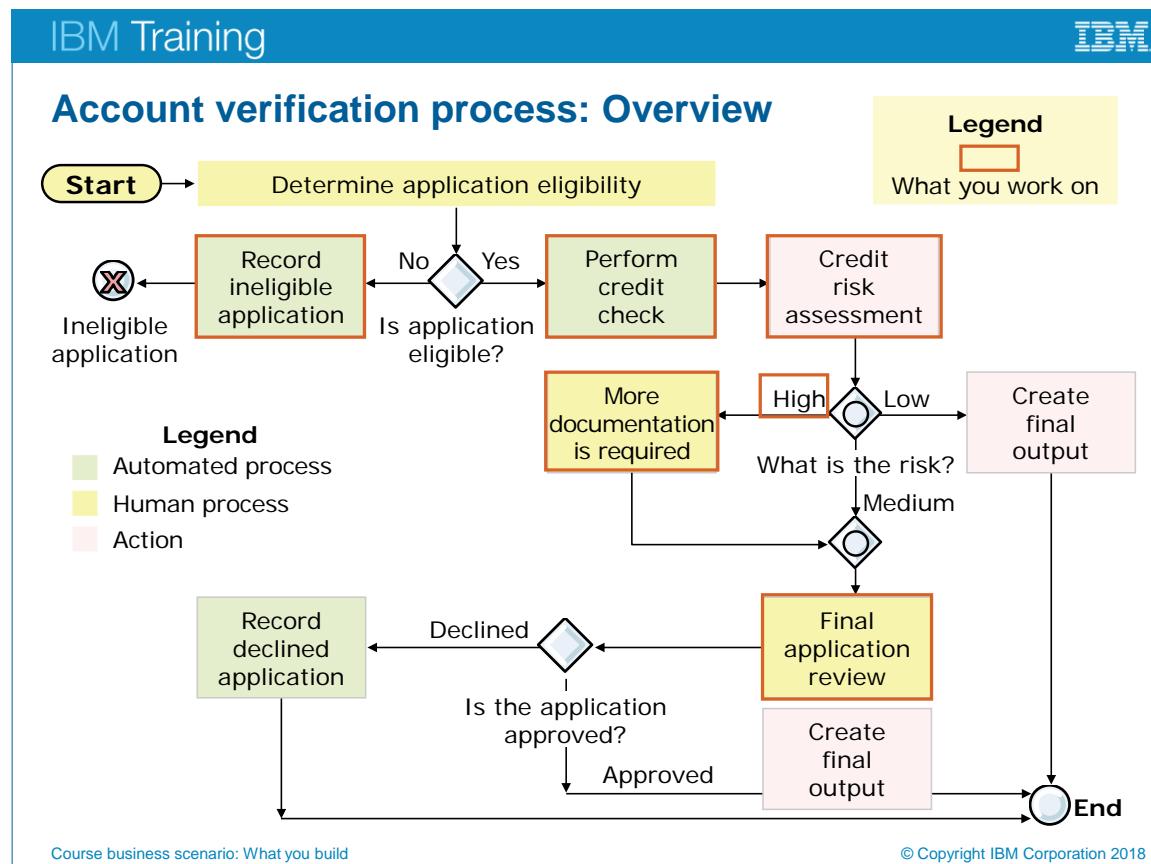
© Copyright IBM Corporation 2018

*Exercise 12: Exploring Business Space*

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

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

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



Account verification process: Overview

IBM Training IBM

## Create business space in Exercise 12

The screenshot shows the IBM Business Space interface. At the top, there's a navigation bar with links like Home, Process Portal, Go to Spaces, Manage Spaces, and Actions. Below the navigation is a search bar labeled "All Widgets (42)". The main area is titled "Tasks" and shows a list of tasks: "Final Application Review" (Status: Available, Priority: Very high). A context menu is open over this task, with "Accept" highlighted. Other options in the menu include Edit, View, Transfer, Return, Delete, Escalate, and Postpone. The bottom of the screen has a footer with copyright information.

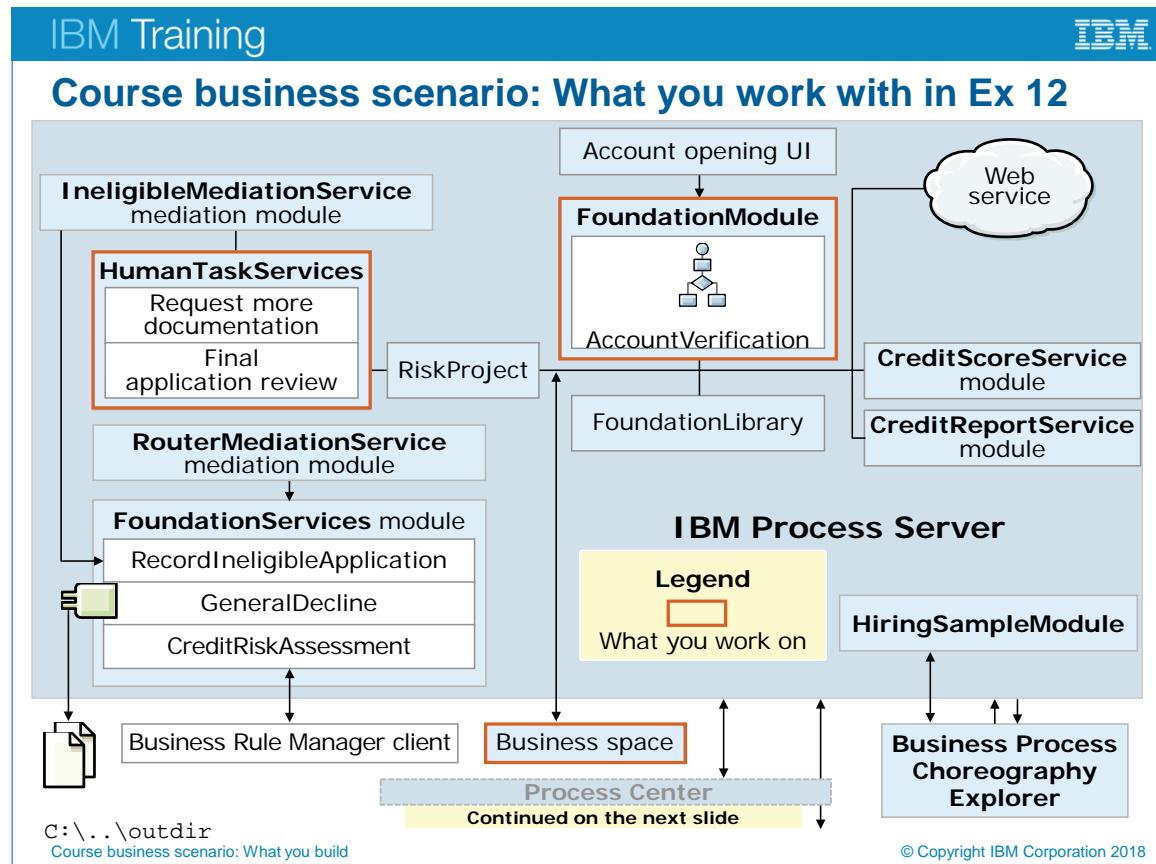
**Tasks**  
All - Unassigned and my work ▾ Group by: Owner ▾  
Actions ▾ Select the process an  
Name Status Priority Due date  
Not set  
Final Application Review Available Very high

**Process Information**  
Edit  
Accept  
View  
Transfer  
Return  
Delete  
Escalate  
Postpone

**MyBusinessSpace**  
Initiate Tasks and Processes Work on tasks Manage Processes Manage Tasks Organize View  
All Widgets (42)  
All Widgets (42)  
Business Configuration (1)  
Human Task Management (13)  
Problem Determination (2)  
**Solution Administration (7)**  
Solution Operation (5)  
User Management (1)  
Viewers (1)  
Widgets - deprecated (10)  
Wiring (2)  
Human Workflow Diagram  
Business Calendars Business Categories  
Business Rules Create Tasks  
Escalations Google Gadgets  
Mediation Policy Administration Module Administration

Course business scenario: What you build © Copyright IBM Corporation 2018

*Create business space in Exercise 12*



Course business scenario: What you work with in Ex 12

## Exercise 13: Using component tests

Using component tests

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 13: Using component tests*

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

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

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

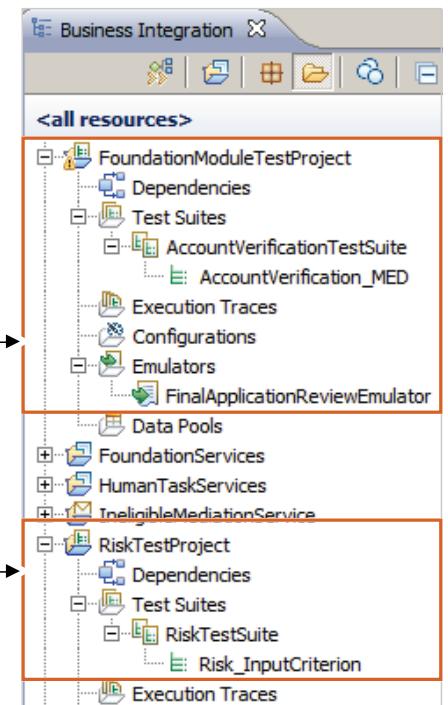
## Components that are required for Exercise 13

Prebuilt components that are imported in this lab:

1. All components from completed Exercise 13

New components that you create in this lab:

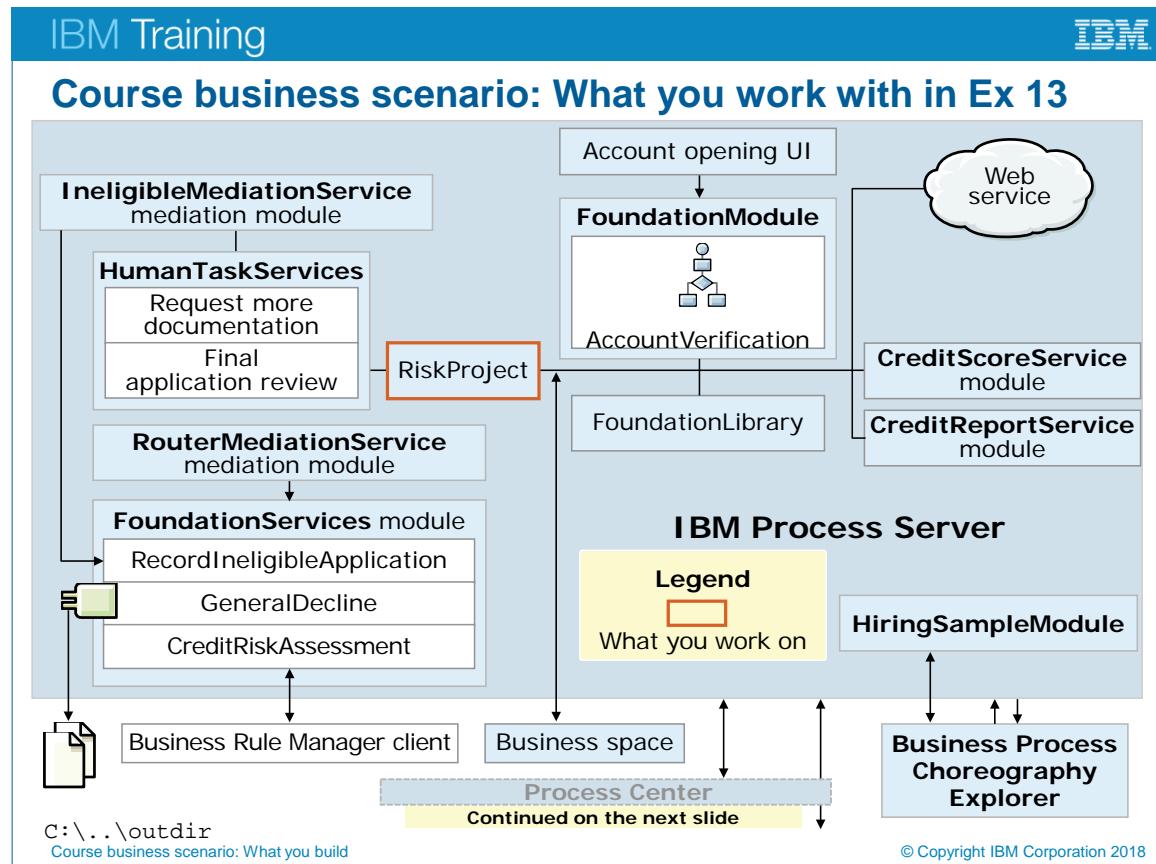
1. **FoundationModuleTestProject** and **RiskTestProject** component test projects
2. **AccountVerificationTestSuite** and **RiskTestSuite** test suites
3. **FinalApplicationReviewEmulator** emulator



Course business scenario: What you build

© Copyright IBM Corporation 2018

*Components that are required for Exercise 13*



Course business scenario: What you work with in Ex 13

## Exercise 14: Exploring IBM Process Center

Exploring IBM Process Center

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Exercise 14: Exploring IBM Process Center*

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

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

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

## Components that are required for Exercise 14

Prebuilt components that are imported in this lab:

1. All components from the completed Exercise 13

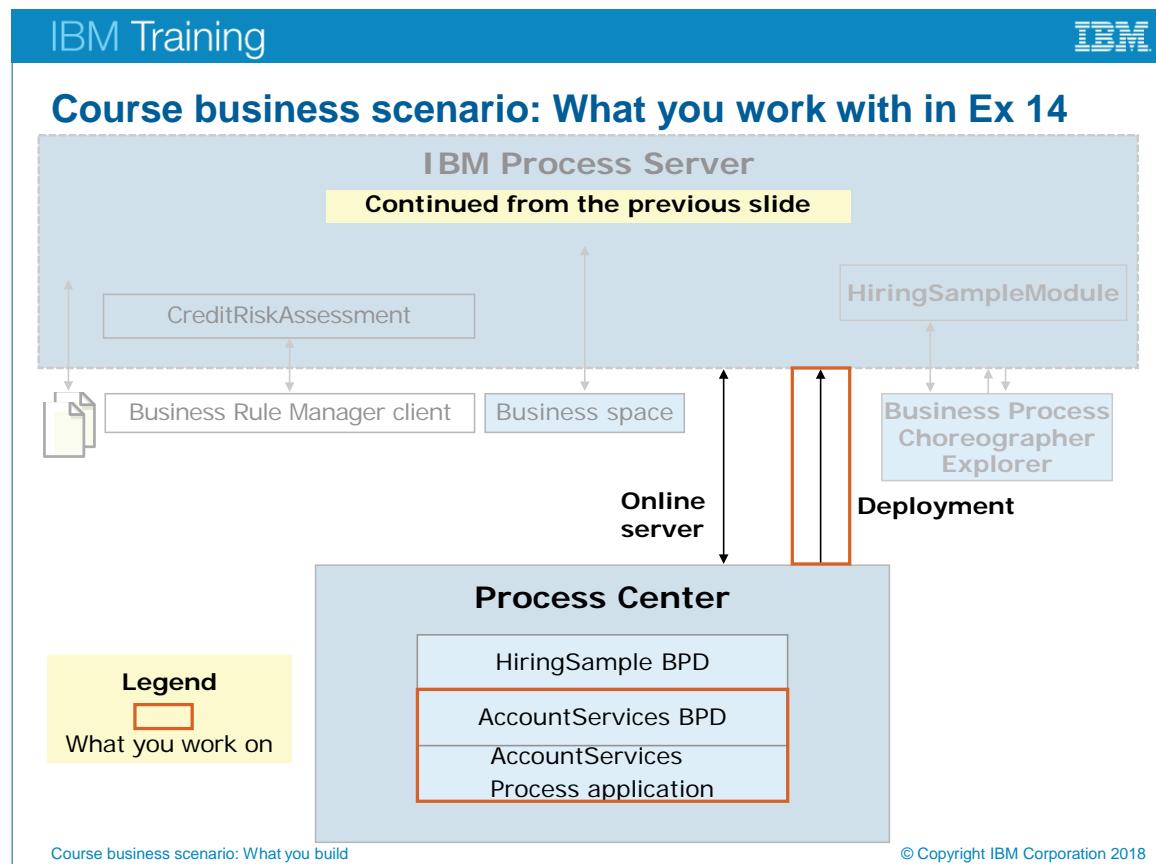
New components that you create in this lab:

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



 ASA snapshot 1 (ASAS1)  (New) 

Created on 6/21/14 by pdeadmin  
Not Yet Deployed to Process Center Server  
Currently Installed:  
 UTEServer(ws2008r2x64.wetraining.com) - 0 instances



Course business scenario: What you work with in Ex 14

## Exercise 15: Implementing Advanced Integration services

Implementing Advanced Integration services

Course business scenario: What you build

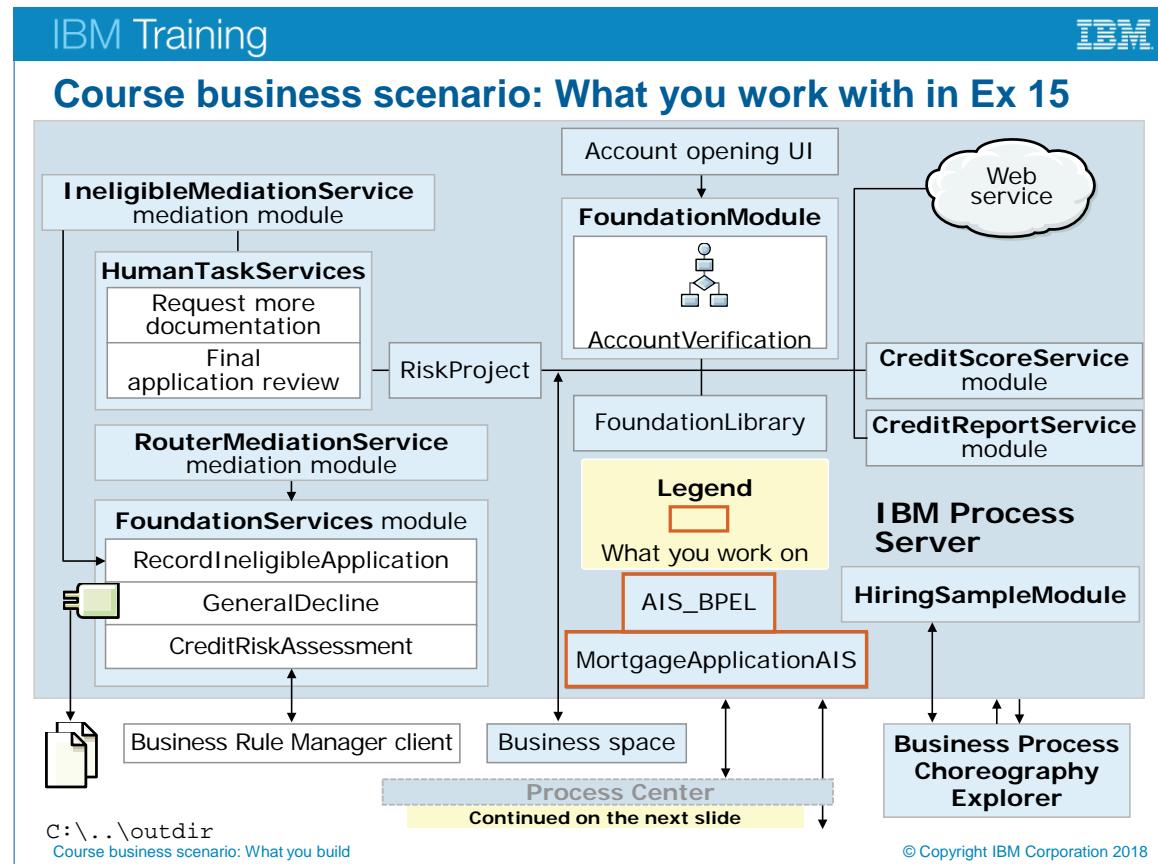
© Copyright IBM Corporation 2018

*Exercise 15: Implementing Advanced Integration services*

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

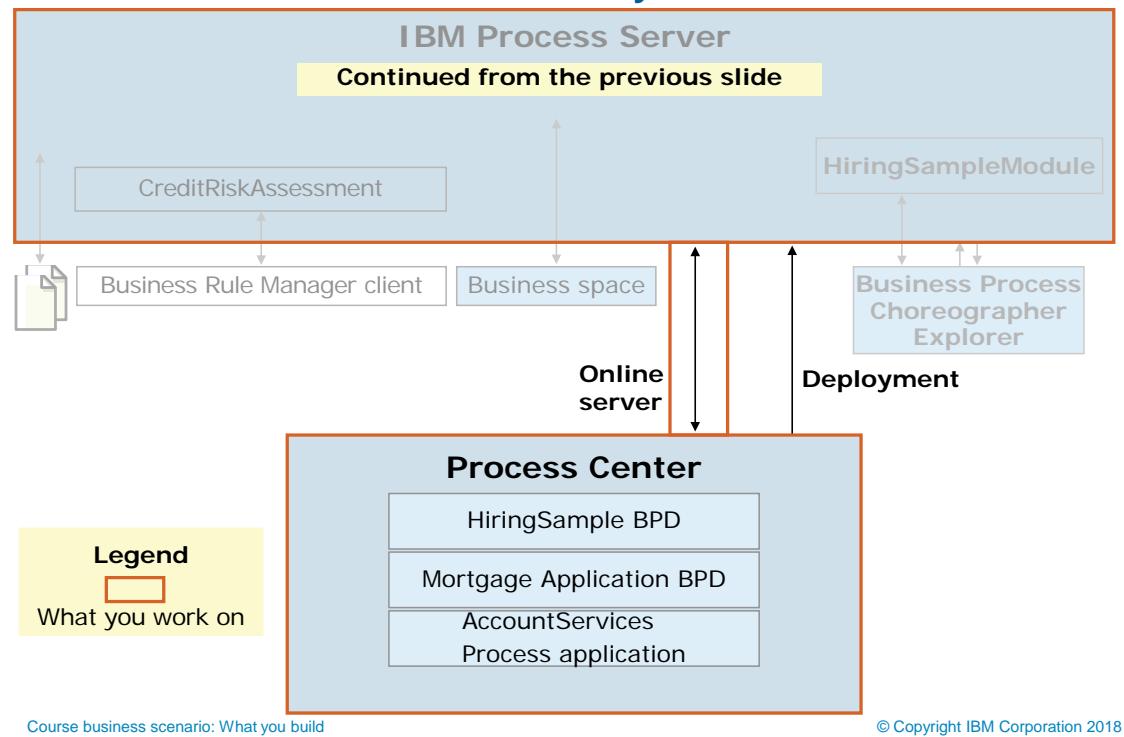
- In this exercise, you create Advanced Integration service, which IBM Process Designer uses to invoke IBM Integration Designer implementations
  - The capabilities of Advanced Integration with IBM Business Process Manager Advanced open a whole world of service orientation
  - Access to process choreography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

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



Course business scenario: What you work with in Ex 15

## Course business scenario: What you work with in Ex 15



## Unit summary

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

Course business scenario: What you build

© Copyright IBM Corporation 2018

*Unit summary*

## **Unit 4** IBM Integration Designer overview

The slide features a blue header bar with 'IBM Training' on the left and the IBM logo on the right. The main content area has a light gray diagonal striped background. It contains the title 'IBM Integration Designer overview' in large blue font, followed by the subtitle 'IBM Business Process Manager V8.6' in smaller blue font. At the bottom, there is a copyright notice: '© Copyright IBM Corporation 2018' and 'Course materials may not be reproduced in whole or in part without the written permission of IBM.'

IBM Training

IBM

**IBM Integration Designer  
overview**

**IBM Business Process Manager V8.6**

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## 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

### *Unit objectives*

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

## Topics

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

# Introduction to IBM Integration Designer

IBM Integration Designer overview

© Copyright IBM Corporation 2018

*Introduction to IBM Integration Designer*

IBM Training 

## 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:  
 - Title: About IBM Integration Designer 8.5.7  
 - Subtitle: IBM Integration Designer  
 - Version: Version: 8.5.7.0  
 - Build id: Build id: 20160211-201551.0.linux  
 - Description: 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.  
 - Copyright: (c) Copyright IBM Corporation and others 2000, 2016. All rights reserved.  
 - Icons: A row of 12 small icons representing various tools and components.

IBM Integration Designer overview © Copyright IBM Corporation 2018

### *IBM Integration Designer: Create, assemble, and test applications*

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

### IBM Integration Designer roles

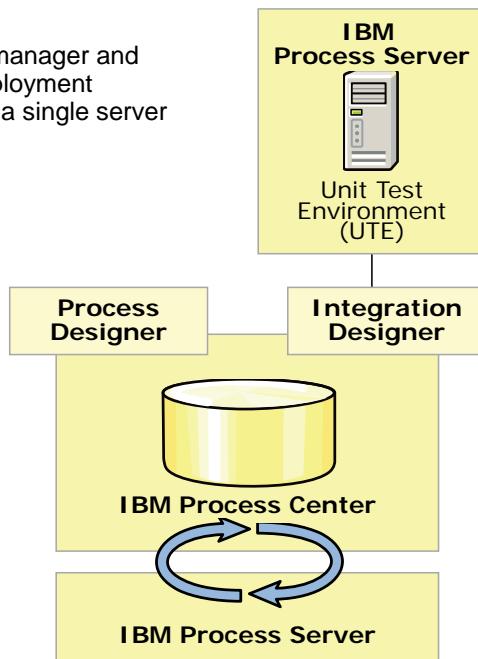
The primary focal group for IBM Integration Designer is integration developers who are not required to be fluent Java programmers. Therefore, IBM Integration Designer hides advanced implementation details as much as possible. However, the base technology is still Rational Application Developer. Application 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 V8.6

- 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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

### *Installing IBM Business Process Manager V8.6*

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.

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

### *IBM Installation Manager*

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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

### *IBM Integration Designer: Platform architecture*

IBM Integration Designer 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

### *Installing IBM Integration Designer*

## 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

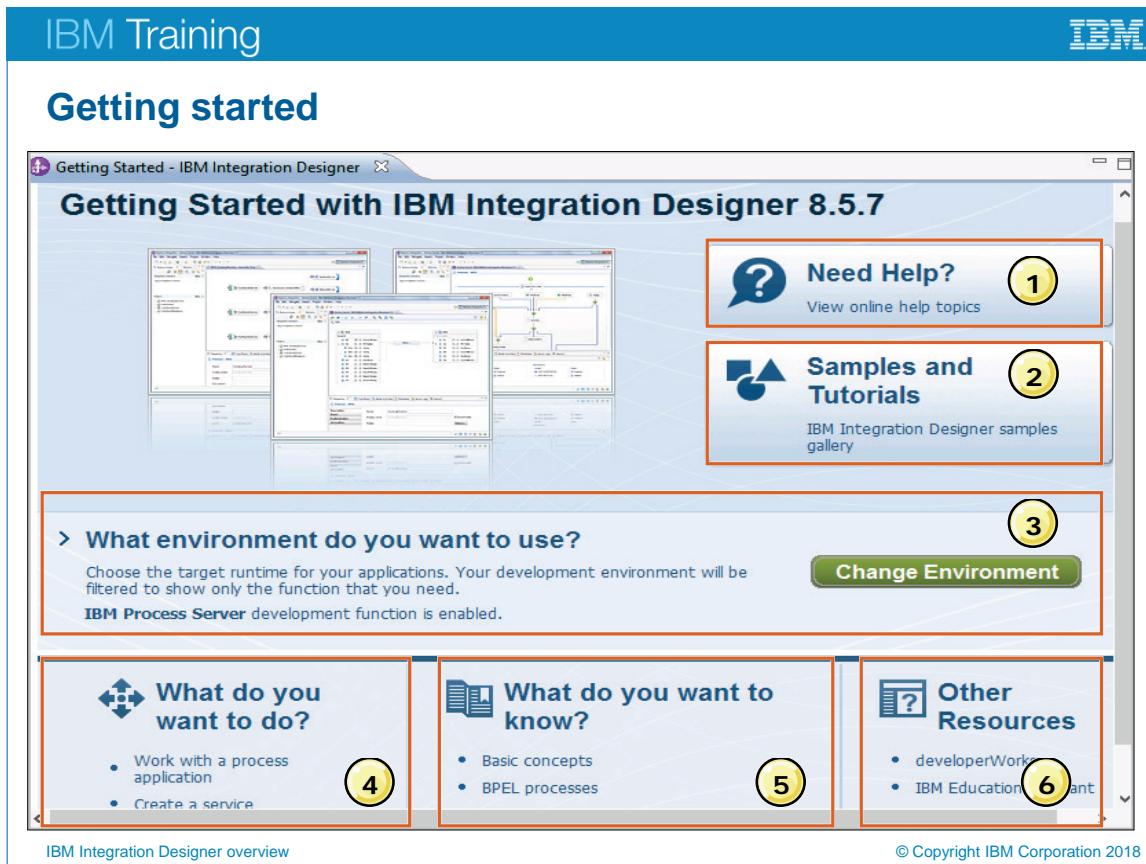
A default profile for IBM Business Process Manager—Process Server is selected on the Features page. You can use this profile to use the test environment immediately. By default, **Start working with the IBM Process Center first** is selected so that you can work with the Process Center. You can use the test environment installation to create an IBM Process Server profile. You can customize the environment to include multiple profiles. 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.

## Application composition and deployment

IBM Integration Designer overview

© Copyright IBM Corporation 2018

*Application composition and deployment*



## Getting started

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.

IBM Training IBM

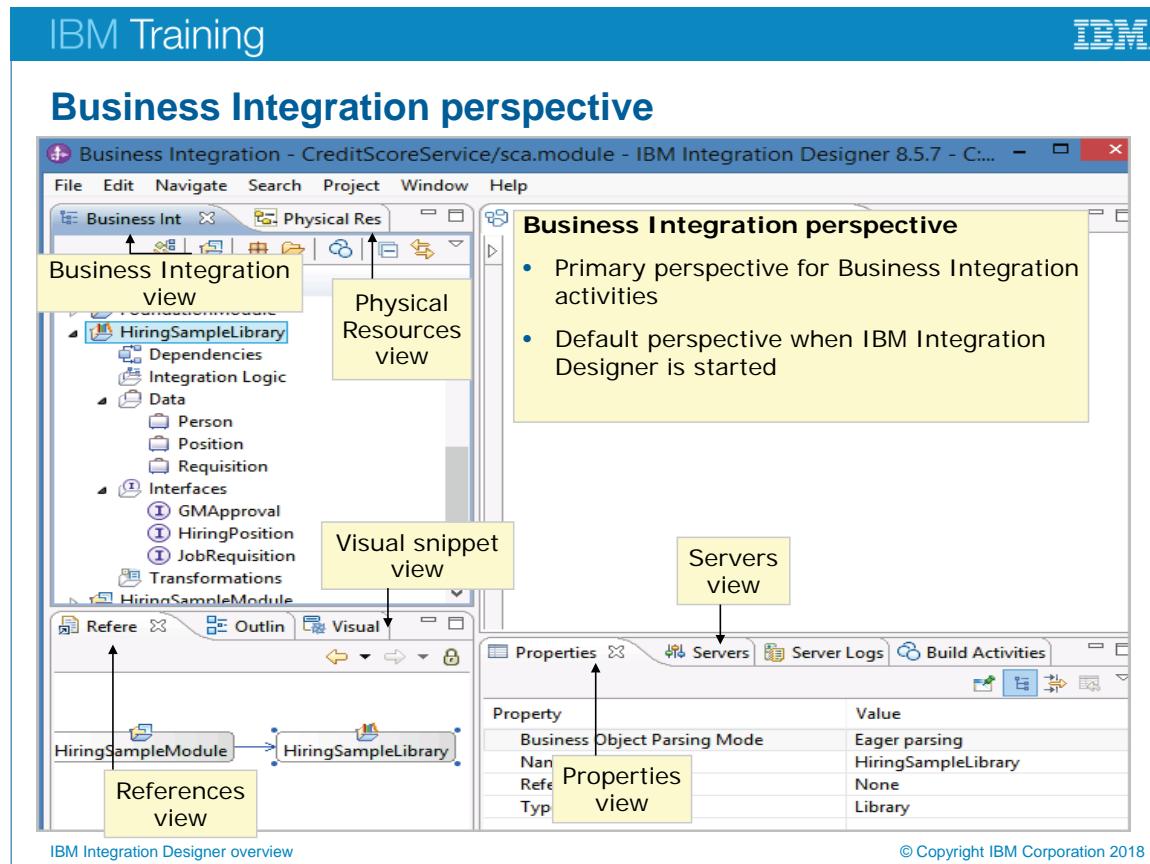
## Application composition and deployment

- Use the Business Integration perspective to develop business integration projects

IBM Integration Designer overview © Copyright IBM Corporation 2018

### *Application composition and deployment*

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*

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.

IBM Training IBM

## Features overview: Visual tools (1 of 4)

The diagram illustrates the visual tools available in IBM Integration Designer. It shows two main editors:

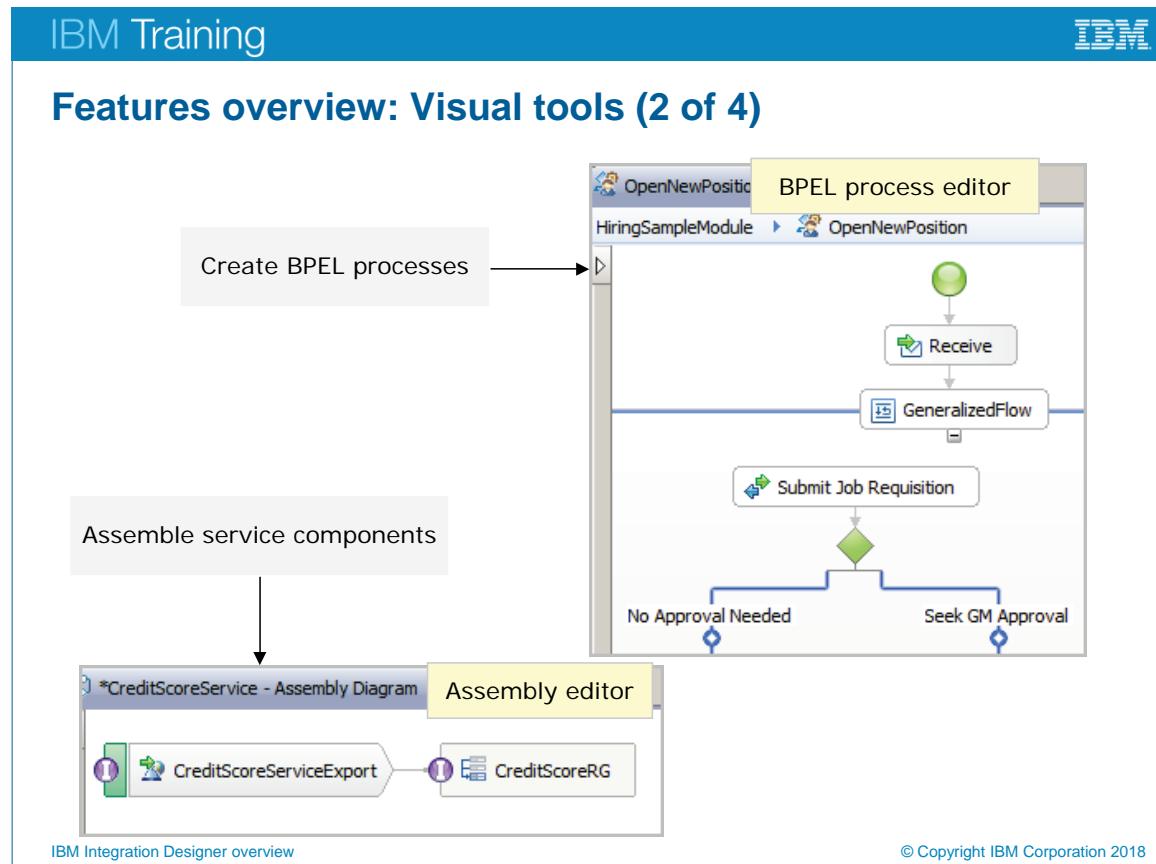
- Business Object editor:** This editor is used for creating business objects. It displays a list of fields for a "CreditCheckRequest" object, including "accountNumber" (string), "companyID" (string), "creditScore" (int), "dateRequested" (string), and "CompanyID1" (CompanyID). A separate list of fields for a "CompanyID" object is also shown, including "Name" (string), "Address" (string), "Phone" (string), "Email" (string), and "Contact" (string).
- Interface editor:** This editor is used for creating service interfaces. It shows the "CreditScoreService" interface with one operation, "calculateCreditScore". The operation has one input, "request" (of type CreditCheckRequest), and one output, "calculateCreditScoreReturn" (of type CreditCheckRequest).

Two features are highlighted:

- Create business objects:** An arrow points from this text to the Business Object editor.
- Create custom visual snippets:** A downward arrow points from this text to the Interface editor.

IBM Integration Designer overview © Copyright IBM Corporation 2018

### Features overview: Visual tools



IBM Training IBM

## Features overview: Visual tools (3 of 4)

Defining business rules

Conditions				
request.companyName	"IBM"	"AbcCo"	"ACME"	Otherwise
calculateCreditScoreReturn.creditScore	11	1	6	1

Decision table editor

```

graph TD
    InitialState((InitialState)) --> Ready((Ready))
    Ready -- "DepositMoney" --> Ready
    Ready -- "UnPlug" --> Ready
    Ready -- "Set Final" --> Ready
    Ready -- "Restock" --> Ready
    Ready -- "GetState" --> Ready
    Ready -- "State Info" --> Ready
    Ready -- "ReturnMoney" --> Ready
    Ready -- "Deposit Return?" --> Ready
  
```

Creating a business state machine

IBM Integration Designer overview © Copyright IBM Corporation 2018

IBM Training IBM

## Features overview: Visual tools (4 of 4)

**Data map editor**

\*MapFromCreditCheckSvc X

MapFromCreditCheckSvc

CreditCheckRequest

- <Click to filter...>
- accountNumber [0..1] string
- companyName [0..1] string
- creditScore [0..1] int
- dateRequested [0..1] string

CustomerApplication

- <Click to filter...>
- accountNumber
- applicationDate
- applicationDecision
- comments
- companyName

Mapping data

Creating custom visual snippets

Implementation:  Visual  Java  Invoke

Main group

x+y Expression

Standard...

Visual snippet editor

StateInfo.MoneyDeposited

greater than or equal to

StateInfo.UnitPrice

IBM Integration Designer overview © Copyright IBM Corporation 2018

IBM Training IBM

## 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.

The screenshot shows the 'Servers' view in IBM Integration Designer. A callout box labeled 'IBM Process Server running in IBM Integration Designer' points to the server entry in the list. Another callout box labeled 'Installed applications' points to a group of applications under the 'CreditScoreServiceApp' entry. On the toolbar above the list, there are 'Start server' and 'Stop server' buttons, each with a corresponding icon. The 'Start server' button has an orange border, and the 'Stop server' button has a red border.

IBM Process Server v8.5.7 at localhost [Started, Synchronized]  
 CreditScoreServiceApp [Started, Synchronized]  
 CreditScoreService [Synchronized]  
 CreditScoreServiceWeb [Synchronized]  
 FoundationModuleApp [Started, Synchronized]  
 HiringSampleModuleApp [Started, Synchronized]  
 HiringSampleTestApp [Started, Synchronized]

Properties Servers Server Logs Problems Start server Stop server

IBM Integration Designer overview © Copyright IBM Corporation 2018

### IBM Process Server unit test environment

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 Training

IBM

## IBM Process Server unit test environment (2 of 3)

The screenshot shows the 'Server Runtime Environments' configuration window and the 'Servers' view in the IBM Integration Designer.

**Server Runtime Environments Window:**

- Title Bar:** Select Windows > Preferences > Preferences (Filtered)
- Left Sidebar:** type filter text, General, Business Integration, Help, Importer, Run/Debug, Samples and Tutorials Settings, Server (with sub-options: Application Restart, Audio, Launching, Profilers, **Runtime Environments**, Security, WebSphere Application).
- Main Content:** Server Runtime Environments. Add, remove, or edit server runtime environments. Server runtime environments table:
 

Name	Type
IBM Process Center v8.5.7	IBM Process Center v8.5.7
<b>IBM Process Server v8.5.7</b>	IBM Process Server v8.5.7

**Servers View:**

- Toolbar:** Properties, Servers, Server Logs, Problems.
- List:** IBM Process Server v8.5.7 at localhost [Started, Synchronized]

Arrows indicate the relationship between the selected 'Runtime Environments' in the sidebar and the 'IBM Process Server v8.5.7' entry in the Servers view.

IBM Integration Designer overview © Copyright IBM Corporation 2018

IBM Training IBM

## IBM Process Server unit test environment (3 of 3)

**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

Server configuration editor

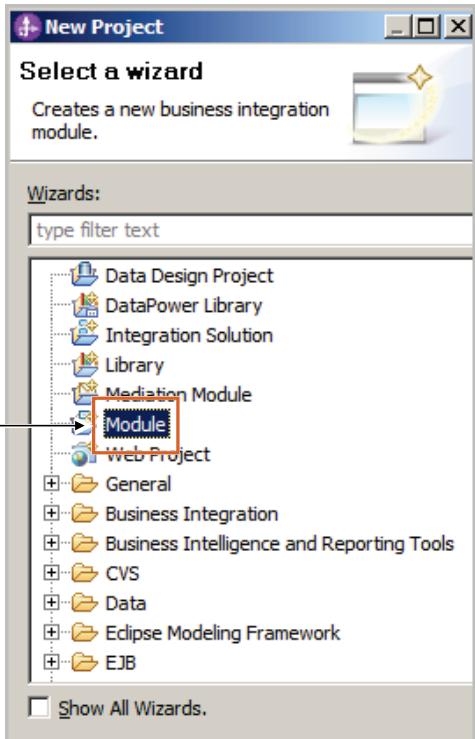
Right-click the server and click Open

IBM Integration Designer overview © Copyright IBM Corporation 2018

IBM Training IBM

## 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



IBM Integration Designer overview © Copyright IBM Corporation 2018

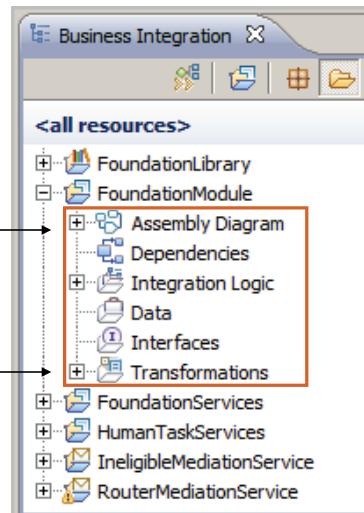
### *Module*

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.

## 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



### Module components

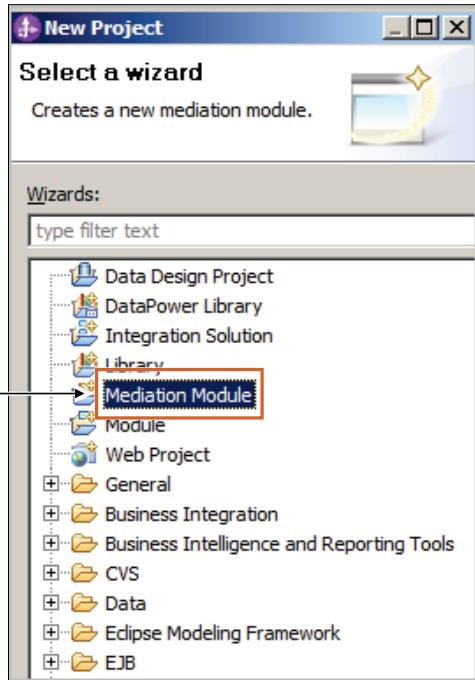
The module provides the business services, which are modeled as Service Component Architecture (SCA) components that are wired together in a module assembly. This module can contain all the resources that are used in the service, but these resources are private and can be used only within the module. 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.

IBM Training IBM

## 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



IBM Integration Designer overview © Copyright IBM Corporation 2018

### Mediation Module

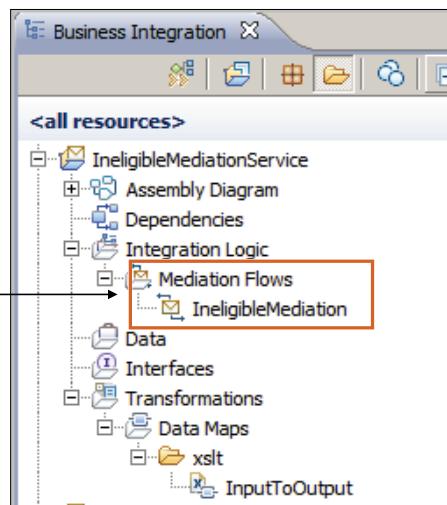
Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

### Mediation Module components

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.

IBM Training IBM

## 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

The screenshot shows the 'New Project' wizard interface. The title bar says 'New Project' and 'Select a wizard'. Below it, a description says 'Creates a new library.' with an icon of a book. A list of wizards is shown in a tree view under 'Wizards:'. The 'Library' option is highlighted with a red box. At the bottom, there's a detailed view of the 'Library' project structure, also enclosed in a red box. The structure includes 'FoundationLibrary' with sub-options like 'Dependencies', 'Integration Logic', 'Data', 'Interfaces', 'Transformations', and 'Web Service Ports'.

IBM Integration Designer overview © Copyright IBM Corporation 2018

### *Library components*

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

### *Creating modules and libraries*

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.](#)

### Library deployment

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.

IBM Training IBM

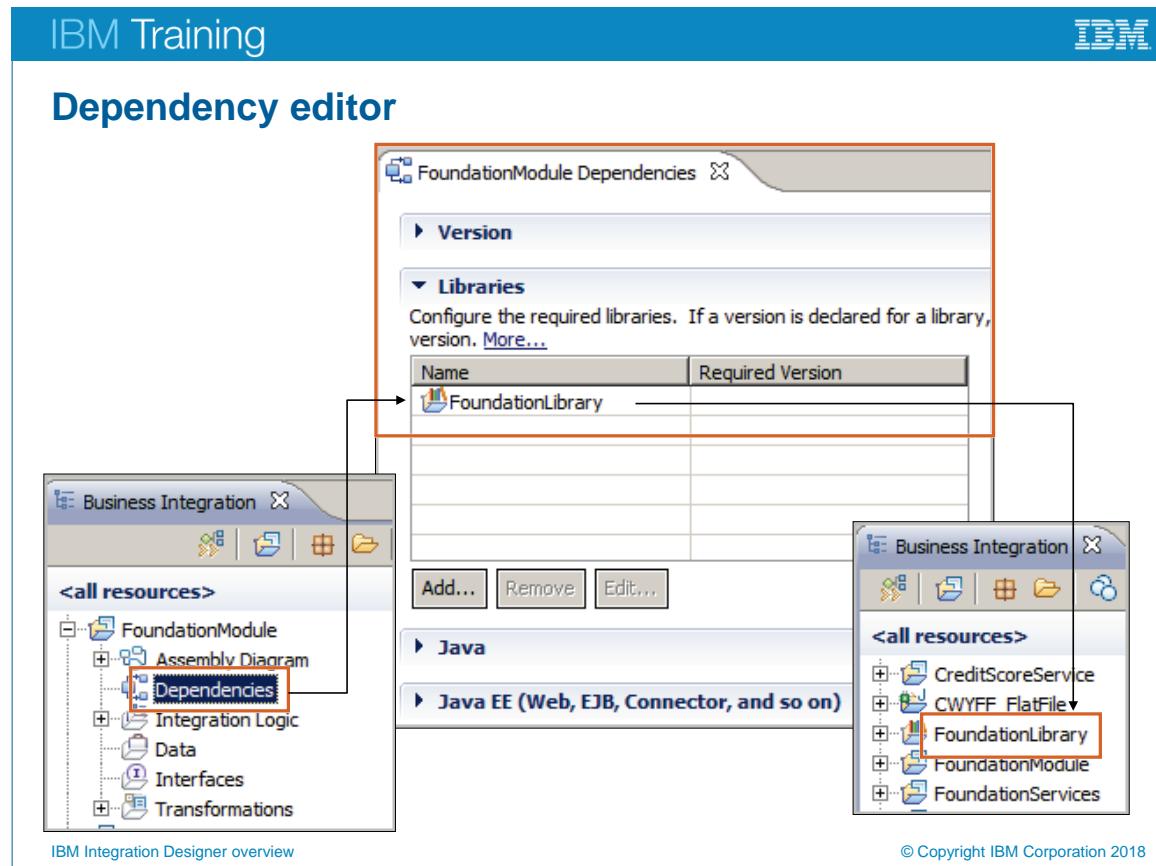
## Business Integration view

- 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

IBM Integration Designer overview

© Copyright IBM Corporation 2018

*Business Integration view*



### Dependency editor

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.

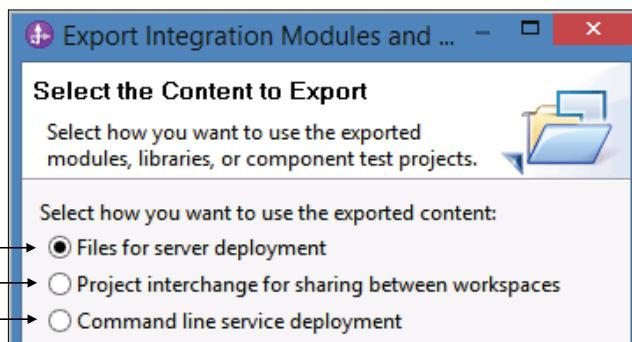
## Packaging modules

- 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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

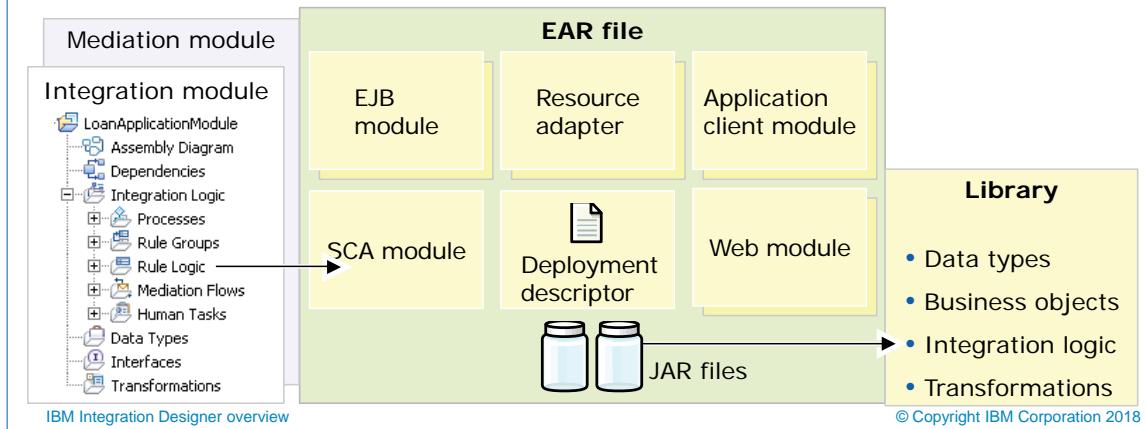
### Packaging modules

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)



### Module deployment

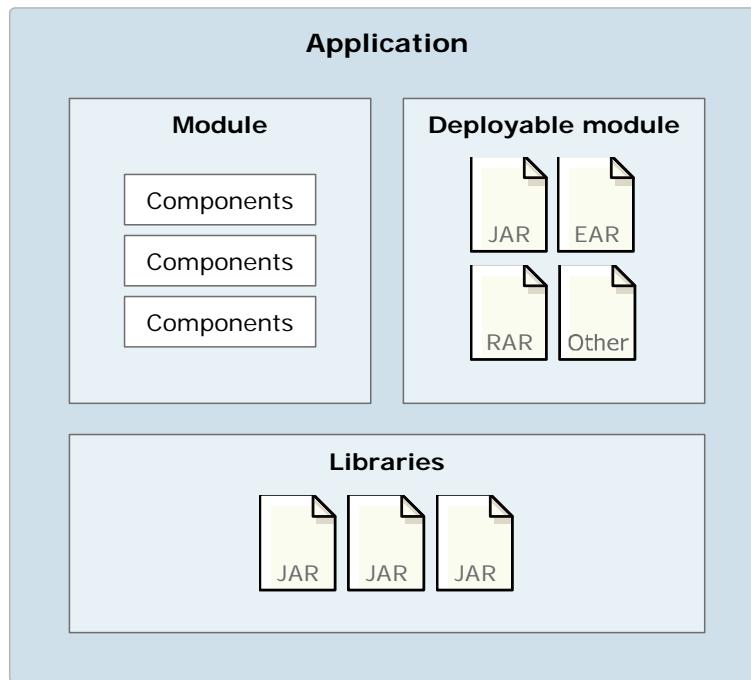
Enterprise archive (EAR) files are compressed archive files that contain Java EE application components such as web applications (packaged in WAR files) and EJB beans. Java EE application containers import EAR files into the application environment at run time and deploy the components within each EAR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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

### *Application, module, and library relationship*

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. Below the navigation bar, there is a search bar with the placeholder 'Sort By: Recently Updated' and a dropdown arrow, followed by the word 'All'. The main area lists four items:

	Item Name	Rating	Last Updated
	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

IBM Integration Designer overview

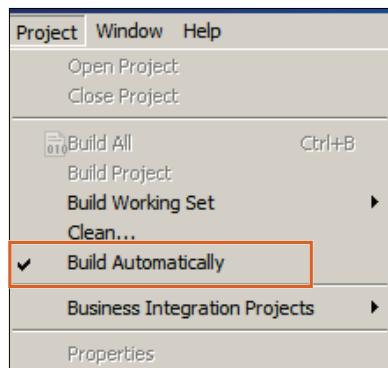
© Copyright IBM Corporation 2018

### Using the process center perspective

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



## Artifacts in IBM Integration Designer and Process Designer

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

IBM Integration Designer overview

© Copyright IBM Corporation 2018

### Artifacts in IBM Integration Designer and Process Designer

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.

## Documentation and organization capabilities

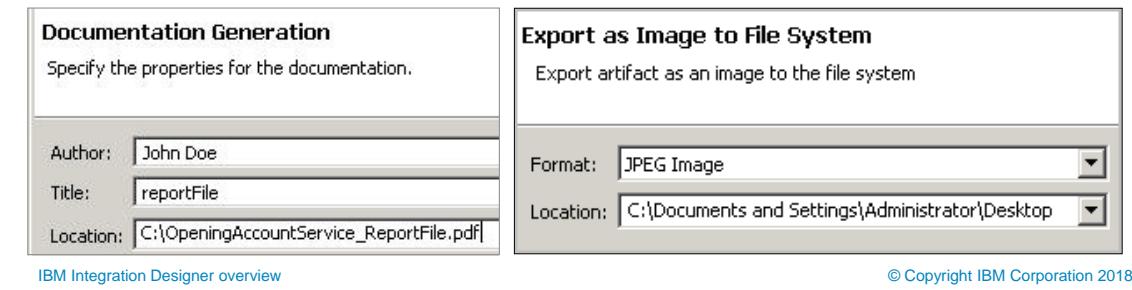
IBM Integration Designer overview

© Copyright IBM Corporation 2018

*Documentation and organization capabilities*

## 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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

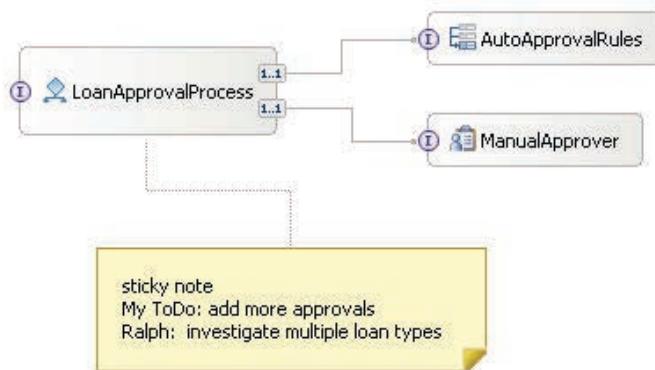
### *Documentation and organization capabilities*

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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

### *Documentation: Sticky 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.

IBM Training IBM

## 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

**Task Tags**

Configure Project Specific Settings...

Strings indicating tasks in Java comments. The entry marked as default will be used in the code templates.

Tag	Priority	New...
FIXME	High	Edit...
<b>TODO (default)</b>	<b>Normal</b>	Remove
URGENT	High	

Tasks Markers Build Activities Properties Problems Servers Asset

33 items

!	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

IBM Integration Designer overview © Copyright IBM Corporation 2018

### Documentation: Java task tags

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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

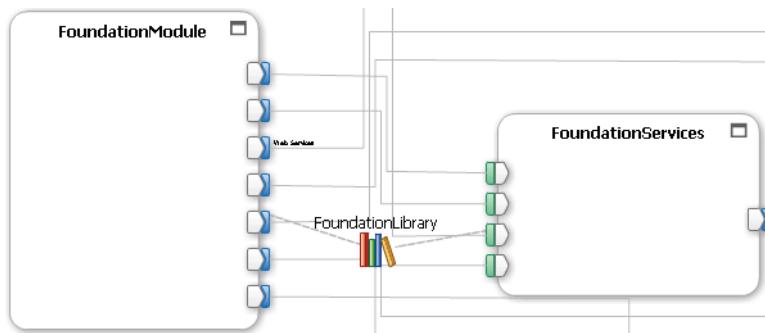
### *Organization: Integration solution*

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



IBM Integration Designer overview

© Copyright IBM Corporation 2018

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

## 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.

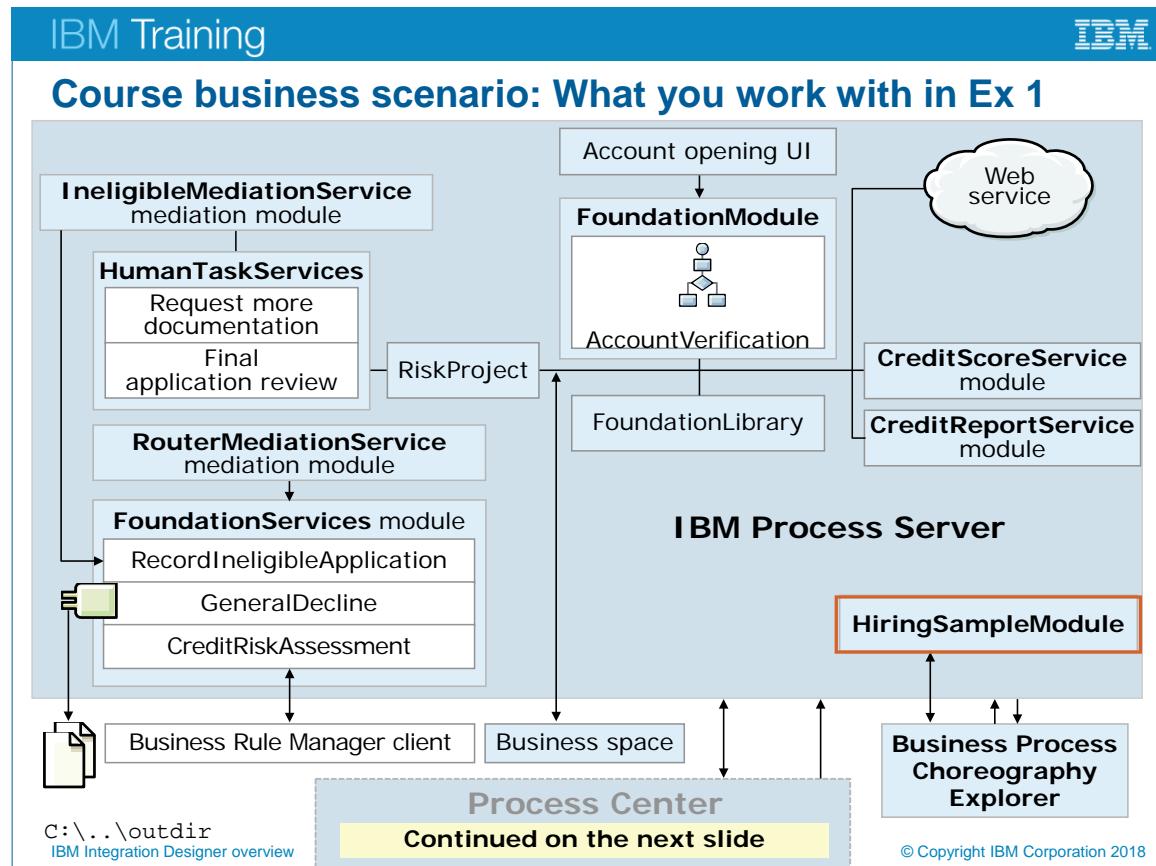
## Checkpoint answers

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.

## Exercise 1: Exploring IBM Integration Designer, part I

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



Course business scenario: What you work with in Ex 1

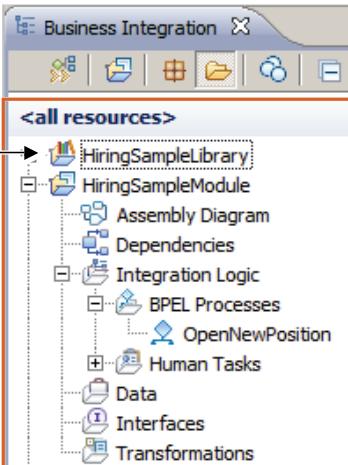
In this exercise, you explore HiringSampleModule.

IBM Training 

## Components that are required for Exercise 1

Prebuilt components that are imported in the lab:

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



The screenshot shows the 'Business Integration' interface with the 'all resources' tree. A red box highlights the 'HiringSampleLibrary' node under 'HiringSampleModule'. The tree structure is as follows:

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

New components that you create in the lab:

**None**

IBM Integration Designer overview © Copyright IBM Corporation 2018

### Components that are required for Exercise 1

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.

IBM Training IBM

## Explore IBM Integration Designer in Exercise 1

The screenshot shows the IBM Integration Designer interface. The title bar reads "Business Integration - HiringSampleModule/OpenNewPosition.bpel - IBM Integration...". The menu bar includes File, Edit, Navigate, Search, Project, Window, and Help. The toolbar has various icons for file operations like Open, Save, Print, and a search icon. A ribbon bar at the top right has "Web" and "Business Inte...". The left sidebar is titled "Business Integratio" and contains a tree view under "<call resources>": "HiringSampleLibrary", "HiringSampleModule" (selected), "Assembly Diagram", "Dependencies", "Integration Logic" (selected), "BPEL Processes" (selected), "OpenNewPosition" (selected), "Human Tasks", and "Notes". The main workspace is titled "OpenNewPosition" and shows a BPEL process diagram. It starts with a "Submit Job Requisition" activity, followed by a decision diamond. One path leads to a "Seek GM Approval" activity, and the other path leads to a "No Approval Needed" activity. The bottom status bar shows "IBM Process Server v8.5.6 at localhost [Stopped]".

### Explore IBM Integration Designer in Exercise 1

You view the HiringSample process in IBM Integration Designer.

## Exercise 1: Exploring IBM Integration Designer, part 1

### Purpose:

This exercise introduces the IBM Integration Designer Eclipse-based development environment. It shows you the components in business integration projects, graphical editors, documentation capabilities, and the Task Flows view. It also explores the preferences, capabilities, product help, and product update features of the tool.

IBM Integration Designer is a member of a family of products that are associated with a service-oriented architecture approach to business process management. It is a complete integration development environment for building applications.

To simplify and accelerate the development of applications, the IBM Integration Designer environment provides a layer of abstraction. It separates the visually presented components that you work with from the underlying implementation. The applications that are created comply with leading industry-wide standards.

Integration developers use IBM Integration Designer to assemble business solutions by using adapters, business processes, code components, and mediation flows. Services that applications invoke can be local or remote, written in various languages, and run on various operating systems. Applications that are created in IBM Integration Designer can be deployed to servers in the deployment stage, and maintained in the IBM Process Center repository. IBM Process Server can run any application that is created in IBM Integration Designer, including applications that contain business processes, mediations, and adapters.

IBM Integration Designer also provides built-in, interactive tools that you can use to learn basic concepts and to create sample applications. You can use the Task Flows view to learn about integration applications. You can also create a series of applications that are designed to acquaint you with mediations, SCA, and BPEL process.

**Important:** There is no IBM Integration Designer 8.6 version. The current version is still at 8.5.7. As a result, the test process server that comes out of the box is also at version 8.5.7. However, that test server has been upgraded to 8.6 level in this environment and you work with the 8.6 release. Furthermore, Process Center 8.6 is also installed and you work with it in a later exercise.

### Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Center, and IBM Integration Designer test environment.

The following information applies to all exercises in this course.

## General Changes

The updates that are made in this version of the course are designed to update the courseware to correspond to IBM Business Process Manager V8.6.

With the release of IBM BPM V8.6, there is a new version of IBM Process Center which is also at V8.6. There is no IBM Integration Designer V8.6. Integration Designer is still at V8.5.7. This means that the test process server that comes with the Integration Designer test environment, is also at V8.5.7.

However, for the hands on exercises in the lab environment, the Integration Designer test process server was upgraded to V8.6. So while Integration Designer itself is at V8.5.7, the test process server running within it is at V8.6.

**Note:** When the test process server is upgraded from V8.5.7 to V8.6, the name of the process server that is displayed by default in Integration Designer is still V8.5.7. To avoid confusion and make it easier to understand, that display name was changed in all workspaces to V8.6, which is in sync with the runtime that is being used. There are prebuilt workspaces created for each exercise and you can access those by using desktop shortcuts. If you create a new workspace, you will see process server 8.5.7 displayed. But if you view the Administration console, you notice that it is V8.6, the upgraded version.

**Starting the User test environment process server (UTE server):** For starting and stopping of Process Sever, use the desktop shortcut. Do not use the start and stop buttons inside Integration Designer to start or stop the process server as sometimes it did not work. Using the shortcut on desktop works each time.

**Migration of projects:** In later exercises, when you first open the workspace, you might be prompted with a window to migrate projects. These projects are already migrated but during tested it was observed that a migration popup was displayed. If you receive that prompt, then accept the defaults, click Next and then Finish. Migration will complete successfully.

**Loading the solution files:** Several files are included which can be loaded as solutions to many of the exercises in this course. These files are in the lab environment under the C:\labfiles\Solutions directory. The names of the solution files match the numbers of the exercises. To use the solution files, import them into an empty workspace

## User IDs and Passwords

The following table contains a list of user ID and password information for this course.

Entry point	User ID	Password
VMware image	Administrator	passw0rd
Windows 2012 R2	Administrator	passw0rd
Process Center Console	pcdeadmin	web1sphere
IBM Process Designer	pcdeadmin	web1sphere
Administration console for IBM BPM	bpmadmin	web1sphere
Administration console for Process Server test environment	admin	web1sphere

### 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 V8.6, 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 and Process Server) are also available in BPM 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 in the course.

### Part 1. Examine IBM Integration Designer capabilities and preferences.

In this portion of the exercise, you start IBM Integration Designer and explore its capabilities in an empty workspace. An Eclipse workspace is a collection of projects and other physical resources that you are currently developing in the workbench. A workspace corresponds to a directory in the file system that contains all of the various source files and resources that are used in a project. In addition, your workspace contains any personalization preferences that you set.

**Note for IBM Business Process Manager on Cloud users:** A version of the IBM Integration Designer that is configured to communicate with your cloud instance can be downloaded from the Process Center tile of IBM Business Process Manager on Cloud.

1. Start IBM Integration Designer, create a workspace, and examine the *Getting Started* tab.

1. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
2. Double-click the shortcut that is labeled **Exercise 1**.
3. Wait few minutes, after which the sample workspace opens. It is an empty workspace.

The Business Integration perspective is the default perspective for developing in Integration Designer. A **Getting Started** tab is displayed in the Business Integration perspective.

The **Getting Started** page includes links to help topics, samples and tutorials, basic concepts, and other resources. The **Getting Started** page is divided into the following sections:

- What do you want to do?
  - Interactive task flows teach you how to work with a process application or create a service.
- What do you want to know?
  - This section teaches you basic concepts, BPEL processes, service integration, and what is new in this version of the product.
- Other Resources
  - This section provides access to developerWorks, IBM Education Assistant, support documents, and frequently asked questions (FAQ).

In addition, you have access to:

- **Help:** Online help documents
- **Samples and Tutorials:** Sample applications and tutorials that demonstrate the use of IBM Integration Designer
- **Change Environment:** Choice of development environments for working with IBM Process Server or WebSphere Data Power

- On the **Getting Started** page, click the **IBM Integration Designer** samples gallery link.



Note the samples available. In the *More samples* section, you can click *Retrieve* to download more samples from the web (when connected).

- Close the **Samples and Tutorials** and **Getting Started** tabs by clicking X.



When all editors are closed, a link is displayed that opens the **Getting Started** tabs for either IBM Integration Designer or IBM Process Center.



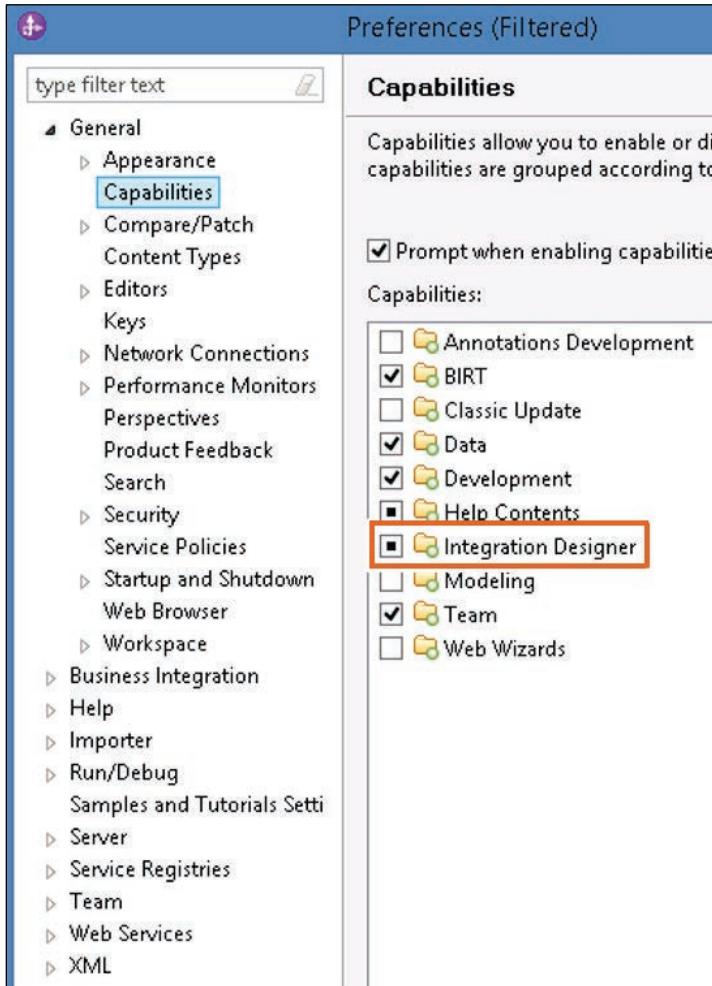
The next time that your workspace opens, the **Getting Started** tab is not displayed. You can open it by clicking **Getting Started with IBM Integration Designer** or by clicking **Help > Getting Started > IBM Integration Designer** from the menu.

- View the capabilities that are currently enabled for the workbench

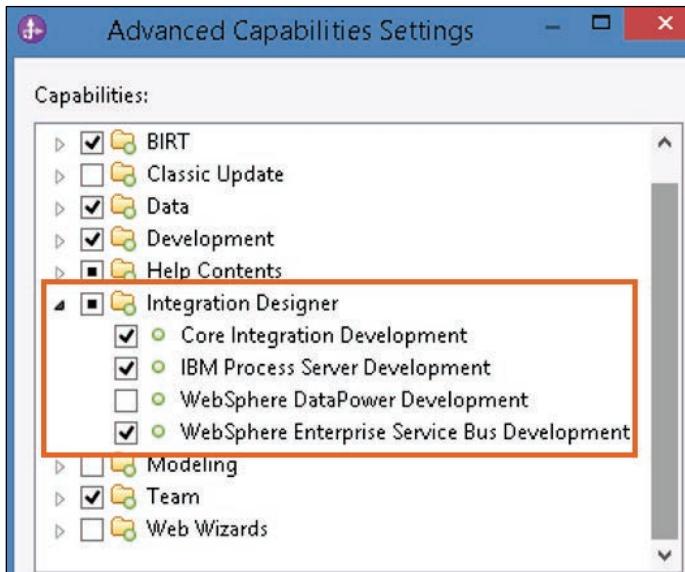
Through Eclipse, IBM Integration Designer provides a filtering function that is known as capabilities. Capabilities are among many workspace preferences you can configure. With capabilities, you can choose to hide or show tools during application development.

- From the menu options, click **Window > Preferences**.
- In the **Preferences** dialog box, expand **General** and select **Capabilities**.

One category is related to the business integration tools.



3. To see a list of the functions in this category, select the **Integration Designer** option (not the check box), and click **Advanced**.
4. In the Advanced Capabilities Settings dialog box, expand **Integration Designer**.



This Integration Designer category has four functions:

**Core integration development:** Provides tools, functions, and related documentation for development of bottom-up integration of applications and services to create:

- Libraries
- Business objects
- Interfaces

**IBM Process Server development:** Includes advanced tools for top-down development of integrated business solutions to create and test components, such as:

- Modules and module assemblies
- Data maps
- Customized visual snippets
- Processes
- Business rules
- Human tasks

**WebSphere DataPower development:** Includes tools, functions, and related documentation to create and test artifacts that can be deployed to WebSphere DataPower Appliances, such as:

- DataPower libraries
- Business objects
- Interfaces
- XML maps

By default, this environment is not enabled.

**WebSphere Enterprise Service Bus development:** Includes core enterprise service bus integration development tools and also provides tools, functions, and related documentation to create and test:

- Mediation modules and assemblies
- Mediation flows

Note: The IBM WebSphere Enterprise Service Bus product is discontinued, and all its capabilities are incorporated into the IBM Integration Bus product.

Customers who purchase IBM Business Process Manager can deploy mediation modules to the IBM Process Server runtime environment. They are not required to use a separate integration bus (enterprise service bus) environment.

5. Click **Cancel** to return to the **Preferences** dialog box.
3. Explore Preferences.

In addition to capabilities, many other preferences can be configured. For more information about each of the preferences, see the online help. In this portion of the exercise, you examine a few of the more commonly used preferences settings.

To examine the IBM Integration Designer preferences:

1. In the **Preferences** dialog box, in the navigation pane, expand **Business Integration**.

Note the available preferences, including the settings for the Integration Designer assembly editors.

2. In the navigation pane, expand the **Server** section and select **Runtime Environments**.

Note the available runtime environments and the ability to add and remove them.

**Important:** As mentioned earlier, there is no IBM Integration Designer 8.6 version. The current version is still at 8.5.7. As a result, the test process server that comes out of the box is also at version 8.5.7. However, that test server has been upgraded to 8.6 level in this environment and that is why you see IBM Process Server v8.6 listed.

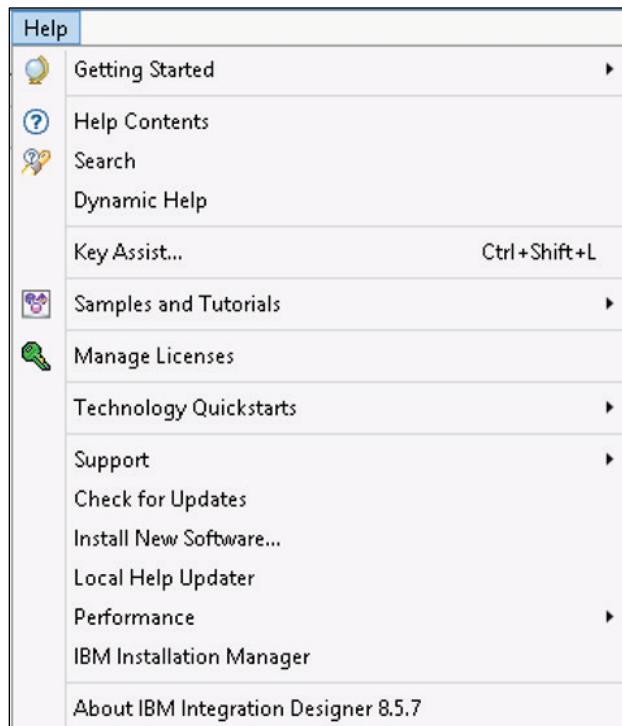
Server Runtime Environments	
Add, remove, or edit server runtime environments.	
Server runtime environments:	
Name	Type
IBM Process Center v8.5.7	IBM Process Center v8.5.7
IBM Process Server v8.6	IBM Process Server v8.5.7

3. Click **Cancel** to close the **Preferences** dialog box.

## Part 2. Use the help menu to search for assistance.

The help menu is a great resource for finding more information about all aspects of IBM Integration Designer. In this part of the exercise, you explore the options available from the help menu.

### 1. Click Help options menu



The help menu in IBM Integration Designer contains the following options (you are encouraged to open each of these resources and explore as you read):

- **Getting Started:** This option opens the Welcome pages that are associated with either IBM Integration Designer or Process Center for Integration Designer. You can view either of these pages when a workspace is first started. Each page contains links to Samples and Tutorials, Task Flows, Reference Materials, and Patterns.
- **Help Contents:** This option opens the IBM Knowledge Center documentation on IBM Integration Designer on the local computer.
- **Search:** This option starts a help view that can be used to search multiple information sources concurrently. You can search the web, the tutorials gallery, eclipse.org, IBM developerWorks, and the IBM Knowledge Center documentation on WebSphere Application Server.
- **Dynamic Help:** This option provides information relevant to the current task you are working on in the workbench.

- **Key Assist:** This option provides you with a list of relevant keyboard shortcuts. Press Esc to close the list.
- **Samples and Tutorials:** This option provides links to the IBM Integration Designer application samples available from the Getting Started page.
- **Manage Licenses:** This option provides status and update capacity for the current license.
- **Technology Quickstarts:** This option provides links to tutorials. Some of these tutorials are as follows: tutorials that are viewed like movies, play-and-learn tutorials that guide you through a simulation, do-and-learn tutorials for hands-on learning, and tours that contain multimedia displays that teach you conceptual information.
- **Support:** This option provides you with several useful links to external support resources such as the IBM user forums, IBM developerWorks, the IBM Education Assistant, IBM Technical Support, and the IBM Support Portal.
- **Check for Updates, Install New Software, Local Help Updater, and IBM Installation Manager:** Each of these options provides access to the IBM Installation Manager, which walks you through installing new or updating existing functions.
- **Performance:** This option provides features such as immediately reducing the amount of memory that is used, generating diagnostic information, and enabling system profiling.
- **About IBM Integration Designer 8.5.7:** This option displays the version information and information about installed features and plug-ins.

Note: In addition to the resources provided directly in IBM Integration Designer, you can also download and install the IBM Support Assistant from [www.ibm.com](http://www.ibm.com). You can use the support assistant to search for an answer to your question or problem in many different locations at the same time. You can also get speedy access to critical product information or run free troubleshooting and diagnostic tools on a troublesome application. You can shorten the amount of time it takes to resolve an issue with automated data gathering and submission tools.

## Part 3. Use the Installation Manager to find product updates.

The IBM Installation Manager utility is used to install IBM Integration Designer product updates (and other Eclipse-based IBM product updates). The IBM Update Installer is used to install the IBM Process Server test environment. For more information about updating the test environment, see the product documentation.

Warning: It is important that no software updates are made in this exercise. The purpose of this section is to demonstrate how the updates are done and not to make those updates to this image. Making any software updates can cause unpredictable image behavior and lab results.

### 1. To check for IBM Integration Designer product updates:

1. Start the IBM Installation Manager utility by clicking **Help > IBM Installation Manager**.
2. Click the **Update** icon.

You might receive errors such as: The following repositories are not connected.



Important: These steps are designed to show you the update process. You do not update IBM Integration Designer in this course. Updating your software can produce unexpected results.

3. If you receive a warning message that the resources are not connected, click **OK**. If no errors are displayed, then at the Update Packages dialog box, click **Next**.

4. If a dialog box is displayed prompting you to enter credentials, then click **Cancel** to close the window. The dialog box might open more than one time. If it does, then click **Cancel** again to close the Password Required dialog box.



5. Click **Cancel** to close the Update Packages window.  
The purpose of this exercise is to explore the update feature, not to update the WebSphere tools.
6. Click **File > Exit** to close Installation Manager.

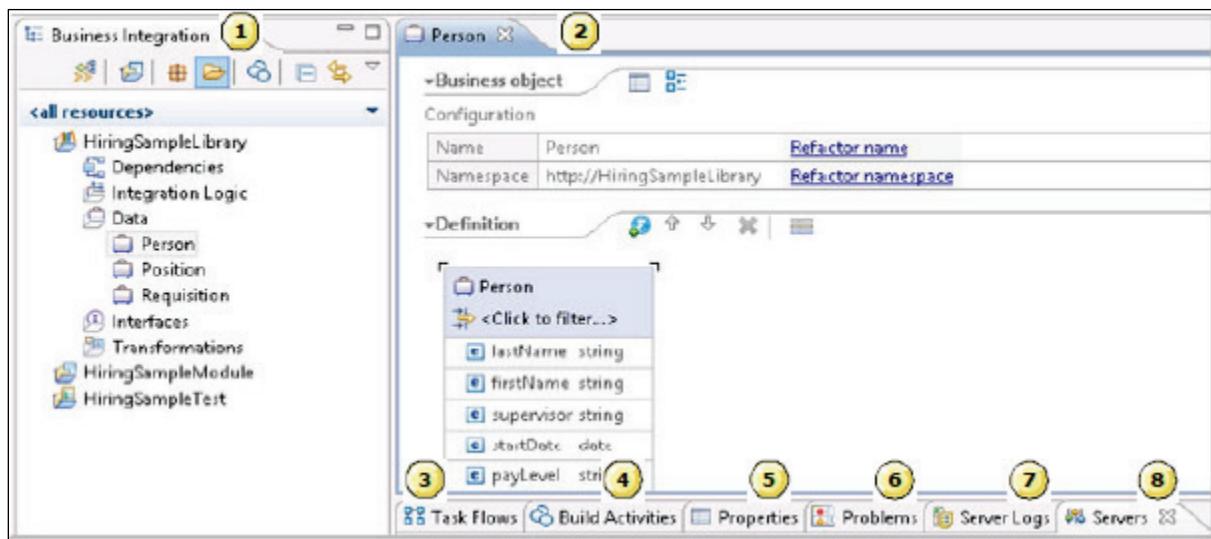
#### Part 4. Browse the Business Integration perspective and views.

In this portion of the exercise, you browse the different views and perspectives in IBM Integration Designer. A perspective is a specific arrangement of views in the workbench. The user's active perspective controls which views are shown on the workbench page and their positions and sizes. A view is typically used to browse a hierarchy of information, open an editor, or display properties for a selected object. Users can switch between perspectives as they change tasks.

Each perspective provides a set of functions that are aimed at accomplishing a specific type of task or working with specific types of resources. Therefore, opening a different perspective opens a different group of views. For example, the Java perspective combines views that you would commonly use while editing Java source files. The Debug perspective contains the views that you would use while debugging applications.

## 1. Examine the Business Integration perspective.

By default, when you start IBM Integration Designer, the Business Integration perspective opens. Most of your integration development is done in the Business Integration perspective. Notice that since you just opened the workspace, it is empty. The Business Integration perspective has the following views and panes (you are encouraged to open each of these components and explore as you read).



**Business Integration view (1):** This view provides a logical view of the key resources in each module, mediation module, and library. Non-SCA projects are displayed in the Business Integration view. You can use this view to edit projects without switching perspectives.

**Editor pane (2):** When you open a resource from the Business Integration view with an editor, the resource is displayed in the editor pane.

**Task Flows view (3):** Task flows provide a way to learn related concepts and tasks in an interactive manner. A task flow puts together all of the different tasks that are required to achieve a goal. Task flows are presented as a series of grouped tasks, where each task is a link. When you click the links, wizards and editors start along with information windows so that you are learning while you work on tasks.

**Build Activities view (4):** This view helps you manage builds. You can view the build and server status of business integration projects.

**Properties view (5):** In this view, you can see information about a selected artifact. When using the Properties view with editors, you are able to modify properties of elements you select.

**Problems view (6):** This view helps you debug errors by providing message text. You can see more help for the problem message by selecting a message and pressing F1.

**Server Logs view (7):** This 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.

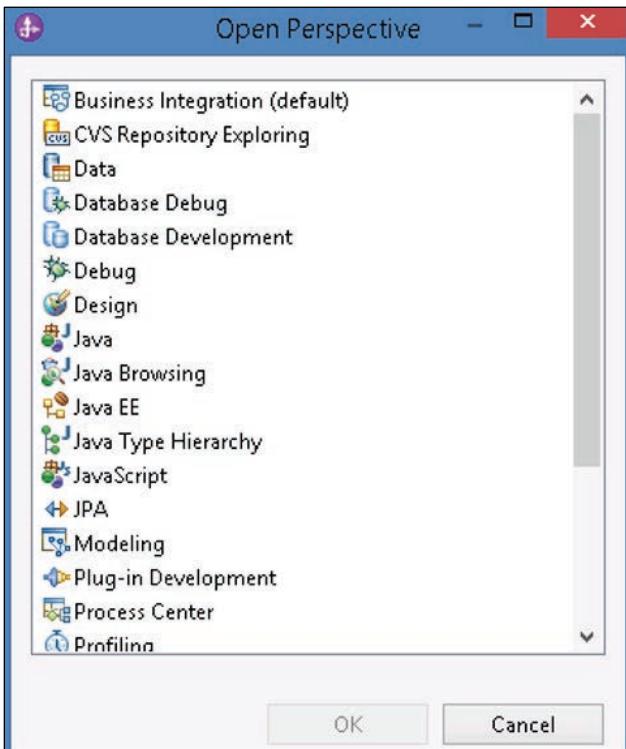
**Servers view (8):** Use the Servers view to create test servers (local or remote). Use the view for testing and deploying business integration applications and for viewing the status of deployed modules.

## 2. Browse other perspectives.

In addition to the Business Integration perspective, several other perspectives are available in IBM Integration Designer. These additional perspectives are designed to accommodate different development tasks.

To switch perspectives:

1. Click **Window > Open Perspective > Other** to see a list of other perspectives.



2. Choose the perspective that you want to open, and click **OK**.

The open perspectives are listed at the upper-right corner of the workbench window.

You can also open a new perspective by clicking the **Open Perspective** icon

3. To switch perspectives, click **Window > Navigation > Next Perspective** or click the right-pointing double arrows **>>** in the upper-right corner of the workbench.

It might be necessary to have several perspectives open to work on these actions. You cannot use the double arrow method in the Process Center perspective.

4. To close a perspective, click **Window > Close Perspective**, or right-click the perspective in the upper-right corner of the workbench window and click **Close**.
3. Customizing a perspective with views.

Perspectives offer predefined combinations of views. To open a view that is not included in the current perspective, click **Window > Show View** from the menu bar. In addition to adding a view to a perspective, you can maximize views, minimize views, move views, dock views, close views, open views, and create fast views.

If you want to minimize and maximize views, you can do so in several ways:

- Double-click the title bar to maximize a view so that it fills the workbench window. Double-click the view again to restore it to its original size.
- Use the minimize and maximize icons that are attached to the view (click the restore icon after maximizing or minimizing it). A minimized view is visible as a toolbar on the perspective. Use restore or maximize  to reestablish the view.
- Right-click the title bar of the view and click **Minimize** or **Maximize** from the menu. After maximizing the window, right-click the title bar and click **Restore** to restore the window to its original size.

Hint: If you close IBM Integration Designer, the last used perspective is preserved. In other words, if you add and format views in a perspective, the new layout is preserved when you close and reopen Integration Designer. However, if you change perspectives, your customized layout is lost. To save your new layout as a customized perspective, click **Window > Save Perspective As**.

A perspective can be customized to fit personal needs and saved. A simple relocation of a view might be all that you need for it to be more productive. You can click the title bar at the top of a view and drag it to a new location. As you move the view around the workbench, the mouse pointer changes to a drop cursor. The drop cursor indicates where the view would be docked in case you release the left mouse button.

To change the location of a view:

1. Click the title bar of the **Build Activities** view, hold the left mouse button down, and drag the view. Drag the view to the left, right, top, or bottom border of another view. You can also drag the view outside of the workbench area to turn it into a detached view.

2. When the view is in the location you want, release the left mouse button.
3. (Optional) If you want to save your changes, click **Window > Save Perspective As** from the main menu bar.  
Note: You can also use the menu to move a view by right-clicking anywhere in the title bar of the view. This menu provides shortcut options for turning a view into either a “fast” or a “detached” view.

#### 4. Explore Fast and detached views.

A fast view provides a quick means to display and hide a frequently used view. When a fast view is created, its icon is placed in the bottom, fast view toolbar. Click that icon to display or hide the view. Fast views work like other views except that they do not take up space in your workbench window.

To create a fast view:

1. Click **Help > Search** from the menu options.

It might take a few minutes for indexing to complete, and its status is displayed at the lower right of the Help view.

2. Right-click the title bar of the **Help** window and click **Fast View** from the menu.

You can also drag an open view to the fast view toolbar in the lower-left corner of the status bar.

3. To access the fast view, click the  icon in the lower-left corner of the status bar.
4. To hide the fast view again, click the minimize icon in the title bar of the view, or click outside the fast view window. If you click the **X** to close the view, the view is removed from the fast view toolbar.

Note: Any of the views in the current perspective can be displayed as fast views by clicking the Show View as fast view icon. The fast view toolbar can be docked on either side of the workbench window.

When a view is detached, the view is free to “float” in the development environment, much like a separate window. To close a detached view, use the **X** in the upper-right corner of the detached view.

#### Part 5. Examine the modules and libraries of a business integration project.

In this portion of the exercise, you browse a business integration project. The first step in developing an application in IBM Integration Designer is to set up one or more projects to hold your resources. Projects are the largest structural units in your workspace.

## 1. Explore types of projects

A project is an organized collection of folders or packages. Projects are used for building, version management, sharing, and organizing resources that are related to a single work effort. The projects that you work with most in IBM Integration Designer are modules, libraries, and mediation modules, but several types of projects are available:

- **Integration solution:** An integration solution is a nondeployable project that is used solely to reference other projects in the workspace, such as modules, mediation modules, libraries, component test projects, Java projects, and simple projects. Integration solutions help you organize groups of related projects and more easily do common actions on the projects, such as sharing them in a team environment. When you create an integration solution, an integration solution diagram is automatically generated. You can open the integration solution diagram in the integration solution editor and view the relationships between the related projects that are referenced in the integration solution.
- **Modules:** Modules provide the business services for your application, which are modeled as Service Component Architecture (SCA) components that are wired together in a module assembly. Modules are the basic units of deployment to the IBM Process Server runtime environment.
- **Mediation modules:** Mediation modules provide mediation service applications, which intercept and modify messages that are passed between existing services (providers) and clients (requesters) that use those services. Mediation modules can be deployed on IBM Integration Bus (formerly WebSphere Enterprise Service Bus) or IBM Process Server. You learn more about mediation modules later in this course.
- **Libraries:** Often interfaces, data types, transformations, business calendars, and web service ports must be shared so that resources in more than one module can use them. The library is a project that is used to store these resources. Libraries are associated with modules through dependencies.
- **Component test project:** To test modules and applications in the workspace, you might want to create test suites that contain sample data. A component test project is a project that is kept separate from your working, production data. It contains testing tools such as test suites, execution traces, server configurations, emulators, and collections of sample data called “data pools”.
- **Modules and library dependencies:** When developing and deploying integration applications, it might be necessary to declare dependencies for your modules, mediation modules, and libraries. Use the dependency editor to manage these required resources.

- **Namespaces:** A namespace is a logical container in which all names are unique. A name can be displayed in multiple namespaces but cannot be displayed twice in the same namespace.
- **Other projects:** You can also bring Java projects into your application. In some cases, you might choose to do bottom-up development by designing Java resources and then importing them into your IBM Integration Designer application. If you have Java code that is used in a business integration module or mediation module, you can create a Java project. You can add that Java project as a dependency in the module that uses the code.

Note: You can easily tell mediation modules from integration modules in the Business Integration view by examining the icons. The icon for a mediation module is a folder with an envelope. The icon for an integration module is a folder with a document.

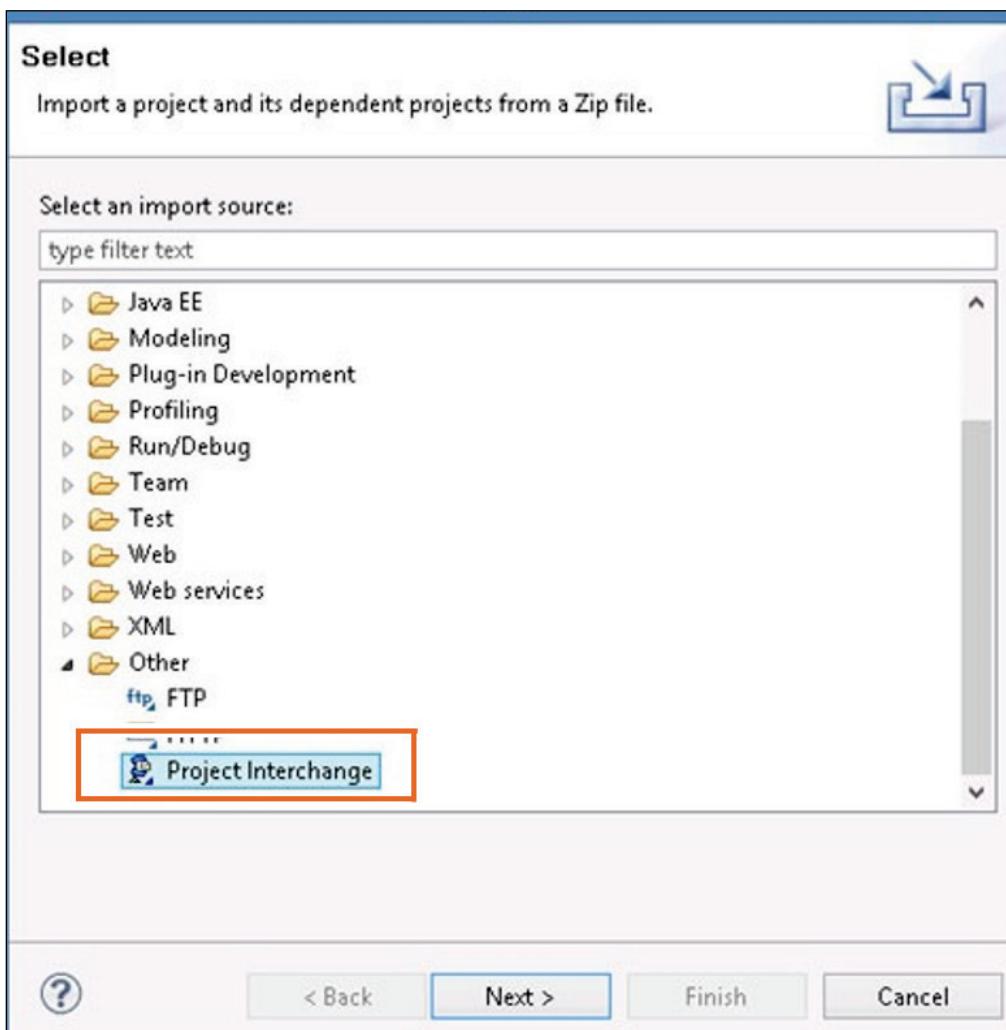
## 2. Explore the structure of business integration modules.

The Business Integration perspective has a Business Integration view, which provides a logical view of the key resources in each module, mediation module, and library. Within each project, the resources are categorized according to type. Logical resources that are displayed in the navigation tree do not necessarily have a one-to-one mapping to files. For example, a business process that is displayed in the integration logic section might comprise several different physical files. You can also use the Business Integration view to browse through Java and Java EE resources.

To browse the resources in a business integration module:

1. If you opened another perspective in the previous section, switch to the **Business Integration** perspective.
2. Right-click within the Business Integration view and click **Import**.

3. In the Import window, select Other > Project Interchange.



4. Click **Next**.
5. In the Import Project Interchange Contents dialog box, select **Browse** next to the **From zip file** field.
6. Browse to the C:\labfiles\Support Files\Ex1 directory and select **HiringSample.zip**.

7. Click **Open**.

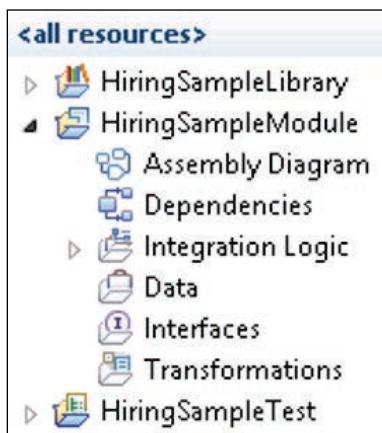
The contents of the hiring sample project interchange file are displayed.

8. Click **Finish**.

Allow a few moments for Integration Designer to import the project file. After the file is imported, the workspace is built. Wait until the status reaches 100% at the lower right of the Integration Designer. When the workspace is built, the status progress disappears.

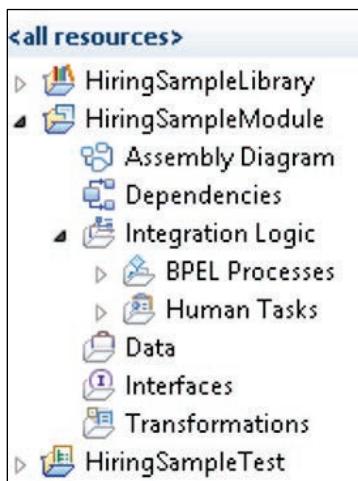
**Important:** Sometimes during testing, it was observed that even though the project is already migrated, a migration window appears. If you see a migration window displayed, accept the defaults and click **Next** to continue. Migration completes and you can then continue with the steps below.

9. Click the arrow to expand the contents of **HiringSampleModule**.

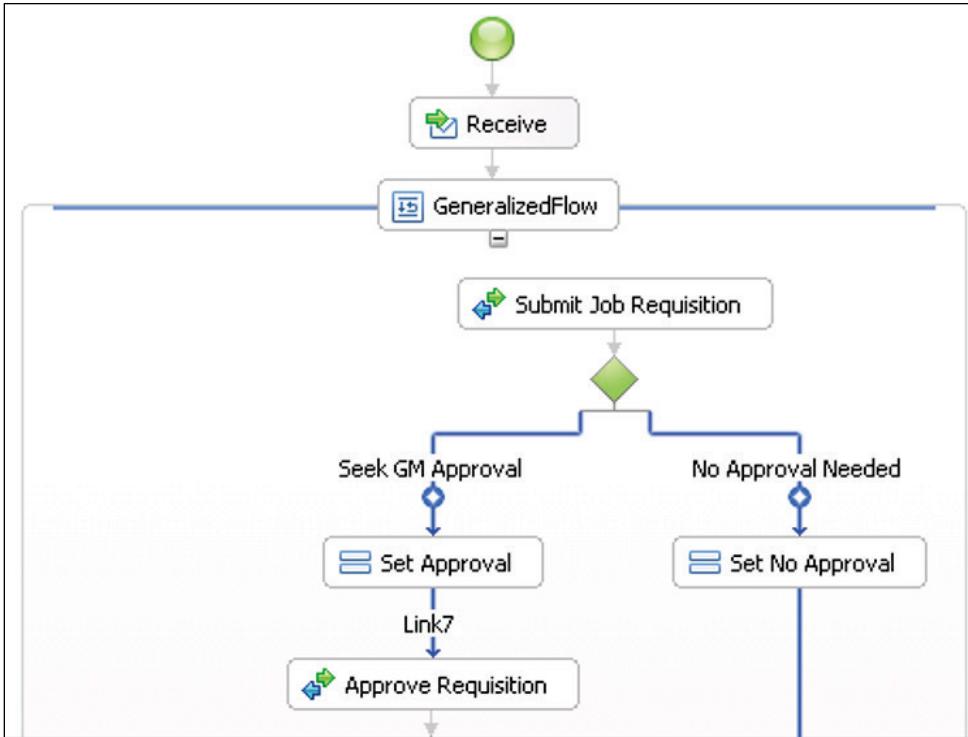


Each business integration module consists of the following resources:

- **Assembly Diagram:** The building blocks of a business solution are Service Component Architecture (SCA) components that are wired together to form modules that can be deployed to IBM Process Server. You can use the IBM Integration Designer assembly editor to build applications by assembling the SCA components on an assembly diagram.
  - **Dependencies:** Double-click **Dependencies** to open the dependencies editor to see and manage the dependencies on other modules or libraries.
  - **Integration Logic:** The Integration Logic section contains all of the artifacts that do specific business tasks. Business processes, human tasks, and mediation flows are some examples of integration logic implementations.
  - **Data:** The Data section contains all of the data representations. For example, it contains business objects and user-defined simple types.
  - **Interfaces:** The Interfaces section contains WSDL interfaces that define the operations that a service exposes and makes available to callers.
  - **Transformations:** The Transformations section contains data transformations such as data maps and Extensible Stylesheet Language (XSL) transformations.
10. Expand the **Integration Logic** section in the **HiringSampleModule** and examine the artifacts.

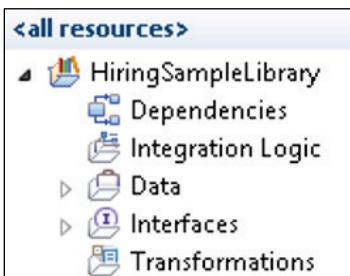


11. The **Integration Logic** section contains **BPEL Processes** and **Human Tasks**. Expand **BPEL Processes** and double-click **OpenNewPosition**.  
The BPEL process opens in the editor.



12. Take some time to examine the steps of this business process. Can you determine the intended activities along each step of the process? Do not change the process.
13. Close the open tabs when you are done.
3. Examine the artifacts in the **HiringSampleLibrary**..
- Each of the artifacts you create in a library can also be included in a module. For example, you can create a business object in the Data section of a module, and you can create a business object in the Data section of a library. However, artifacts that are created in a module cannot be shared among projects, while artifacts created in a library can be shared among projects.
1. In the Business Integration view, expand **HiringSampleLibrary**.

2. Note the artifacts that can be included in the library:



- **Dependencies:** Double-click **Dependencies** to open the dependencies editor to see and manage the dependencies on other modules or libraries.
- **Integration Logic:** Integration Logic contains all of the artifacts that do specific business tasks. Business processes, human tasks, and mediation flows are some examples of integration logic implementations. When defined in a library, these artifacts can be reused in other projects.
- **Data:** The Data section contains all of the data representations. For example, it contains business objects and user-defined simple types. You can define these data representations in a library to use them in other applications.
- **Interfaces:** The Interfaces section contains WSDL interfaces that define the operations that a service exposes and makes available to callers. Interfaces in a library can be used in multiple modules by several components.
- **Transformations:** The Transformations section contains data transformations such as data maps and XSL transformations. Data transformations that are defined in a library are available for use by multiple applications.

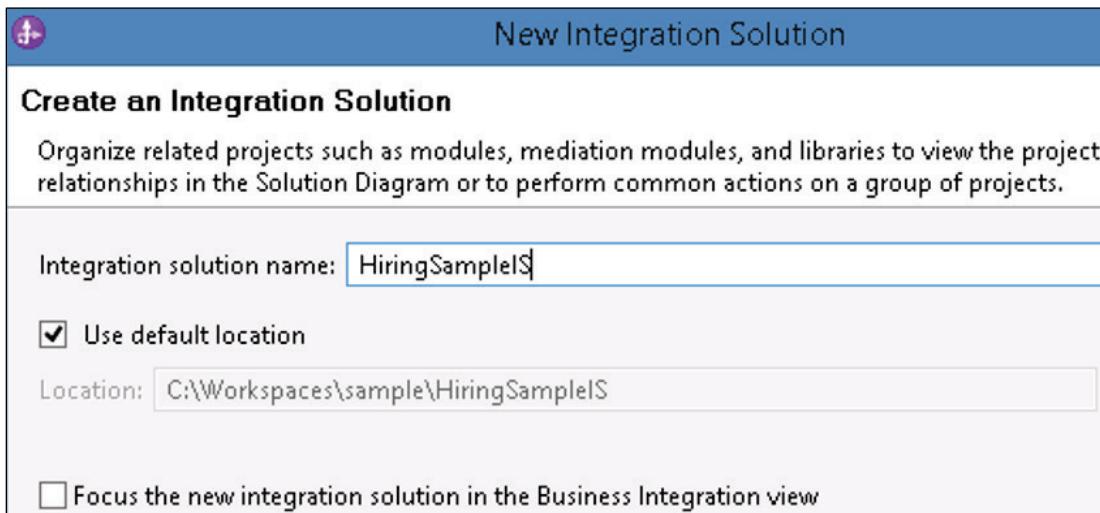
4. Add an integration solution.

An integration solution is a nondeployable project that is used solely to reference other projects in the workspace, such as modules, mediation modules, libraries, component test projects, Java projects, and simple projects. You use the loan application to create an integration solution. This exercise is intended to introduce you to integration solutions. In a later exercise, you have the opportunity to explore integration solutions in detail.

To create an integration solution:

1. Click File > New > Integration Solution.

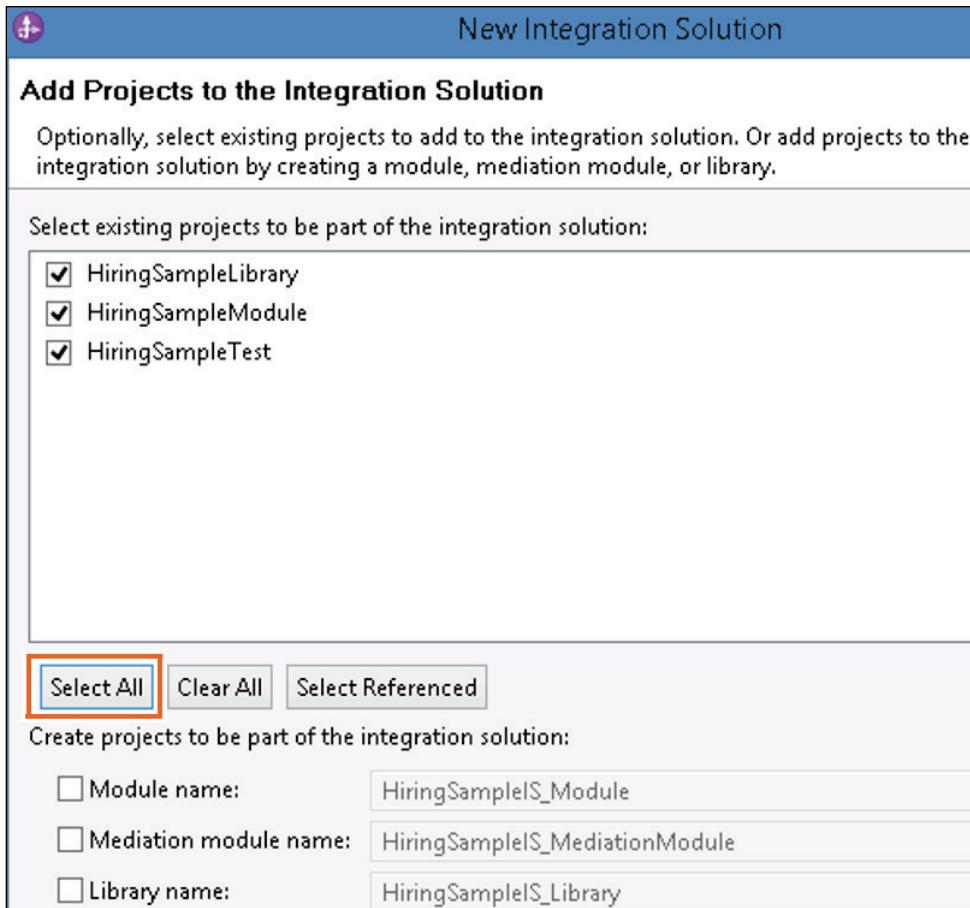
2. In the **Integration solution name** field, enter: HiringSampleIS



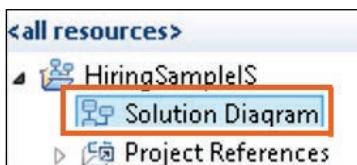
3. Ensure that the **Focus the new integration solution in the Business Integration view** check box is not selected and click **Next**.

If you have multiple integration solutions, this option hides all of them except the one you are creating.

4. Click **Select All** to add all of the existing projects to the integration solution. If the Select All button is not displayed then maximize the window to view it.



5. Click **Finish**.
6. Expand **HiringSampleIS** and double-click **Solution Diagram**.



7. In the solution diagram editor, click the **Show All Projects** icon.  
This action changes the view from the default: Only Show Modules.  
The solution diagram is a graphical representation of the artifacts that were selected when you created the integration solution.



As time permits, feel free to experiment with the other options in the integration solution feature.

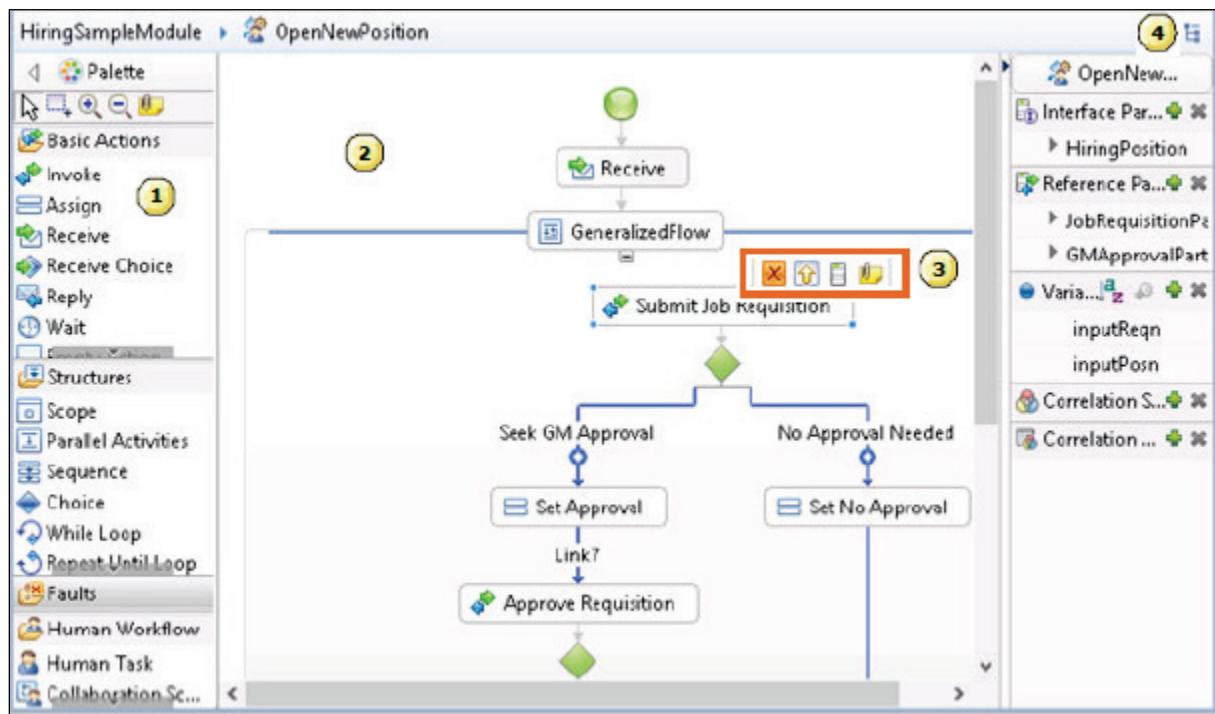
8. Close the **HiringSampleIS - Solution Diagram** tab.
9. Save your changes.

## Part 6. Browse the IBM Integration Designer graphical editors.

In this portion of the exercise, you browse two of the more commonly used IBM Integration Designer editors. An editor is typically used to edit or browse a document or input object. Modifications that are made in an editor follow an open-save-close model, much like an external file system editor. The platform text editor and Java editor are examples of workbench editors. Each editor has the same basic structure, but the specific areas that are available differ depending upon the object that you are editing.

1. Open **OpenNewPosition** in the business process (BPEL) editor.
  1. In the Business Integration view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.
  2. Double-click **OpenNewPosition** to open the business process (BPEL) editor, if not already open.

The business process editor is divided into the following components. Feel free to explore as you read.



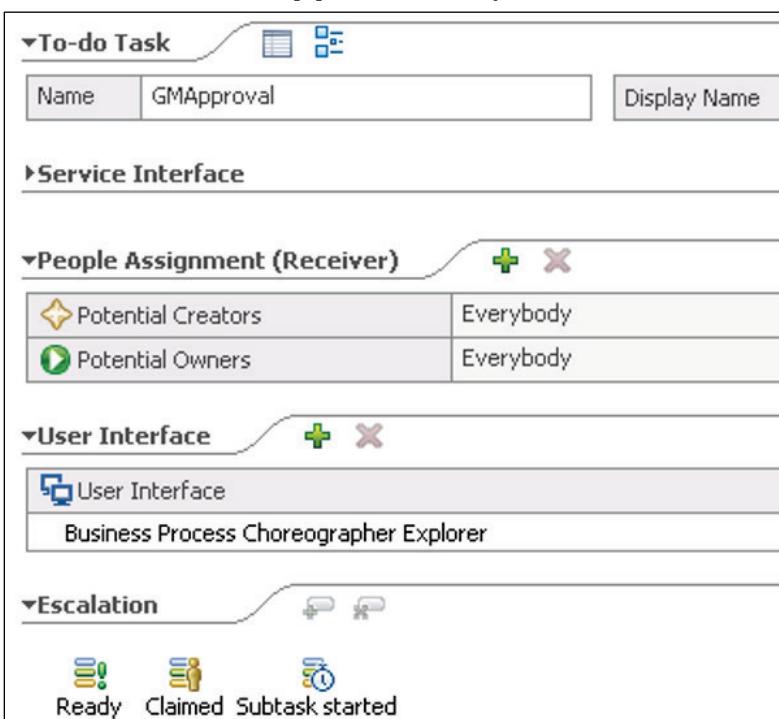
**Palette (1):** The palette is the area to the left of the canvas that houses the activities you drag onto the canvas to build your process. The icons are organized under several different headings that act as toggles. Click the heading, and the icons remain hidden until you click that heading again. The complexity of the palette is reduced and hides those icons that you rarely use. You can gather frequently used items into a favorites folder by right-clicking the icon and clicking **Add to Favorites**. (If the favorites folder does not exist, it is created.) To hide the text labels when you are familiar with the icons, right-click the palette, and toggle the **Show Labels** setting. To increase the size of the icons, right-click the palette, and toggle the **Use Large Icons** setting.

**Canvas (2):** The canvas is the white, empty area in the middle of the editor that you use to compose your business process. When you drag an activity from the palette onto the canvas, the icon beside your cursor has a plus (+) symbol, and you can decide where you want to drop the activity. When the cursor becomes a crossed-out circle, continue moving the cursor until it becomes a plus sign again.

**Action bar (3):** The action bar is a miniature dialog box that is displayed next to certain activities when you select them, and it contains a series of one or more icons that are relevant to that activity.

**Tray (4):** The tray displays the partners, variables, correlation sets, and correlation properties that are associated with your process (each of these items is defined later in the course). To see the interfaces and operations that are associated with the partners, click the small gray arrow beside the name of the partner. To create an item, click the green + icon; or to remove one, highlight it and click the red X icon. Click the gray horizontal arrow to collapse or expand the tray.

3. After examining the components, close the process editor by clicking X on the title bar. If you changed any components, do not save them.
2. Open the GMApproval human task editor.
  1. In the Business Integration view, expand HiringSampleModule > Integration Logic > Human Tasks.
  2. Double-click **GMApproval** to open the human task editor.



3. View the options in the human task editor. After examining the task, close the editor.

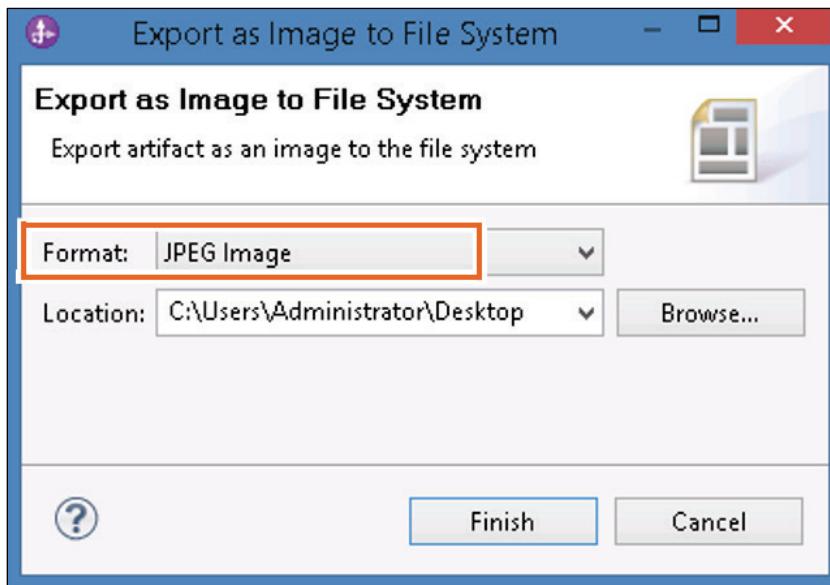
## Part 7. Use the IBM Integration Designer documentation capabilities.

In this portion of the exercise, you examine some of the project documentation features in IBM Integration Designer.

### 1. Export a business process image

If you want to share a business process with others who do not have IBM Integration Designer, you can export the process as an image for them to review. You can also use this feature to create project documentation for later review or for governance. To export a business process as an image file:

1. In the Business Integration view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.
2. Right-click **OpenNewPosition** and click **Export as Image** from the menu.
3. In the **Export as Image to File System** dialog box, enter the following information:
  - For **Format**, select **JPEG Image** from the list. Other supported files include PNG Image, SVG Document, and TIFF Image.
  - For **Location**, click **Browse**, go to your desktop, and click **OK**.

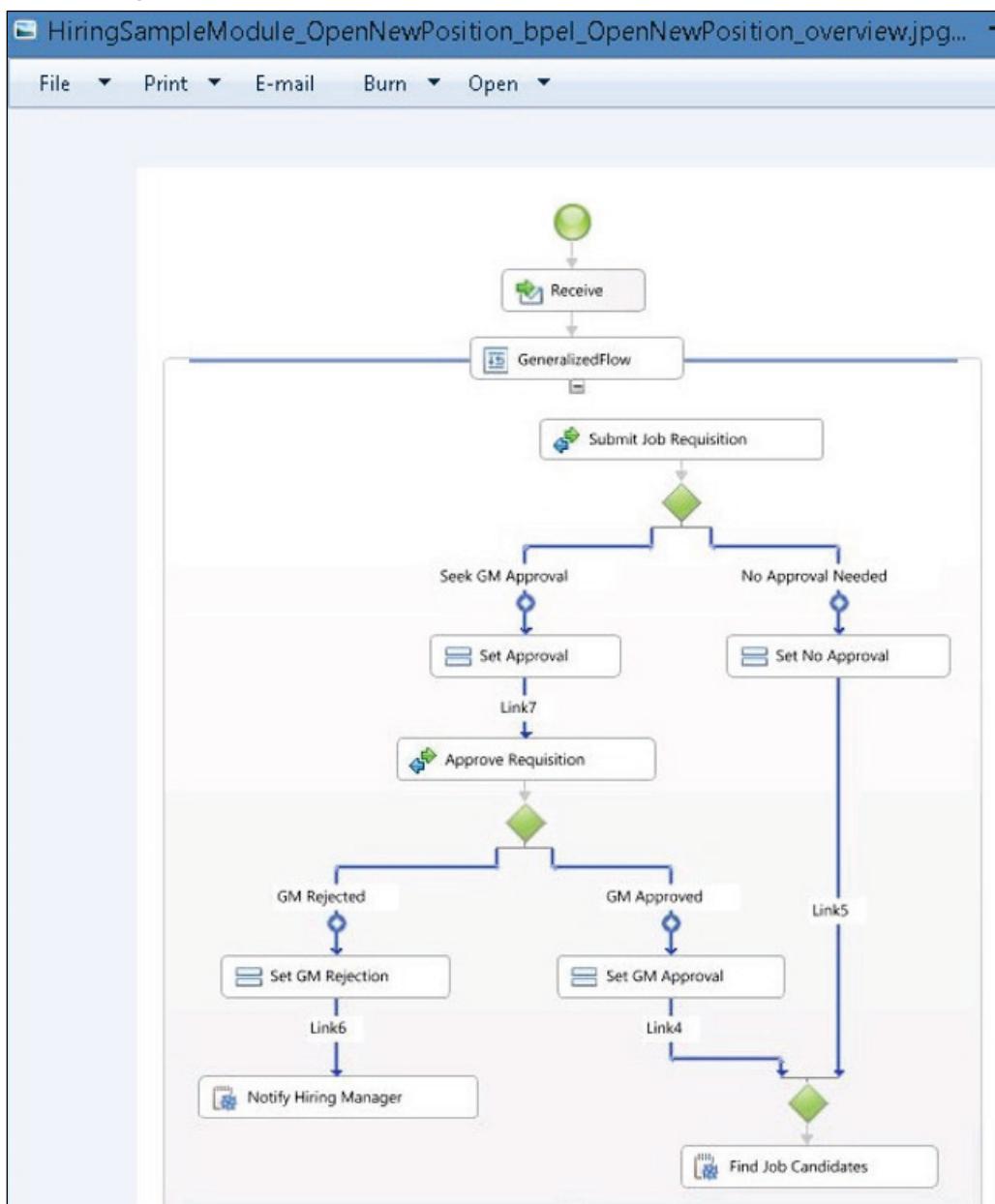


4. Accept the remaining default options and click **Finish**.
5. When you receive the **Export as Image Result** dialog box, the message **The following images have been exported successfully** is displayed.

The following images have been exported successfully:			
Image	Logical Artifact	Project	File
C:\Users\Administrator\Desktop\HiringSampleModu...	OpenNewPositi...	HiringSampleMod...	OpenNewPosition.bpel

6. Click **OK**.
7. Locate the newly saved image on the desktop and double-click **HiringSampleModule\_OpenNewPosition\_bpel\_OpenNewPosition\_overview.jpg**.

The image file loads in the Windows Photo Viewer.



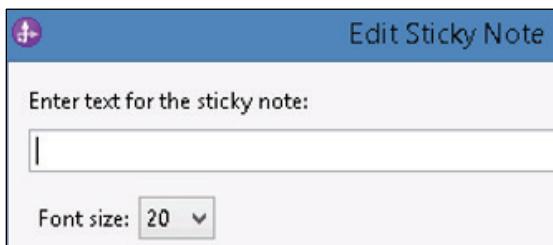
8. Close the Windows Photo Viewer and maximize IBM Integration Designer.

## 2. Create sticky note documentation

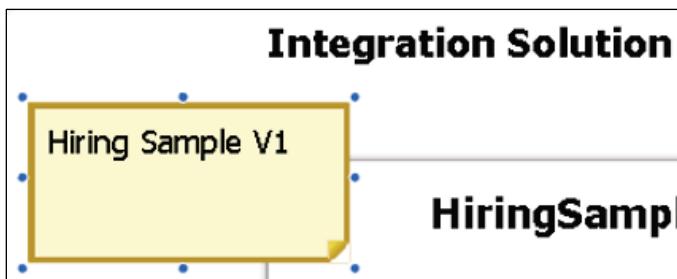
In addition to exporting project components as images, you can also create embedded documentation by adding sticky notes. Sticky notes can be used to create reminders for developers or to create task lists by using Javadoc task tags.

To create sticky note documentation in your integration solution diagram:

1. In the Business Integration view, expand **HiringSampleIS** and double-click **Solution Diagram**.
2. Click the **Create sticky note**  icon:
3. Click the diagram in the location where you want to place the note.
4. To open the note editor, right-click the note and click **Edit Sticky Note** from the menu.



5. In the **Enter text for the sticky note** field, type: `Hiring Sample V1`
6. Click **OK**.



You can resize the sticky note by selecting it and dragging one of the handles.

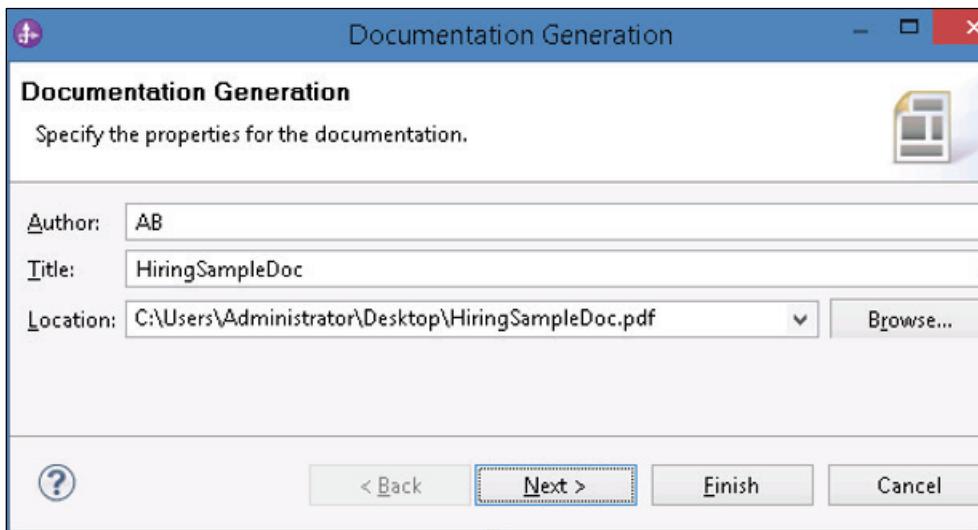
7. Save your changes and close the solution diagram.

## 3. Generating documentation reports

In addition to images and sticky notes, you can generate project documentation reports. You can generate documentation for the artifacts in your modules, mediation modules, and libraries. You can customize the contents of the documentation by selecting multiple resources from one or more modules and libraries so that all of the information is written to a single report, in a PDF file. You can also specify the layout and fonts for the output.

To generate a documentation report:

1. In the Business Integration view, in the Projects window, right-click **HiringSampleModule** and click **Generate Documentation** from the menu.
2. In the **Documentation Generation** dialog box, enter the following information:
  - In the **Author** field, type your initials.
  - In the **Title** field, type: `HiringSampleDoc`
  - In the **Location** field, click **Browse**, browse to your desktop, type `HiringSampleDoc` in the **File name** field and click **Save**.



3. Click **Next**.
4. Ensure that Generate documentation from the selected resource is selected and click **Next**.
5. Accept the default **Paper size** and **Orientation** settings, note the other options on the **Layout Settings** dialog box, and click **Finish**.
6. When the report is generated, click **Yes** when you are prompted to open the generated report.  
Adobe Reader is installed in your lab environment, so you can view the generated report.
7. After reviewing the report, close Adobe Reader.

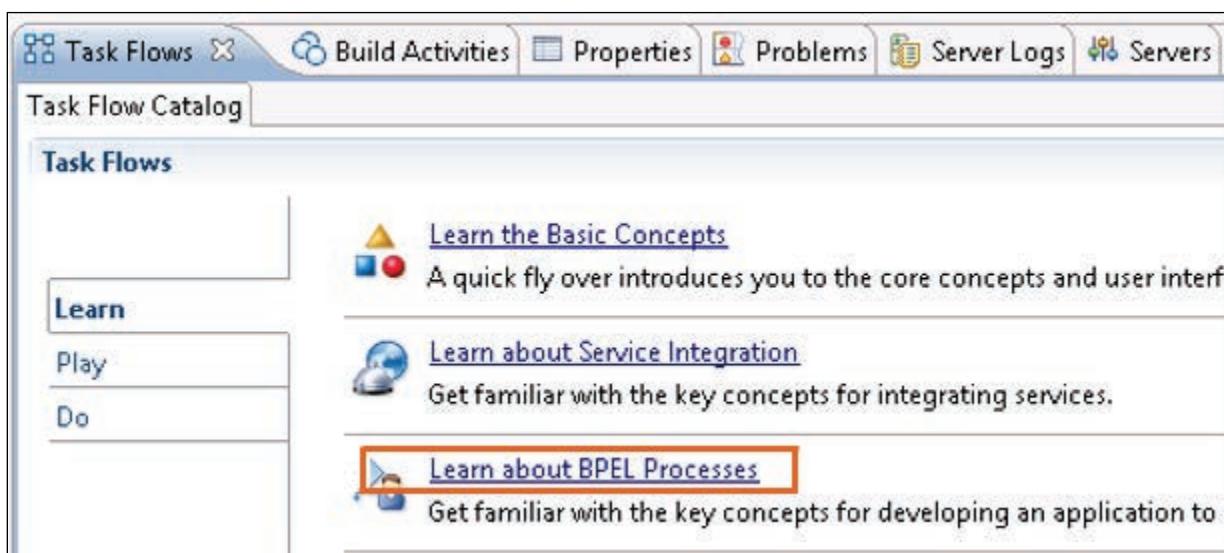
## Part 8. Use the Task Flows view.

Task flows are a way to learn related concepts and tasks in an interactive manner. Learning task flows briefly describes key concepts to quickly get you up to speed with IBM Integration Designer. Creating task flows shows you how to do key tasks.

A task flow puts together all of the different tasks that are required to achieve a goal such as creating a service, starting with setup or planning, and ending with testing. Task flows are presented as a series of grouped tasks; each task is a link. When you click a link, wizards and editors start, along with an information window, so that you are learning in the context of the task you are doing.

1. Use the Task Flows view to open the education resources on business processes.

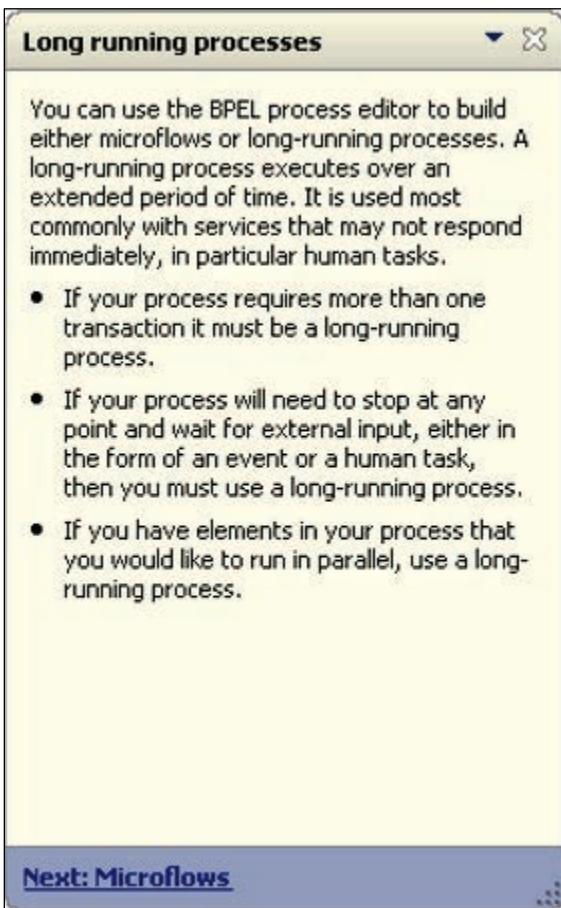
1. Switch to the **Task Flows** view.
2. In the **Task Flows** navigator, verify that you are on the **Learn** tab.
3. Click the **Learn about BPEL Processes** link.



After clicking the link, you are presented with a series of tasks that are involved in creating a business process. Feel free to click any of the links that interest you.



Each task that you click reveals a dialog box with information about the task.



4. When you are done exploring in the **Task Flows** view, switch to the **Task Flow Catalog** tab.



2. Explore the interactive Task Flows samples.

1. In the **Task Flows** navigator, switch to the **Play** tab.
2. Click **Create a Sample BPEL Process**.

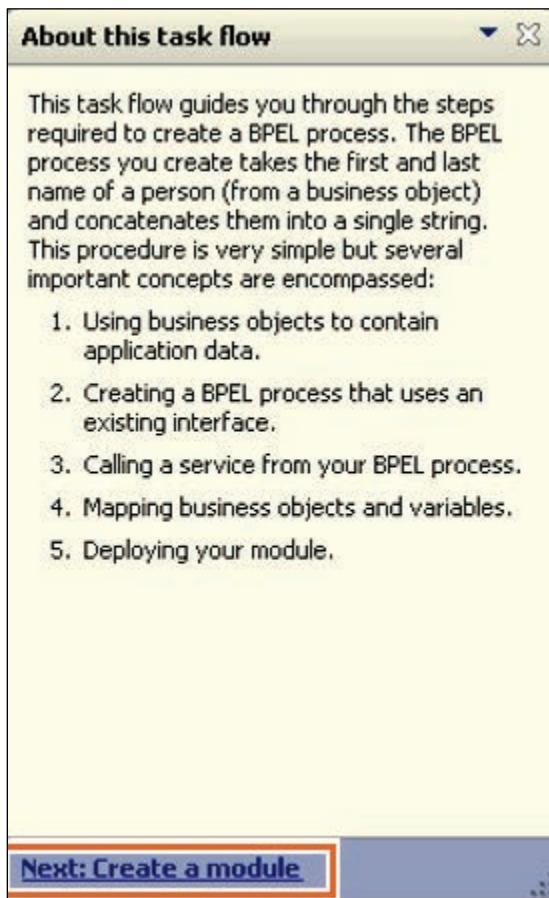
The screenshot shows the 'Task Flow Catalog' interface. At the top, there are tabs for 'Task Flow Catalog' and 'Learn about BPEL Processes'. Below the tabs, a sidebar on the left has three sections: 'Learn', 'Play' (which is currently selected and highlighted in blue), and 'Do'. In the main area, there are two items listed under 'Task Flows': 'Create a Sample Service' and 'Create a Sample BPEL Process'. The 'Create a Sample BPEL Process' item is highlighted with a red border.

When you click a link for one of the interactive samples, you are presented with a series of tasks that are involved in creating the relevant sample. You are expected to work on the tasks one at a time, in the order presented.

3. Click **About this task flow**.

The screenshot shows a dialog box titled 'Create a Sample BPEL Process: In progress'. On the left, under 'Set up', there is a list of tasks: 'About this task flow' (which is highlighted with a red border), 'Create a module', 'Create a business object', and 'Create an interface'. An arrow points from 'Set up' to 'Build'. On the right, under 'Build', there is a list of tasks: 'Create a BPEL process', 'Implementing the process', 'Create the interface of the sample service to be called', 'Set up reference partners', 'Create an invoke activity', 'Create data assignments for the service call', and 'Assign values to variables'.

4. You are presented with a dialog box that explains the task. At the bottom of the dialog box is a link to the next task.

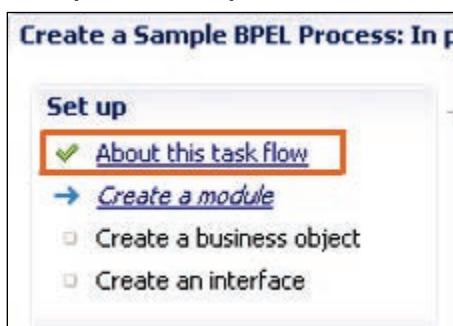


5. Click **Next: Create a module**.

Clicking the link automatically opens the New Module dialog box for input.

6. Close the dialog box and click **Cancel** in the New Module dialog box.

As you complete a task, your status is reflected in the Task Flows view. Each completed step is checked.



As time permits, you can continue creating the sample. Do not proceed past the tasks in the **Setup** section. If you have time at the end of the lab, you can return to this sample and continue.

7. When you are finished exploring the sample, close IBM Integration Designer. You use a different workspace in the next exercise. Do not save any changes.

## Results

**In this exercise, you worked with IBM Integration Designer and explored its capabilities and preferences. You used the Installation Manager to find product updates. You also browsed through several IBM Integration Designer graphical editors, views, and perspectives.**

## 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](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.wbp.m.main.doc/kc-homepage-bpm.html](http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.7/com.ibm.wbp.m.main.doc/kc-homepage-bpm.html)

## **Unit 5** Service Component Architecture overview

IBM Training



# **Service Component Architecture overview**

**IBM Business Process Manager V8.6**

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## 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

### *Unit objectives*

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

## Topics

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

## Overview of SCA modules

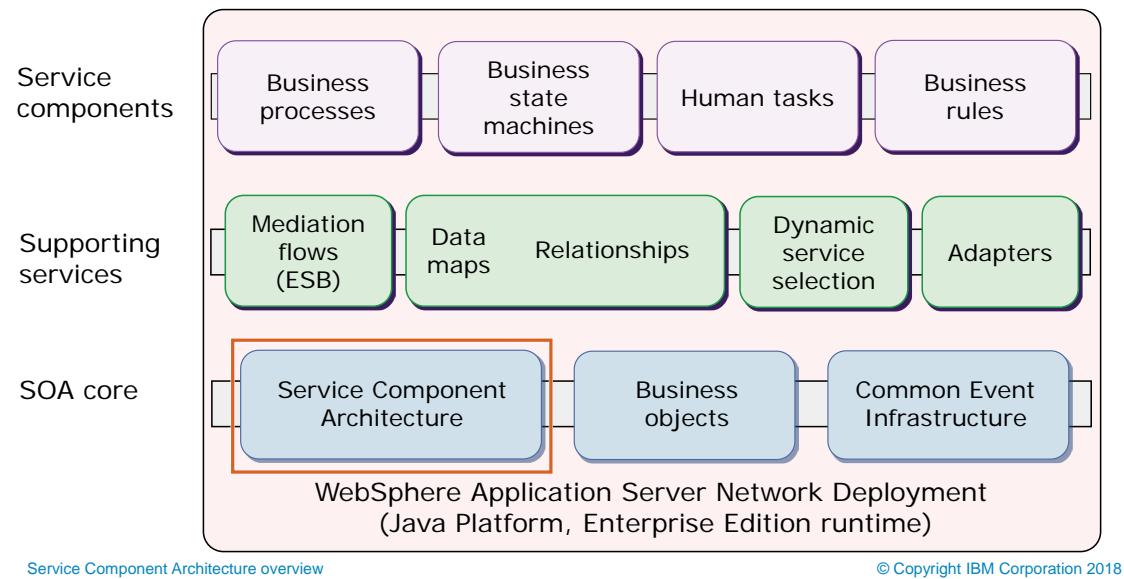
Service Component Architecture overview

© Copyright IBM Corporation 2018

*Overview of SCA modules*

## Service Component Architecture (SCA)

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



### Service Component Architecture (SCA)

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

### *Overview and business value of SCA*

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.

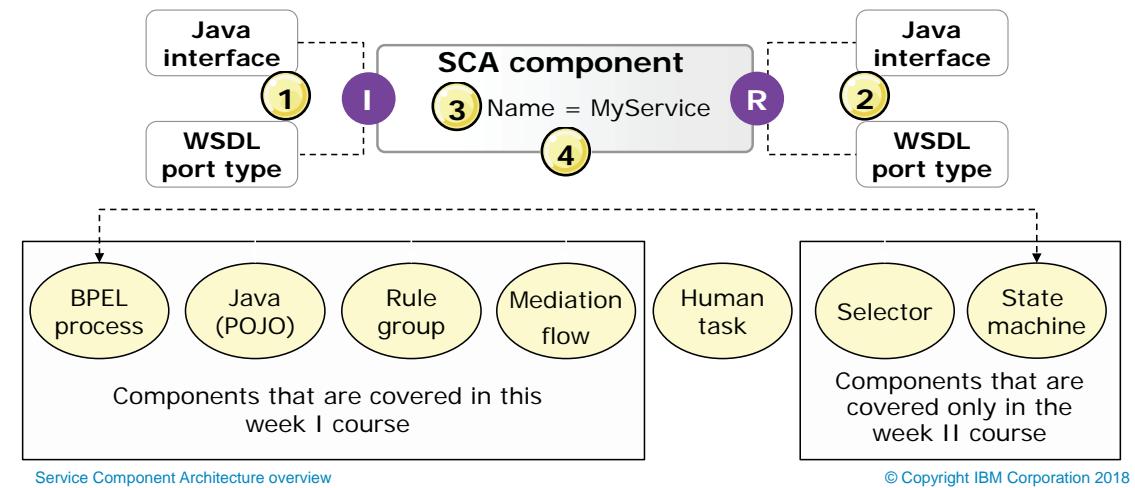
The Service Component Definition Language (SCDL) provides the basis of SCA. SCDL is an XML-based definition language, and is used to define all SCA artifacts in a project. IBM Integration Designer automatically generates the appropriate SCDL definitions when building an SCA-based application. A component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



Service Component Architecture overview

© Copyright IBM Corporation 2018

### SCA components

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.

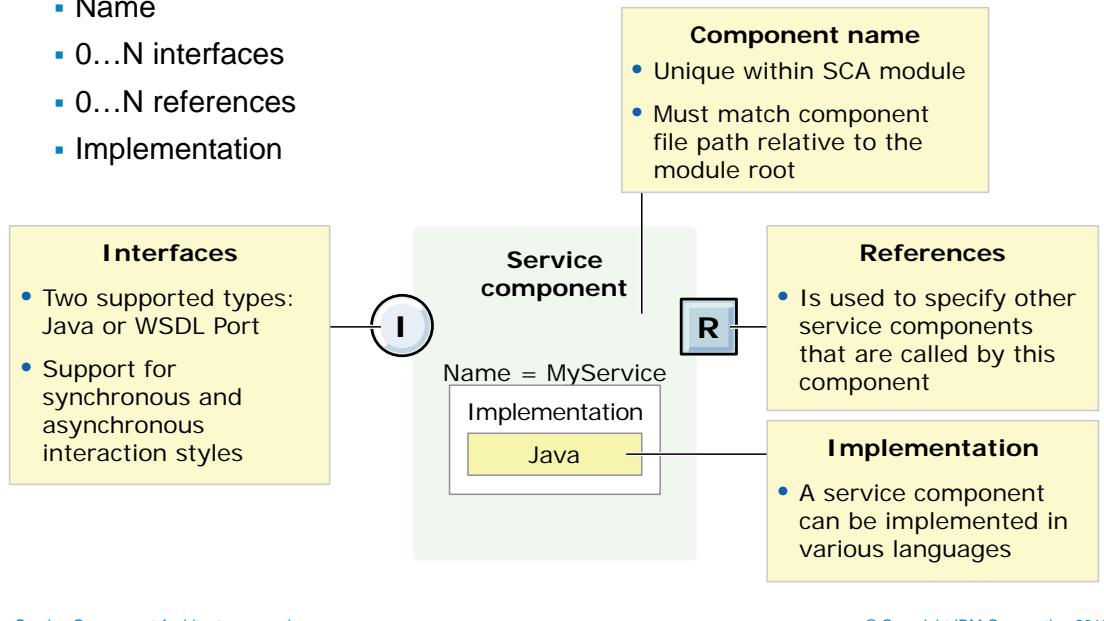
Each service component can be implemented in various ways, which the implementation definition specifies. Service components can invoke other service components or imports that are defined in the current service module. When a component invokes another component or import, the appropriate reference must 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



Service Component Architecture overview

© Copyright IBM Corporation 2018

### SCA component definition

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

Service Component Architecture overview

© Copyright IBM Corporation 2018

### *SCA component interfaces*

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**.

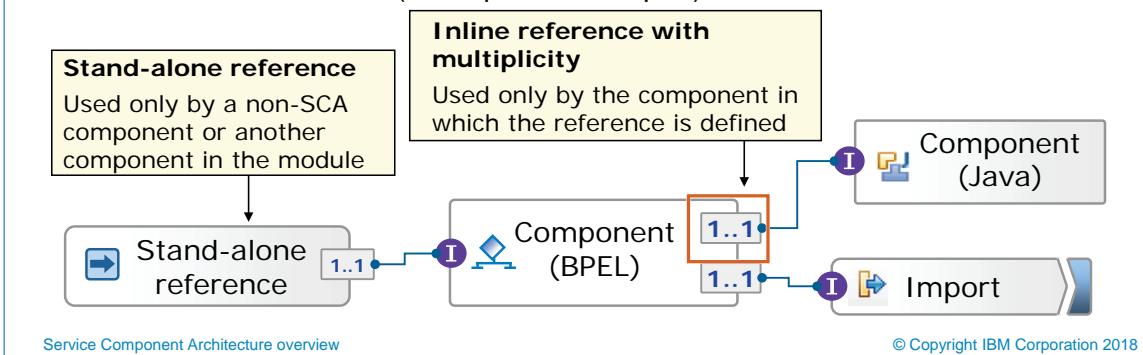
A WSDL document contains the following definitions:

- 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)



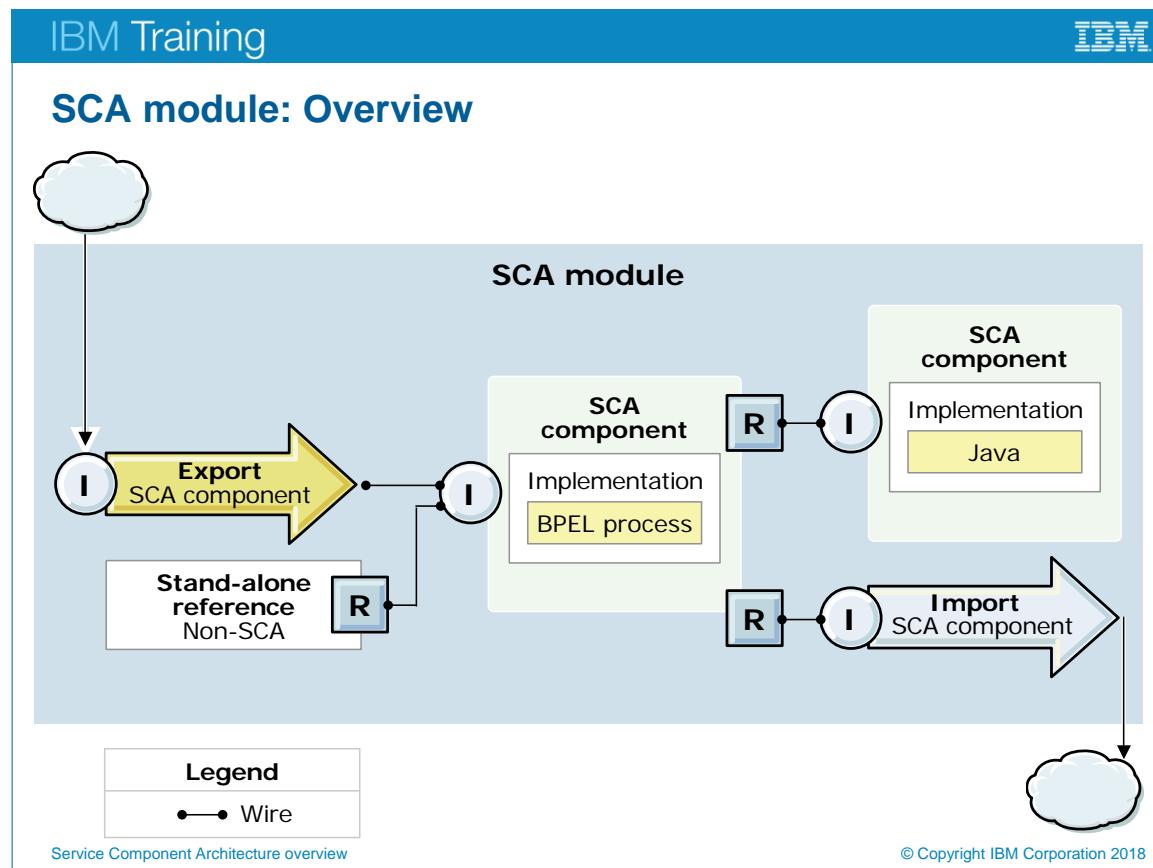
### SCA component references

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

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

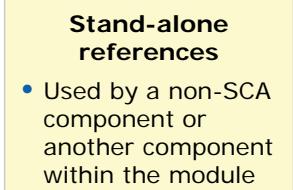
A service module can have zero or more exports included with it. An export is used to expose a particular service to clients outside the current SCA module.

### 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.

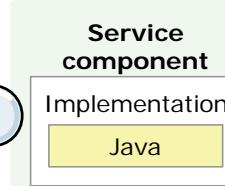
## References and wires

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



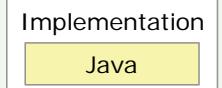
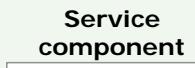
### Inline reference

- Used only by the component in which the reference is defined



### Wire

- Identifies the target service component or import for the reference definition



Service Component Architecture overview

© Copyright IBM Corporation 2018

## References and wires

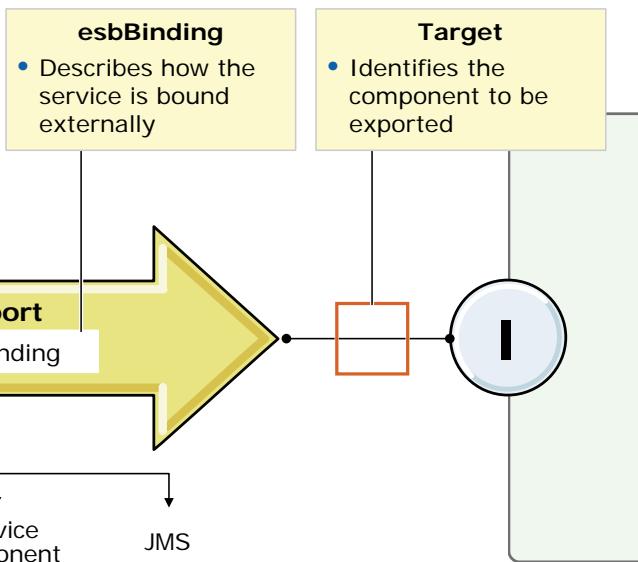
This slide reveals some of the details of defining references. Each reference has a name, and this name is used to look up the appropriate service by a client that uses the client programming model. In addition to the name, a reference also includes an interface element. The multiplicity for a reference 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



Service Component Architecture overview

© Copyright IBM Corporation 2018

## Exports

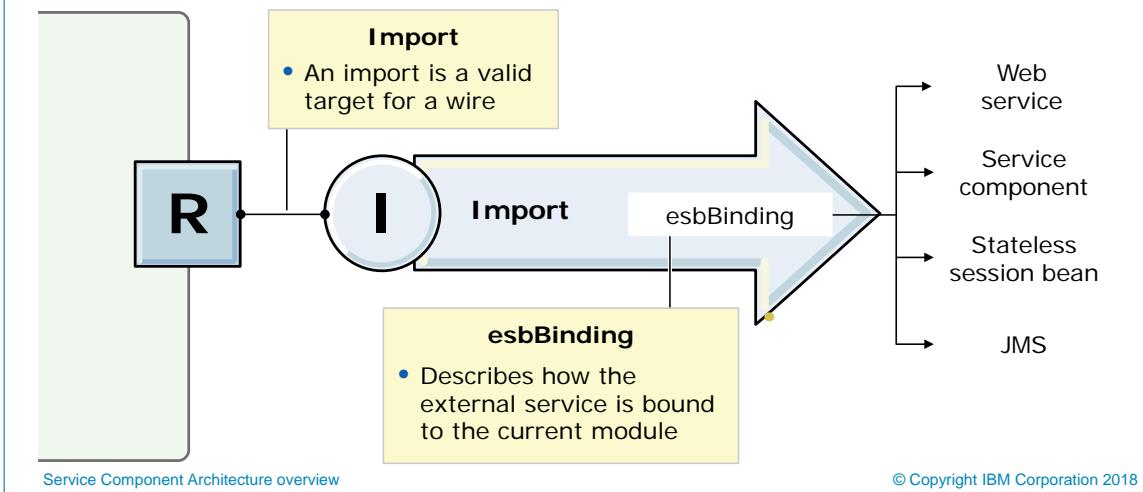
Exports define interactions between SCA modules and service requesters. SCA modules use exports to offer services to others.

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



### Imports

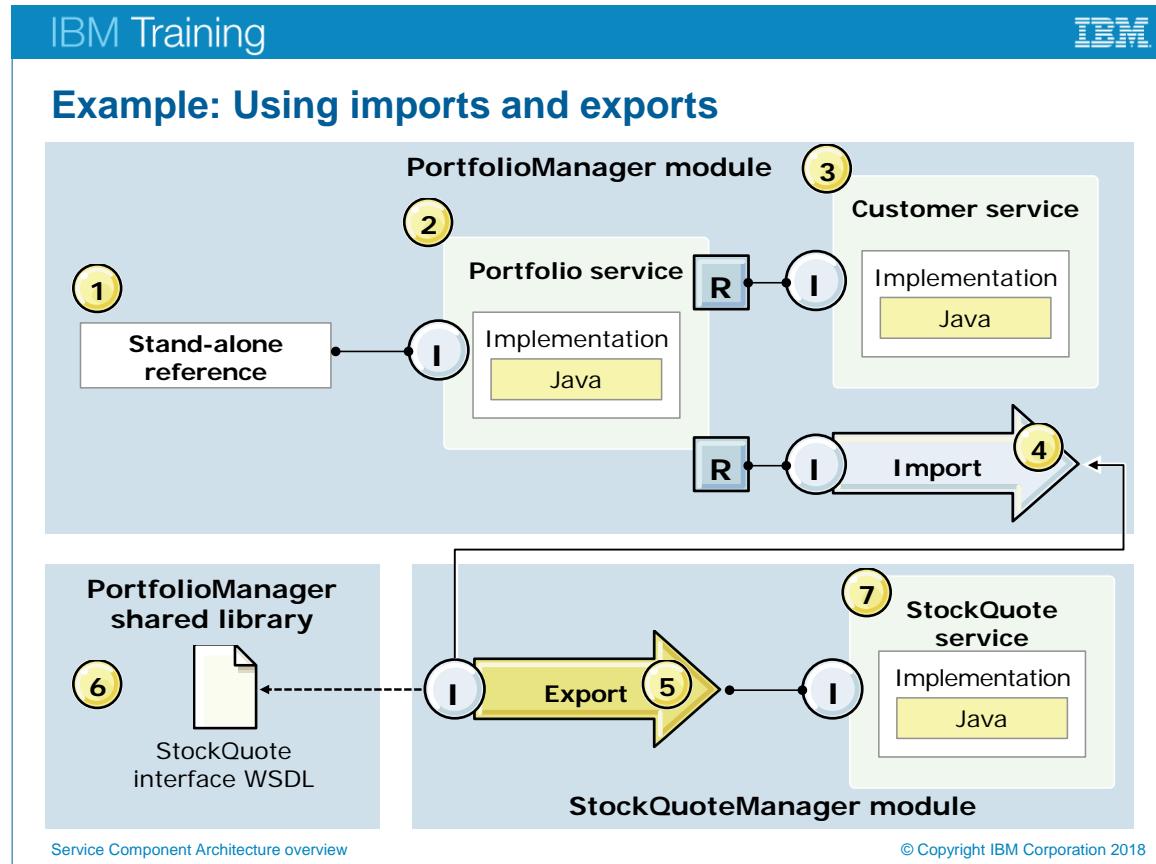
Imports define interactions between SCA modules and service providers. SCA modules use import components to access external services (services that are outside the SCA module) by using a local representation. Import bindings define the specific way that an external service is accessed.

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.

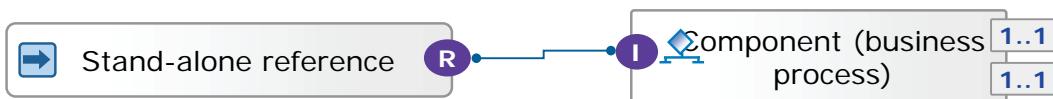


### Example: Using imports and exports

- 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.
- 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.
- The customer service is an SCA component that is implemented in Java and gets invoked by the portfolio service.
- The import invokes the StockQuote service, which is in a separate module through the export.
- The StockQuote service exposes itself through an export component.
- 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.
- 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



### SCA module components: Stand-alone references

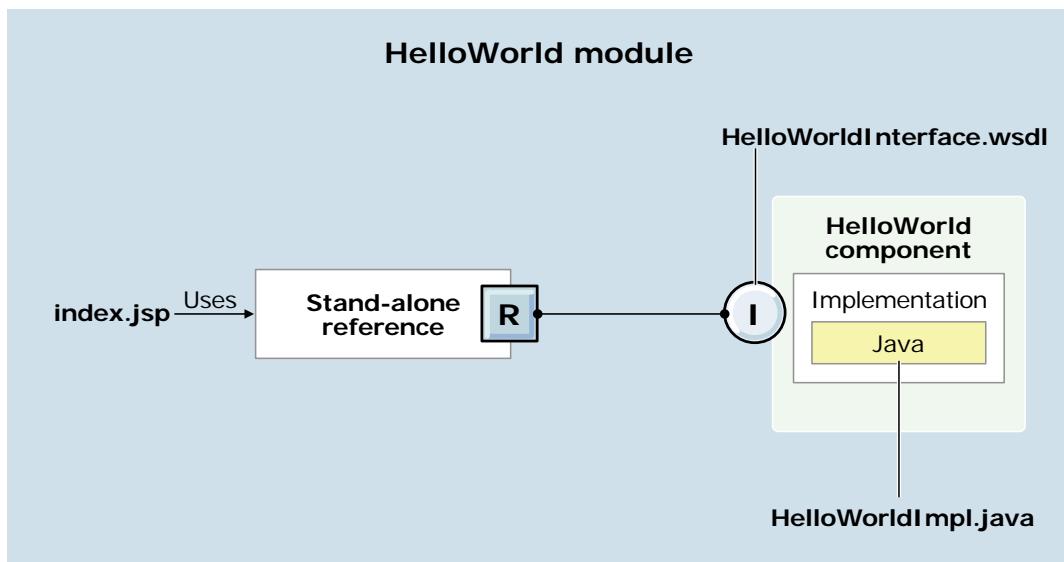
Stand-alone references are references to applications that are not defined as SCA components (for example, JavaServer Pages or servlets). With stand-alone references, these applications interact with SCA components. Stand-alone references can be used in a mediation module. Stand-alone references do not 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.

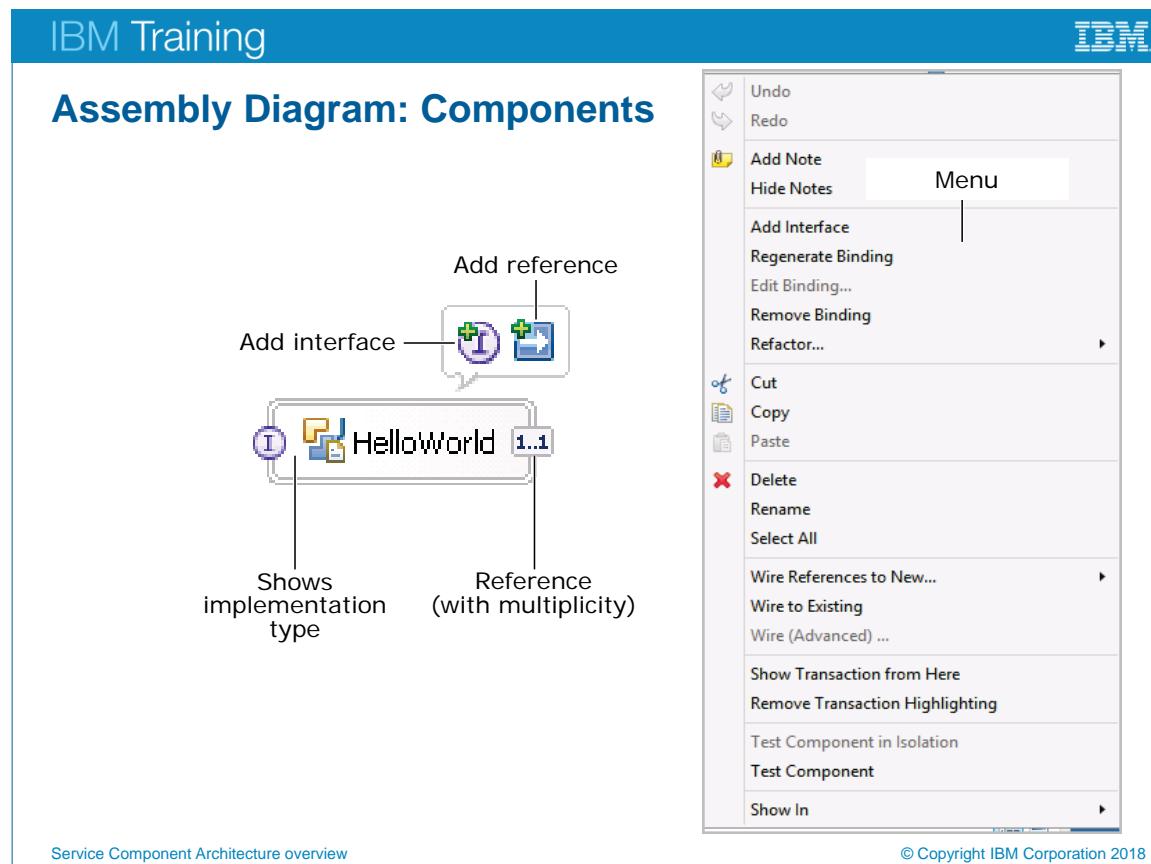
## Example: HelloWorld



Service Component Architecture overview

© Copyright IBM Corporation 2018

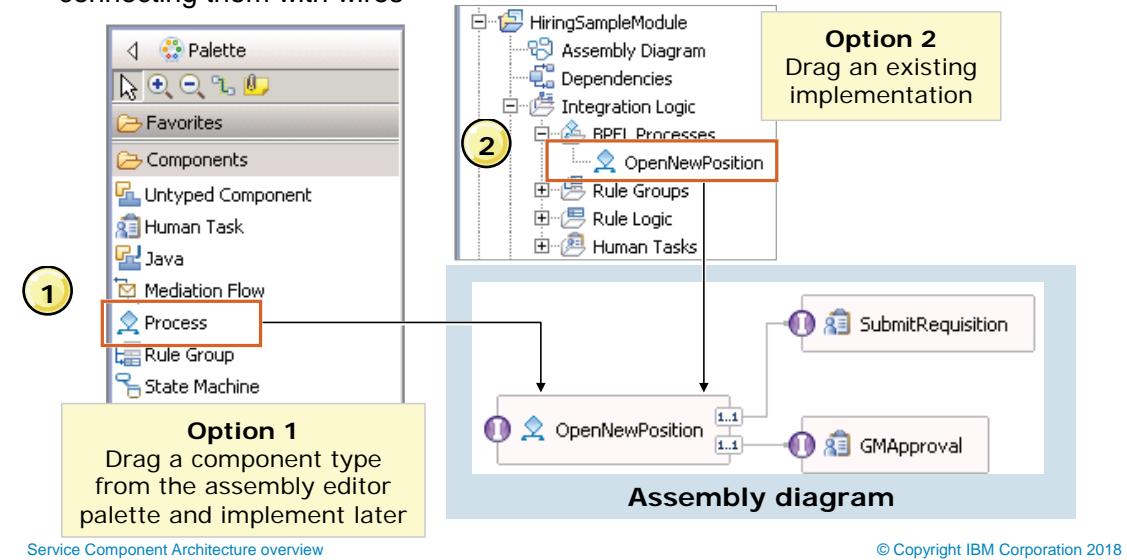
*Example: HelloWorld*



### Assembly Diagram: Components

## 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



### Composing SCA modules: Assembly Diagram editor

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.

## Testing SCA applications

Service Component Architecture overview

© Copyright IBM Corporation 2018

*Testing SCA applications*

## 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

### *Testing SCA components in the integration test client*

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.

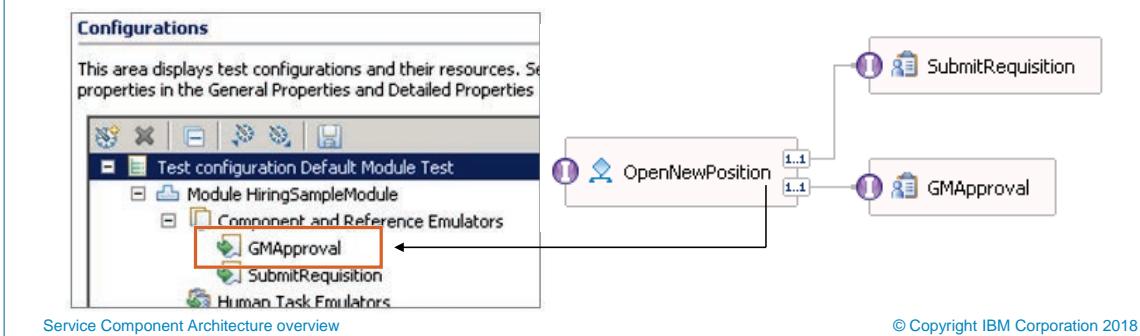
Like other IBM integrated development environments (such as IBM Rational Application Developer), IBM Integration Designer provides an integrated unit test environment for deploying and testing your business applications during the development process. As part of the installation process of IBM Integration 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

## 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



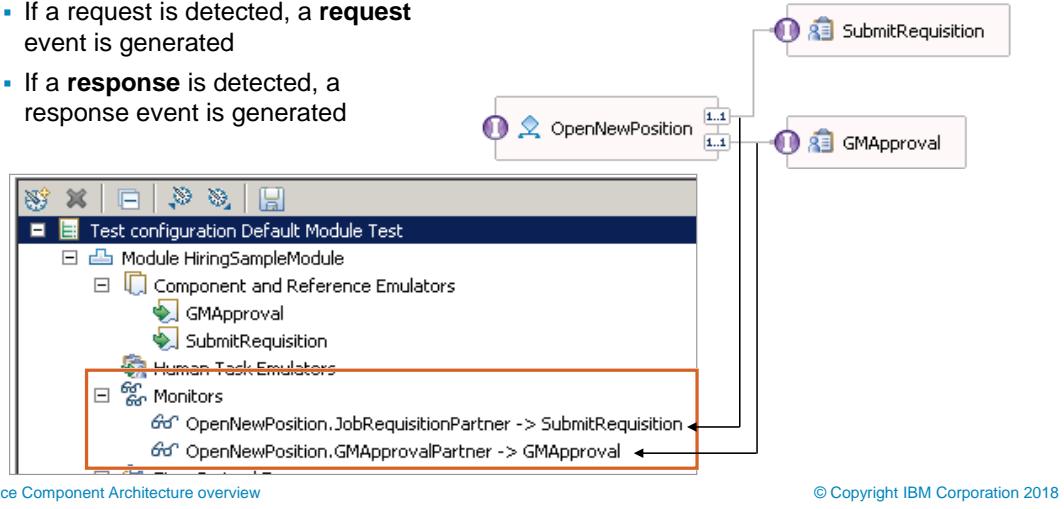
### Emulators

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



### Monitors

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:

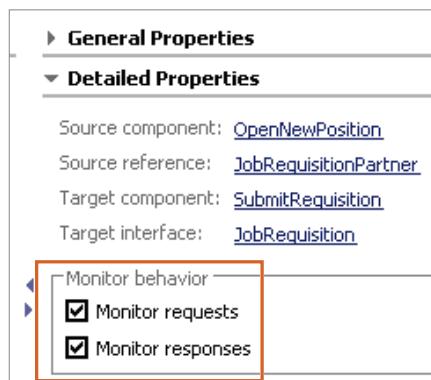
- Invoke event (manual):** This event is an interactive event. The event is generated when you start the integration test client or when you click the “invoke” icon in the integration test client. By using this event, you can select a test configuration, module, interface, and operation. You specify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



The screenshot shows a configuration interface for a monitor. On the left, there's a sidebar with 'General Properties' and 'Detailed Properties' sections. Under 'Detailed Properties', there are fields for 'Source component' (OpenNewPosition), 'Source reference' (JobRequisitionPartner), 'Target component' (SubmitRequisition), and 'Target interface' (JobRequisition). Below these, the 'Monitor behavior' section is expanded, showing two checkboxes: 'Monitor requests' and 'Monitor responses', both of which are checked and highlighted with an orange border.

## 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

## Using a test data table for testing

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

Name	Type
CustomerApplication	CustomerApplication
accountNumber	string
applicationDate	string
applicationDecision	boolean
comments	string
companyName	string
contactFirstName	string
contactLastName	string
contactPhoneNumber	string
creditRating	string
creditReportNeeded	boolean

Service Component Architecture overview

© Copyright IBM Corporation 2018

### Using a test data table for testing

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

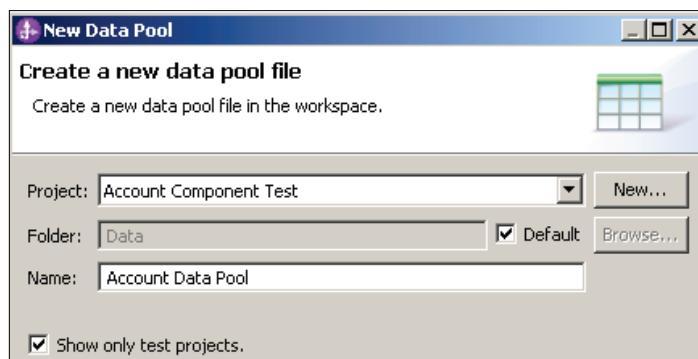
- The integration test client has a stand-alone data pool wizard

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



### *Using a stand-alone data pool for testing*

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.

IBM Training IBM

## Overview: Testing SCA components (1 of 2)

**Assembly editor**

1. Right-click the component and click **Test Component** or **Test Component in Isolation**

2. Switch to the **Configurations** tab to view emulators and monitors

**Configurations**

This area displays test configurations and their resources. Select a test configuration or properties in the General Properties and Detailed Properties sections. [More...](#)

**Test Component in Isolation**  
tests the component and automatically configures monitors and emulators for all other components

**Test Component**  
automatically configures monitors but not emulators; it tests the module that starts with the selected component

Service Component Architecture overview © Copyright IBM Corporation 2018

### Overview: Testing SCA components

The test client interface has two main features:

1. Events page
2. 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.

1. 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.

## Overview: Testing SCA components (2 of 2)

The screenshot shows the 'Events' and 'Detailed Properties' sections of the SCA testing interface.

**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)

Service Component Architecture overview © Copyright IBM Corporation 2018

## 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

### *Testing SCA modules in the integration test client*

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 parameters 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

Service Component Architecture overview © Copyright IBM Corporation 2018

- 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

### Overview: Testing SCA modules

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

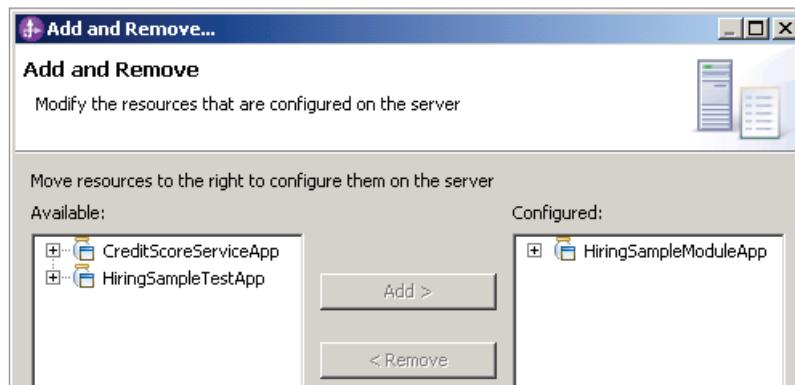
- You can use the “maximize” icon to view the input editor in full screen.
- You can use the arrow keys to browse through the input fields.
- The input editor supports multi-line data entry, and the input dialog box is resizable.
- Icons indicate the input structure type and whether the input value is 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**.

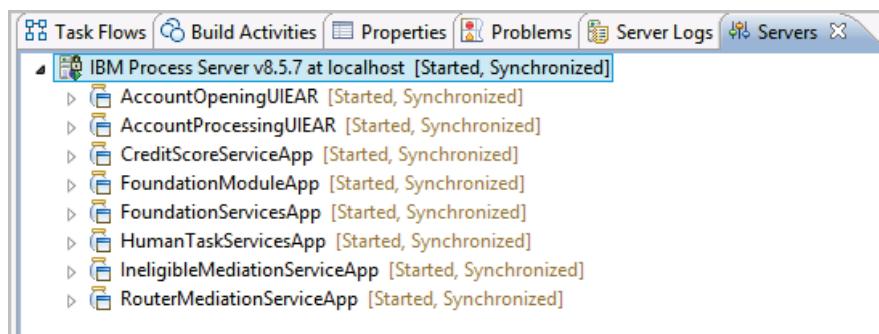
## Overview: Testing SCA modules (2 of 2)

- 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



## 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



Service Component Architecture overview

© Copyright IBM Corporation 2018

*Using the Servers view when testing modules*

## 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)

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

Service Component Architecture overview

© Copyright IBM Corporation 2018

### Server Logs view

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:
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 OpenNew
End invoke (OpenNewPosition:createPosition)	000000f5	End of the one-way invocation of operation OpenNewP

Service Component Architecture overview

© Copyright IBM Corporation 2018

### Cross-component tracing

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

## 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.

## 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.

## Exercise 2: Exploring IBM Integration Designer, part II

- 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

### *Exercise 2: Exploring IBM Integration Designer, part II*

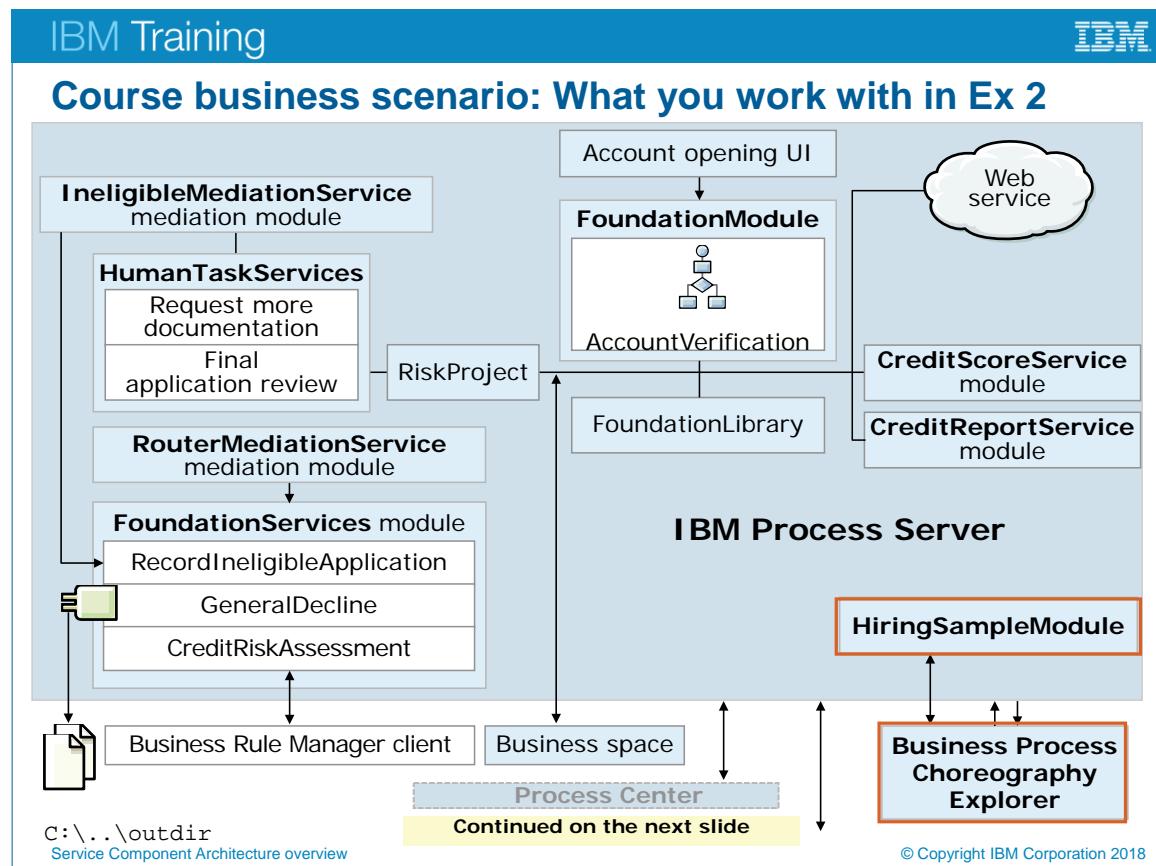
In this exercise, you:

- 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

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.



Course business scenario: What you work with in Ex 2

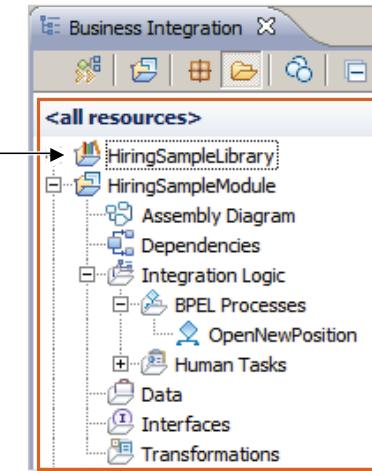
## Components that are required for Exercise 2

Prebuilt components that are imported in the lab:

1. **HiringSampleLibrary**
2. **HiringSampleModule**  
Previously imported in Exercise 2

New components that you create in the lab:

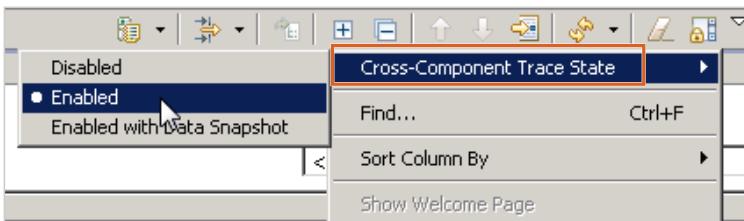
**None**



### *Components that are required for Exercise 2*

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 2



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-be64-54345
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

Service Component Architecture overview © Copyright IBM Corporation 2018

### Use cross-component tracing in Exercise 2

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 ◊	Ov
<input checked="" type="checkbox"/>	5	GMApproval	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

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

Items found: 2 Items selected: 0 Page 1 of 1

Service Component Architecture overview © Copyright IBM Corporation 2018

### *Test by using the Business Process Choreographer Explorer*

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.

## Exercise 2: Exploring IBM Integration Designer, part 2

### Purpose:

In this exercise, you use the IBM Integration Designer environment to explore various SCA components and examine them in the test environment.

To simplify and accelerate the development of applications, the IBM Integration Designer environment provides a layer of abstraction that separates the visually presented components from the underlying implementation. You can use IBM Integration Designer to easily assemble an SCA application visually and generate the underlying physical resources automatically. IBM Integration Designer also includes a robust set of test components and a fully functional runtime environment.

### Requirements

Completing the exercises for this course requires a lab environment. This environment includes the exercise support files, IBM Process Designer, IBM Process Center, and the IBM Integration Designer test environment.

#### Part 1. Assemble an SCA application.

In this portion of the exercise, you add SCA components to an assembly diagram and wire them together to form an application. You use a modified version of the “hiring sample” process application from the previous exercise.

##### 1. Open the Exercise 2 workspace

1. On your desktop, open the folder that is labeled **Exercise Shortcuts**.
2. Double-click the shortcut that is labeled **Exercise 2**.

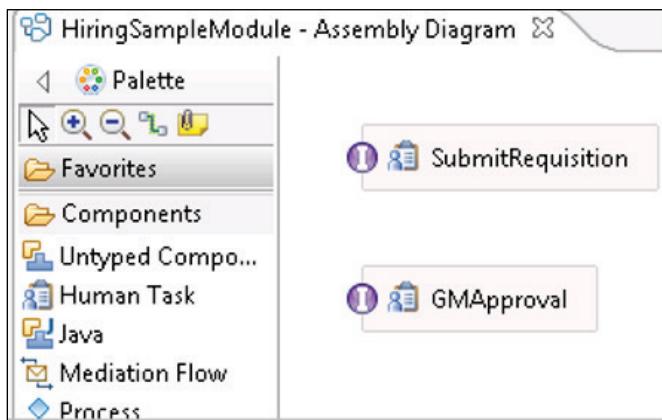
Allow Integration Designer a few moments to build the workspace. You can view the workspace build status at the lower-right corner of the Integration Designer. Wait until the status reaches 100%, at which point the workspace is built, and the status progress bar disappears.

3. If the **Getting Started** tab is open, close it.

##### 2. Assemble the SCA components.

1. In the Business Integration view, expand **HiringSampleModule** and double-click **Assembly Diagram**.

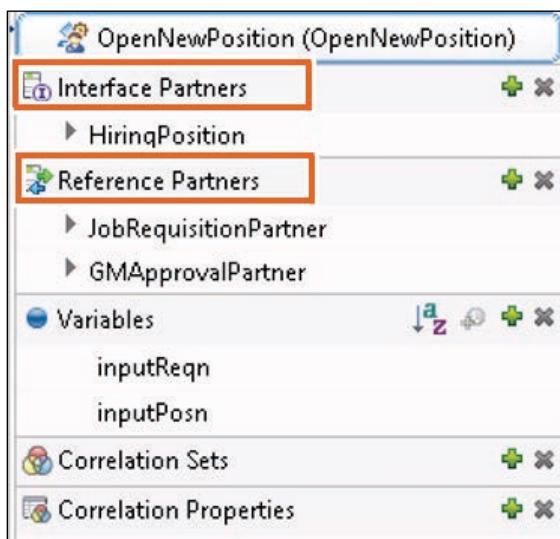
Some SCA components are already added to the diagram: the **SubmitRequisition** and the **GMAApproval** human tasks, for example.



2. In the **Business Integration** view, expand **HiringSampleModule > Integration Logic > BPEL Processes**.
3. Double-click **OpenNewPosition** to open it in the process editor.

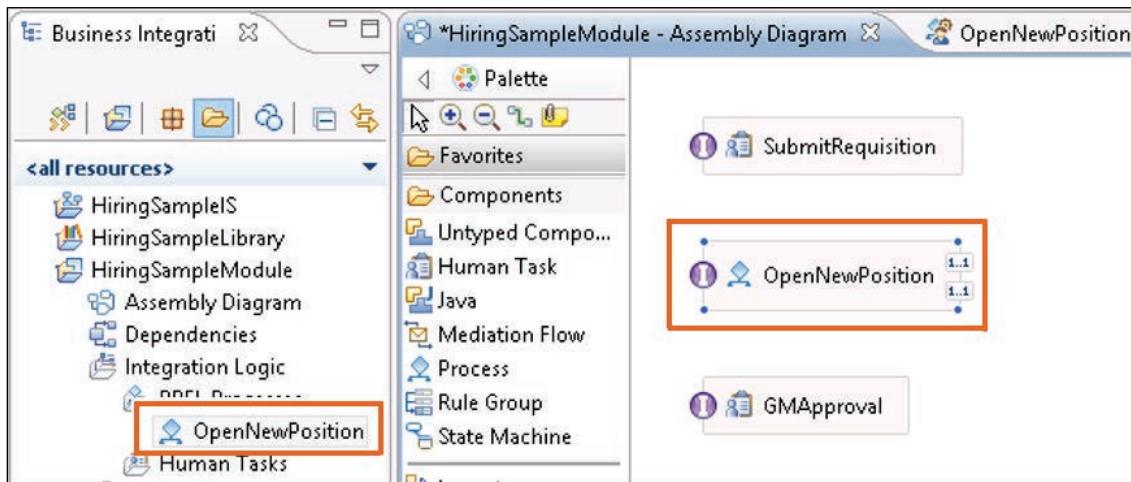
This process is a series of activities that are designed to work through the business logic that you tested in the previous exercises.

In the **tray** to the right, note the **Interface Partners** and **Reference Partners**. The interface partner is used to invoke this process. The reference partners represent the interfaces of the services that this process invokes.

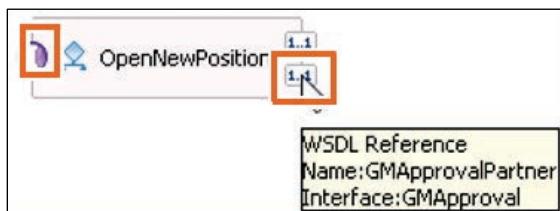


4. Switch to the **HiringSampleModule** assembly diagram.

5. Drag the **OpenNewPosition** process onto the **HiringSampleModule** assembly diagram.



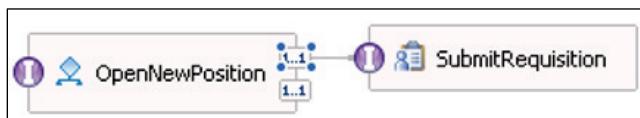
When you drag the process onto the assembly diagram, it is represented as an SCA component. The interface partner becomes the interface of the component. The reference partners become references on the SCA component. If you hover the mouse over the interface or references, a dialog box provides details. The BPEL process that you examined previously becomes the implementation behind the component and is represented as the implementation icon.



The cardinality of both references is  $1..1$ . This cardinality indicates that the reference expects that one (and only one) service is wired to it. The process activity that invokes the service that is represented as the reference partner invokes only one possible service.

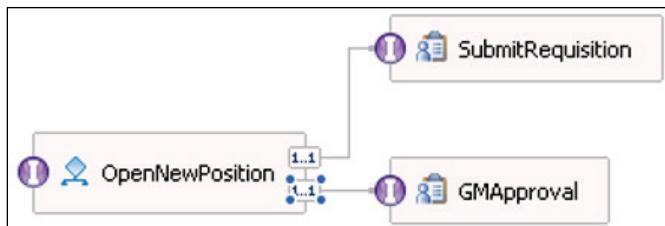
6. Hover over the references until you locate the **JobRequisitionPartner** reference. When you locate it, right-click it and click **Wire to Existing** from the menu.

This action wires the reference to the **SubmitRequisition** component.



7. Right-click the **GMAccrualPartner** reference and click **Wire to Existing** from the menu.

The assembly of the application is complete.



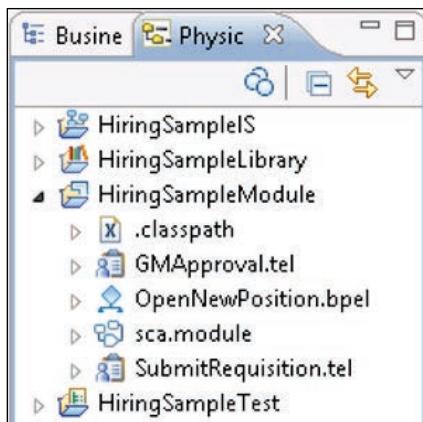
8. Press **Ctrl+S** to save your changes.

## Part 2. Examine project components.

In this portion of the lab, you examine some of the physical components that IBM Integration Designer generates for the visual components with which you interact.

1. Examine the physical components of a project.

1. Click **Window > Show View > Physical Resources** from the menu options.
2. Expand **HiringSampleModule**.



The **Physical Resources** view shows you the files that IBM Integration Designer generates when you create a visual component in one of the editors. For example, the `.tel` file contains the human task rules, the `.bpel` file contains the business process, and the `.module` file contains the module assembly.

3. When you are done exploring, close the **Physical Resources** view.  
Warning: Although it is possible to use the physical resources view to modify project components, it is intended for advanced, experienced users only. Manually changing any of the data in an `sca.module` file, or directly in a `.bpel` file, can result in unpredictable consequences.

## Part 3. Explore staging projects.

In this portion of the exercise, you examine the staging projects that IBM Integration Designer builds during automatic (or manual) builds. For any module project, two Java EE staging projects exist: an enterprise application project and a dynamic web project. If your module contains generic JMS bindings or JAX-RPC web service bindings, you also see an EJB project.

The enterprise application project can be exported as an EAR file for deployment. The enterprise application project consists of Java Platform, Enterprise Edition (Java EE) modules that can be deployed onto application servers. The modules are created from code artifacts such as web application archive (WAR) files, resource adapter archive (RAR) files, enterprise bean (EJB) JAR files, and application client archive (JAR) files. This packaging and configuring of code artifacts into enterprise archive (EAR) modules or stand-alone web modules is necessary for deploying the modules onto an application server.

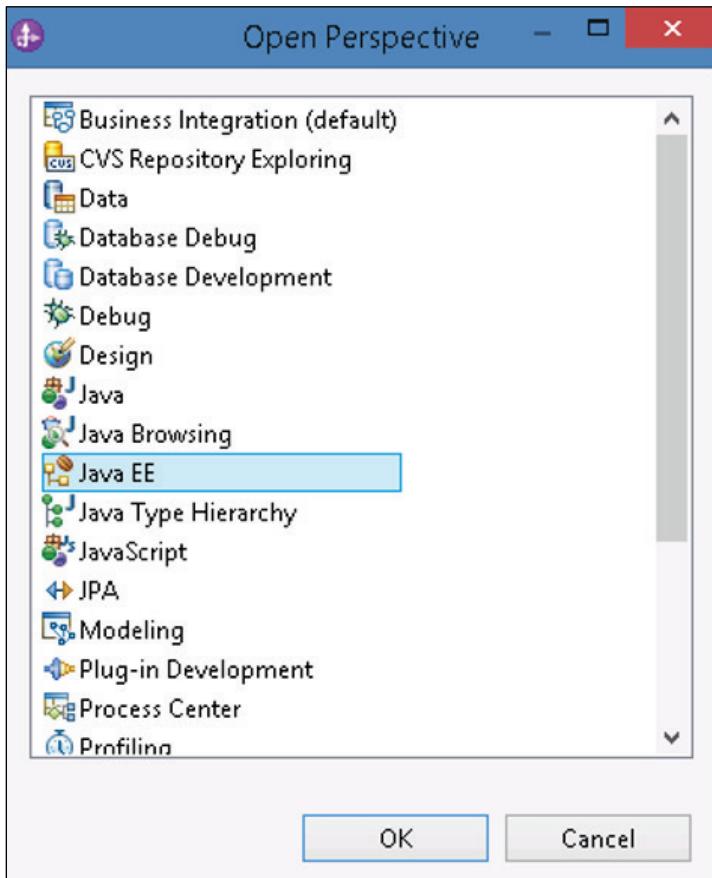
A dynamic web project is a web module that contains servlets, JavaServer Page (JSP) files, and related code artifacts. After assembling a web module, you can install it as a stand-alone application or combine it with other modules into an enterprise application.

### 1. Examine staging projects:

Because the staging projects are platform-specific artifacts that are generated for the Java EE runtime, you must switch to the Java or Java EE perspective to view them.

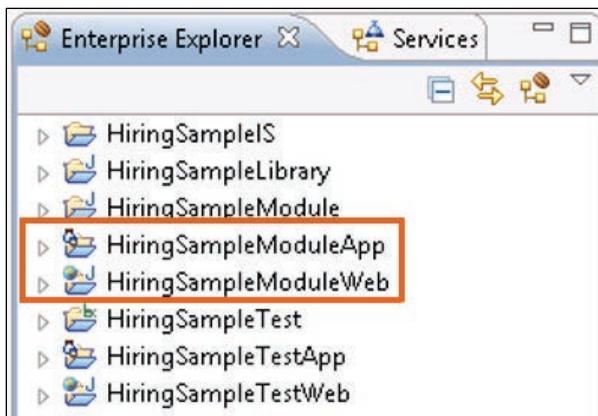
1. Click Window > Open Perspective > Other from the menu options.

2. In the **Open Perspective** window, select **Java EE**.



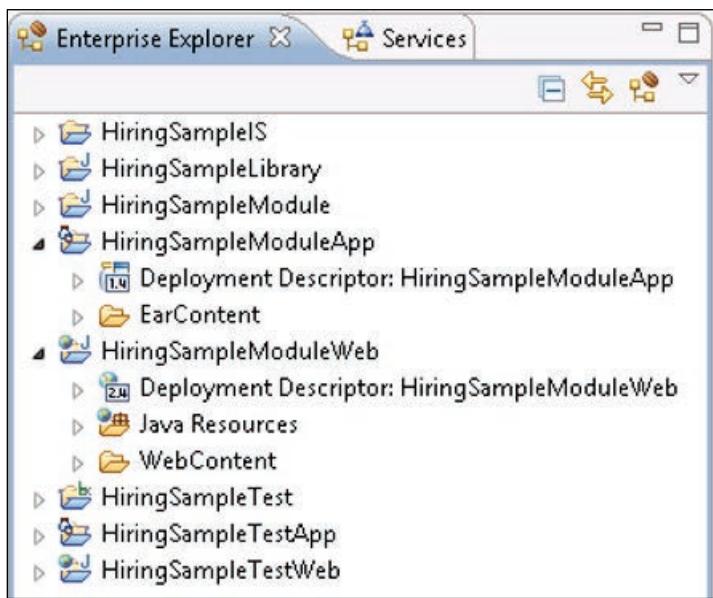
3. Click **OK**.

In the Enterprise Explorer view, the **HiringSampleModule** has an enterprise application project and a dynamic web project. The project names are built based on the application name. Thus, the enterprise application project is named **HiringSampleModuleApp** and the dynamic web project is named **HiringSampleModuleWeb**.

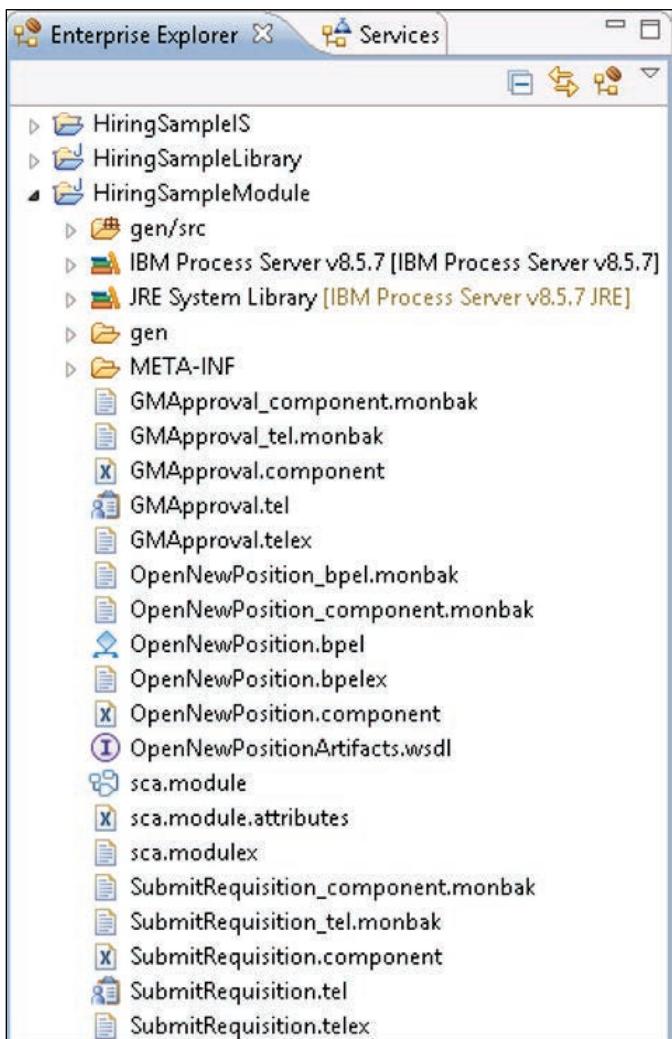


When you deploy projects to the test environment, you are deploying the applications that are represented here. These project names are displayed in the **Add and Remove** dialog box when you deploy to the server.

4. Expand **HiringSampleModuleApp** and **HiringSampleModuleWeb** and examine the artifacts.



5. If time permits, expand **HiringSampleModule** and examine the other Java EE artifacts in the view.



6. When you are finished exploring, switch to the Business Integration perspective by clicking the icon in the upper right.

#### Part 4. Use the IBM Integration Designer test environment.

In this portion of the exercise, you learn about the IBM Integration Designer test environment. In IBM Integration Designer, the integration test client is the designated tool for testing modules and components. You can use the sophisticated user interface of the test client to easily manage and precisely control your tests. The testing is generally done on the interface operations of your components. Additionally, you can determine whether the components are correctly implemented and the references are correctly wired.

Using the integration test client, you can test:

- An individual module
- A set of interacting modules
- An individual component
- A set of interacting components

When you begin a test by invoking an operation, the integration test client automatically detects the deployment state of the modules that are going to be tested. If any of the modules are not deployed, the deployment location wizard automatically opens. This wizard can be used to select the server where you want to deploy the modules. If the server is not already running, it is started automatically.

## Test configurations

Test configurations are used to control your tests. A test configuration specifies one or more modules to test. Each of these modules might 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 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. However, you can choose to edit and customize the default test configuration, or you can create and edit one or more test configurations.

## Emulators

You can use the integration test client 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 are available:

- Manual
- Programmatic

A manual emulator is an emulator for which you must specify runtime response values for an emulated component or reference. If you are testing an entire module, the default test configuration contains manual emulators for all unimplemented components and unwired references. However, suppose that you are testing a set of components or an individual component within a module. In this case, the default test configuration contains manual emulators for any other components that were not selected for testing regardless of whether they are implemented or not. Although manual emulators are added by default, you can remove the manual emulators or redefine them as programmatic emulators.

When a manual emulator is encountered during a test, a manual emulate event is generated. When the test pauses, you can manually specify some output parameter values or throw an exception for the emulated components or references. By comparison, when a programmatic emulator is encountered during a test, a programmatic emulate event is generated. A Java program that is contained in a visual snippet or a Java snippet automatically provides the output parameter values or exceptions.

## Monitors

When the integration test client generates a default test configuration or you add a test configuration, monitors are automatically added for any component wires and exports that are found in the modules of the test configuration. When you invoke an operation and run a test, these monitors listen for any 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. These events show parameter data that flows across the wires. They are added to the test trace of events that are displayed in the Events area of the integration test client.

Although monitors are automatically added for the wires and components of your test configuration modules, 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.

## Testing an individual component

In this section, you test on an individual component in the hiring sample application. You examine the emulators and monitors that the integration test client automatically configures, and you learn how to save test data.

Three different use cases in the hiring sample application are:

- The hiring requisition is for a new job, which does not need approval.
- The hiring requisition is for an existing job, which needs approval from the general manager.
- The general manager might approve or reject requisitions for existing job actions.

In the component test, you verify the function of the first use case where a requisition for a new job is automatically approved.

- To run an individual component test, start IBM Process Server.

**Reminder:** There is no IBM Integration Designer 8.6 version. The current version is still at 8.5.7. As a result, the test process server that comes out of the box is also at version 8.5.7. However, that test server has been upgraded to 8.6 level in this environment and you work with the 8.6 release. Furthermore, Process Center 8.6 is also installed and you work with it in a later exercise.

**Important:** Normally, you go to the **Servers** view, right-click **IBM Process Server v8.6 at localhost** and click **Start** from the menu to start the User Test Environment (UTE) server. However, during testing of the exercises, the server often failed to start at the first attempt when started from within the test environment graphical interface. For that reason, there is a shortcut on the desktop that calls the start script for the process server. **Use that shortcut to start and stop the server as that always works.**

- On your Windows desktop, select the **Start UTE Process Server shortcut**.



- Double-click the shortcut or press Enter to start the server
- A DOS command window is displayed; press any key to continue when prompted.

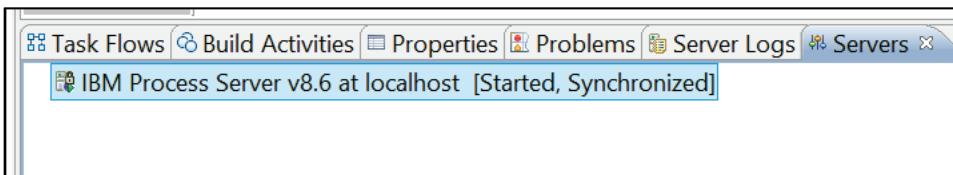
```

Start UTE Process Server - X

CWUPO0001I: Running configuration action detectNewProducts.ant
ADMU0116I: Tool information is being logged in file
C:\IBM\IID\PS\v8.5\profiles\qbmmaps\logs\server1\startServer.log
ADMU0128I: Starting tool with the qbmmaps profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 1564
Press any key to continue . . .

```

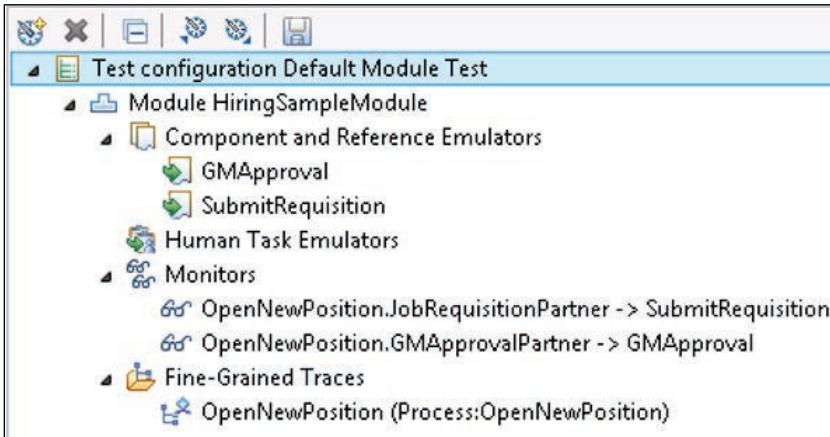
- Return to Integration Designer and verify that the status of the server is **Started**.



2. Use the Test Component in Isolation option to test the OpenNewPosition component.
  1. In the Business Integration view, expand **HiringSampleModule**.
  2. Double-click **Assembly Diagram** to open the assembly editor.
  3. On the assembly editor canvas, right-click the **OpenNewPosition** SCA component and click **Test Component in Isolation** from the menu.  
The **OpenNewPosition** component contains the process that determines whether a job requisition can be automatically approved.
  4. In Integration Test Client, click the **Configurations** tab in the lower-left corner of the window.



5. Examine the emulators and monitors.



6. Switch back to the **Events** tab (in the lower-left corner of the test window) and browse to the **Initial request parameters** table on the right side of the **Detailed Properties** section.

Initial request parameters:	
<input checked="" type="radio"/> Value editor	<input type="radio"/> XML editor
Name	Type
inputReqn	Requisition
reqNum	string
requestor	string
status	string
approvalNeeded	boolean
date	date

7. Enter the following information in the **Value** column.

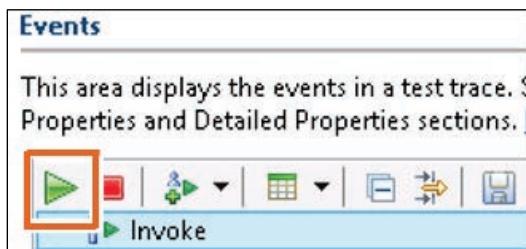
To aid data entry, you can use the maximize icon to maximize the input window to full screen.

- inputReqn
  - reqNum: 001
  - department: Sales
  - location: Atlanta
- inputPosn
  - positionType: New
  - lastName: Smith
  - firstName: John

Initial request parameters		
Name	Type	Value
# inputReqn	Requisition	[...]
reqNum	string	[...] 001
requestor	string	[...]
status	string	[...]
approvalNeeded	boolean	[...] false
date	date	[...] 2016-03-22
department	string	[...] Sales
location	string	[...] Atlanta
empNum	int	[...] 0
gmApproval	boolean	[...] false
gmComments	string	[...]
# inputPosn	Position	[...]
positionType	string	[...] New
replacement	Person	[...]
lastName	string	[...] Smith
firstName	string	[...] John
supervisor	string	[...]
startDate	date	[...] 2016-03-22
payLevel	string	[...]
payType	string	[...]
notes	string	[...]
jobTitle	string	[...]

To edit values, start typing or press F2.

8. Click **Continue** on the Events toolbar.



Information: If you are testing a component for the first time, several actions occur before you are able to test:

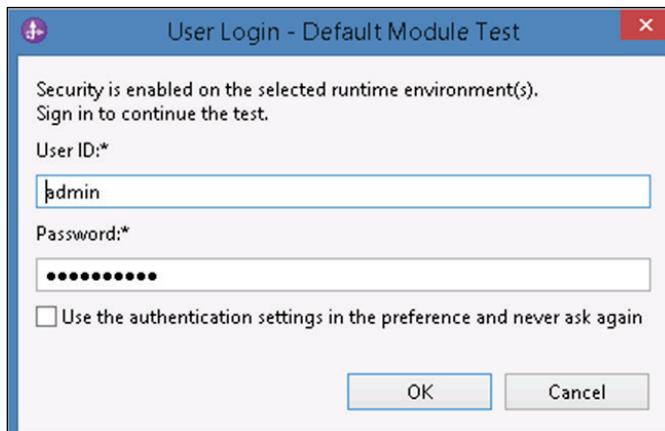
- The IBM Process Server test environment starts, unless it is already started.
- The **HiringSampleModule** is published to the server. You might need to specify a deployment environment.
- The **HiringSampleModule** is started.

This procedure might take several minutes to complete. If you see a message that the module is going to be automatically published to the server, then select **Do not show this message again** and click **OK**.

Note: If you maximized the input window, minimize it to locate the Events toolbar.

9. When the Deployment Location dialog box is displayed, select **IBM Process Server v8.6 at localhost**.
10. Accept the remaining default options and click **Finish**.
11. In the **User Login - Default Module Test** dialog box, accept the default options and click **OK**.

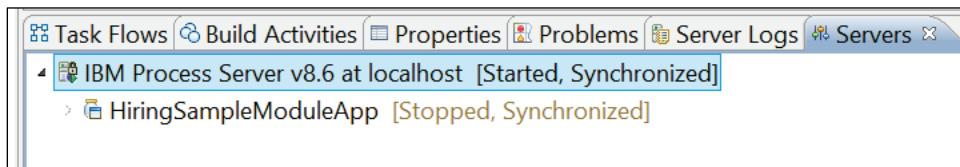
By default, the **User ID** is set to `admin` and **Password** is set to `web1sphere` during IBM Integration Designer installation.



Note: You are not selecting the Use this location as the default and Use the authentication settings in the preference check boxes because they are specific to the individual workspace. Each of your exercises has a specific workspace, so you would select these options for each exercise.

12. Switch to the **Servers** view and expand **IBM Process Server v8.6 at localhost**.

You can see the state of the modules (and the server) in the **Servers** view when the module is deployed.

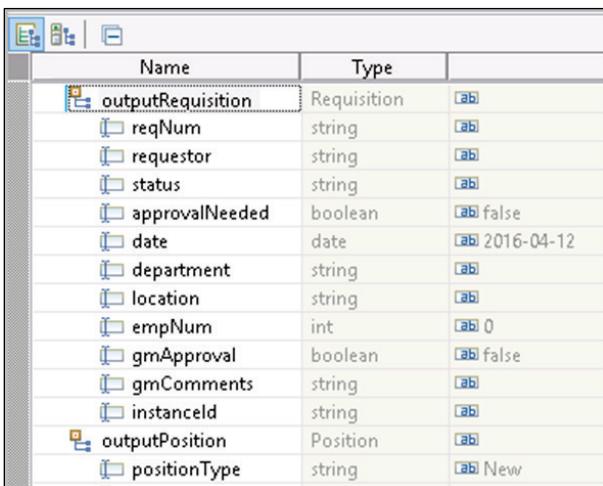


The test stops at the human task emulator **SubmitRequisition**. A client GUI is displayed for the user to enter requisition details. For this test, the most important detail is to set the job position to New.

13. Set the outputPosition > positionType value to: New



14. Click **Continue** again.
15. When the test is complete, select the **Response (OpenNewPosition<-SubmitRequisition:createRequisition)** event.
16. Examine the **Response parameters**.



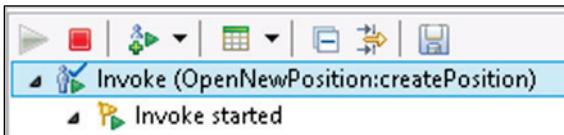
The screenshot shows a table titled "Data Pool Editor" with columns "Name" and "Type". The data rows are as follows:

Name	Type
outputRequisition	Requisition
reqNum	string
requestor	string
status	string
approvalNeeded	boolean
date	date
department	string
location	string
empNum	int
gmApproval	boolean
gmComments	string
instanceId	string
outputPosition	Position
positionType	string

Because the job requisition is new, no approval is needed, so the **approvalNeeded** and **gmApproval** fields have both been set to **false**.

3. Save the test data to the data pool to reduce manual input in subsequent tests.

1. Select the top-most Invoke event in the Events pane, **Invoke (OpenNewPosition: createPosition)**.



2. In the **Initial Request Parameters** section, right-click **inputPosn** and click **Add Value to Pool** from the menu.

This action starts the data pool wizard.

**Information:** A test suite, **HiringSampleTest**, is part of the project interchange file that you imported earlier. This test suite contains a data pool. A data pool is a collection of test data that you might reuse when testing components for this project.

3. Expand **HiringSampleTest** and select **HiringRequisitionDatapool.objectpool**.
4. Click **Finish**.
5. Set the name to: **NewPosition**
6. Accept the remaining default options and click **OK**.

The next time that you run a test, you can then retrieve the data from the data pool by right-clicking the input parameter and clicking **Use Value from Pool** from the menu.

7. Close the **OpenNewPosition\_Test** tab and click **No** when you are prompted to save the test trace.

The module remains deployed until you manually remove it from the server.

### Testing an application

In this portion of the exercise, you test the hiring sample application. You test the application by deploying the modules to the IBM Process Server runtime and by interacting with the BPEL Process Choreographer Explorer user interfaces. Three different use cases in your application are:

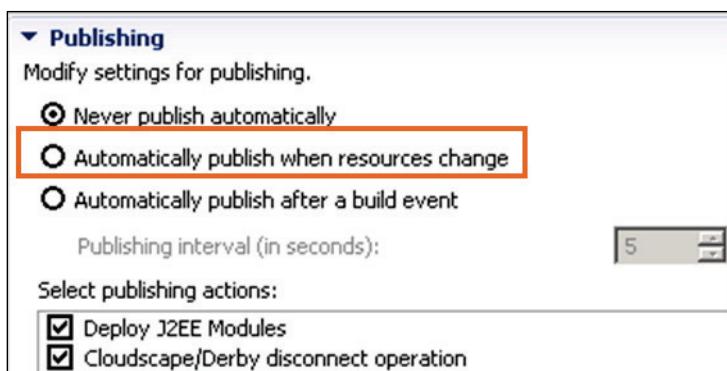
- The hiring requisition is for a new job, which does not need approval.
- The hiring requisition is for an existing job, which needs approval from the general manager.
- The general manager might approve or reject requisitions for existing job actions.

## Examining the test server profile

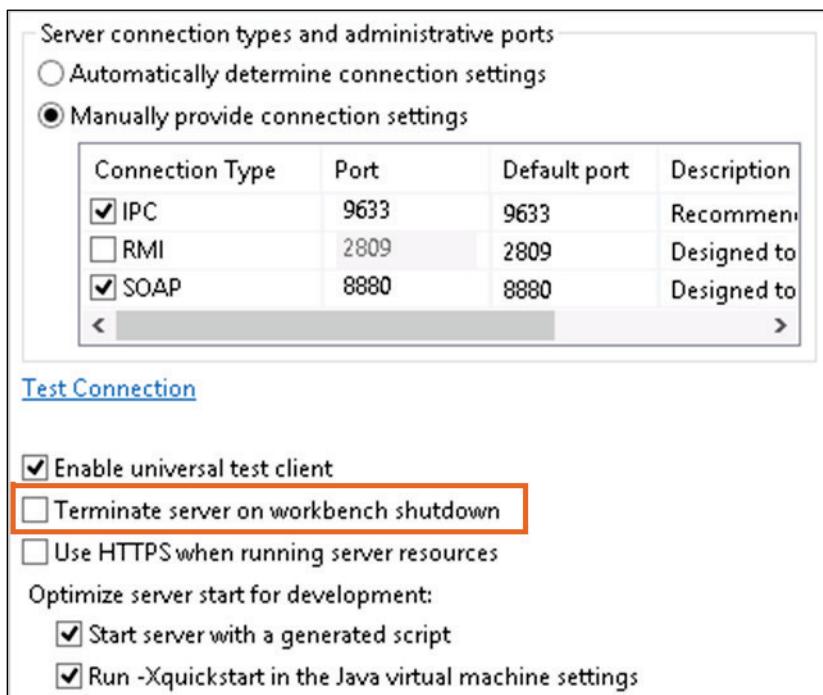
The IBM Process Server runtime that is installed in IBM Integration Designer has a default server profile. You can alter this profile, create profiles, and reset your profile back to the original state. Using profiles, you can deploy and test modules on a remote IBM Process Server or WebSphere Enterprise Server Bus runtime environment. For example, you might want to deploy your application to the preproduction environment of your organization. For more information about creating a server profile, see the product documentation.

### 4. View the configuration of the IBM Process Server v8.5 test runtime:

1. Switch to the **Servers** view.
2. Right-click **IBM Process Server v8.6 at localhost** and click **Open** from the menu.
3. Examine (but do not change) the server properties in the server configuration editor. In particular, note the following options.
  - **Automatically publish when resources change:** This option in the **Publishing** section automatically republishes modules to the local test server if you modify them while they are deployed.



- **Terminate server on workbench shutdown:** This option in the **Server** section stops the test server when you close IBM Integration Designer.



4. After examining the server properties, close the server configuration editor.

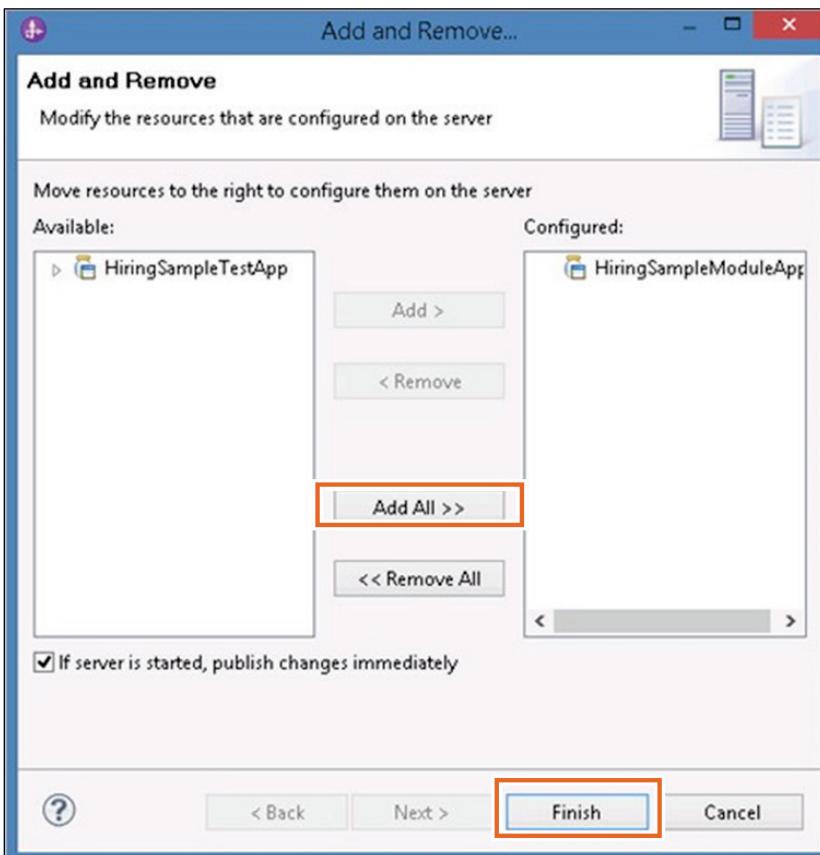
### Testing the hiring sample application

You test the hiring sample application by interacting with it through a web-based tool that is called the Business Process Choreographer Explorer. This tool is included with IBM Integration Designer for working with process applications and human tasks. To test the application, you must deploy the application to the IBM Process Server test environment and enter test data in the user interface.

5. Deploying the hiring sample application

To deploy the hiring sample application to the test server, you must first start IBM Process Server (if necessary). The server was started automatically in the previous section.

1. In the **Servers** view, verify that the server is started.
2. Right-click **IBM Process Server v8.6 at localhost** and click **Add and Remove**.
3. Click **Add All** and then click **Finish**.



Note: In IBM Integration Designer, you can add projects to the IBM Process Server runtime environment without first starting the server. If you do so, when the server is started, the projects are automatically deployed and started. Similarly, if you remove projects while the server is stopped, the projects are removed the next time that the server starts. These options are valid only if the test server is local.

Wait for the publishing action to complete and for the modules to start. Publishing is complete when no messages are in the status bar such as: Publishing to IBM Process Server v8.6 at localhost.

4. When the projects are deployed and started, switch to the **Servers** view, expand **IBM Process Server v8.6 at localhost**, and view the modules that are currently deployed to the server. Verify that the status of all the modules added is **Started**.

**Troubleshooting:** Occasionally, the status of the server and modules does not automatically refresh. It is possible that, while the server and modules are started, the status remains **Publishing**.

Press F5 to refresh the view.

Also, notice that often a module that is added has a **Stopped** status. To change its status, right-click the module, click **Restart** from the menu, and republish the module if prompted. Repeat the step if necessary to ensure that the module starts.

## Part 5. Opening the customer user interface

**Note** for BPM on Cloud users: Click **Launch** in the Business Process Choreographer Explorer tile to use this tool.

1. Open the Business Process Choreographer Explorer user interface and create an application.

1. In the Servers view, right-click the server and click **Launch > Business Process Choreographer Explorer**.

2. If you receive security alert messages, click **Yes** each time.

Note: Alternatively, you can open a browser and type:

`http://localhost:<portnumber>/bpc`

The port number can be found in the console output. For a local IBM Process Server instance configured during IBM Integration Designer product installation, the port number of the internal web container is typically 9080.

3. Use `admin` for the user name and `web1sphere` for the password to log in to the tool.

4. The page automatically begins at the **My To-dos** list.

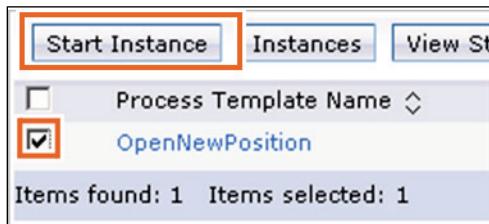
This list contains the human tasks that are waiting for user input. It is expected that no tasks are waiting.

Configurable menus on the left of the page work either with processes or with human tasks.

5. From the **Process Templates** menu, click the **Currently Valid** link.



6. Select the **OpenNewPosition** template and click **Start Instance**.



7. In the **Process Input Message** section, set the **Process Name** to: NewPositionTest
8. Click **Submit**.

A screenshot of a 'Submit' dialog box. At the top left is a 'Submit' button (highlighted with a red box). Below it, there are two sections: 'Process Template Name' (set to 'OpenNewPosition') and 'Process Description'. Further down are sections for 'Operation' (set to 'createPosition'), 'Process Name' (highlighted with a red box and containing the value 'NewPositionTest'), and 'Process Input Message'. At the bottom right of the dialog are 'Form' and 'View' buttons.

Note: You do not have to complete any other data. It is accomplished in a later step.

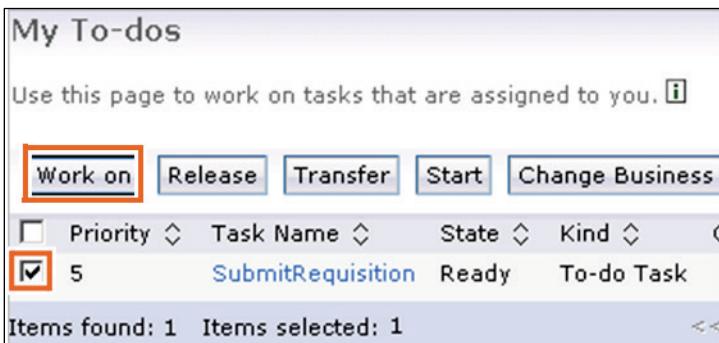
The first activity for the user is to complete the job requisition. This job requisition is implemented as a human task by the process application. The Business Process Choreographer Explorer works with processes and human tasks.

9. From the left side of the explorer, from the **Task Instances** menu, click **My To-dos**.



The instance of the process application stops at the human task: **SubmitRequisition**.

10. Select **SubmitRequisition** and then, click **Work on**.



The **Task Message** display is divided into two sections: the input message and the output message. The input message is empty; you did not enter any information for it in the previous step.

11. Type the following information for the output message:

- `outputPosition > positionType: New`

outputPosition	positionType	New
	replacement	- <a href="#">Add</a>
	jobTitle	<input type="text"/>
	iId	<input type="text"/>

12. Click **Complete**.

Expect to see that no tasks are waiting for you. If necessary, scroll up to locate the **Complete** button.

2. Examine the Server Logs view.

1. If you do not see any messages, click the **Load Server Console or Log** icon.



2. Select **Load from server console** and click **Finish**.
3. Examine the server log. You see a message that the requisition was automatically approved because the position was new.

Type	Time	Thread ID	Contents
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:16.204 EDT	00000271	WSVR0221I: Application started: HiringSampleModuleApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:20.204 EDT	00000056	WSVR0200I: Starting application: TestController70
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:20.532 EDT	00000056	WSVR0221I: Application started: TestController70
<input type="checkbox"/> Log message	Apr 15, 2016 14:01:55.720 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed
<input type="checkbox"/> Log message	Apr 15, 2016 14:04:47.673 EDT	0000024b	WSVR0200I: Starting application: HiringSampleTestApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:04:48.173 EDT	0000024b	WSVR0221I: Application started: HiringSampleTestApp
<input type="checkbox"/> Log message	Apr 15, 2016 14:10:59.986 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed

### Seeking manager approval for an existing job position

In the previous example, the job requisition was automatically approved because the position was listed as `New`. In this example, set the position to `Existing`, which forwards the job requisition to the general manager for approval.

4. Return to the Business Process Choreographer Explorer.
5. Select **Process Templates > Currently Valid**.
6. Select **OpenNewPosition** and click **Start Instance**.
7. Set the **Process Name** to `ExistingJobA` and click **Submit** to create an instance.

8. Select **OpenNewPosition** and click **Start Instance** again. Set the **Process Name** to **ExistingJobB** and click **Submit** to create a second instance with a different process ID.

In this way, you can test two samples at the same time: one that the general manager approves, and one that is rejected.

9. In the **Currently Valid Process Templates** view, select **OpenNewPosition** and click **Instances**.

Both of your instances are active and running.

The screenshot shows a user interface for managing process templates. At the top, there are buttons for Migrate, Terminate, Delete, Suspend, Resume, and Restart. Below these are dropdown menus for 'Process Instance Name' and 'Process Template Name'. Under 'Process Instance Name', 'ExistingJobB' and 'ExistingJobA' are listed. Under 'Process Template Name', 'OpenNewPosition' is listed twice. A status bar at the bottom indicates 'Items found: 2 Items selected: 0'.

10. Select **Task Instances > My To-dos**.

Both instances stop at the same activity:  
**SubmitRequisition**.

11. Select the first instance and click **Work on**.  
12. Set **outputPosition > positionType** to: **Existing**  
13. Click **Complete**.

The screenshot shows a configuration dialog for 'outputPosition'. It has four fields: 'positionType' (set to 'Existing'), 'replacement' (with a 'Add' button), 'jobTitle' (empty), and 'iID' (empty). At the bottom is an 'Edit Source' button.

14. Refresh the **MyTo-dos** page by clicking **Refresh**.

One process is still at the **SubmitRequisition** activity, but the one you just completed moves forward to **GMAApproval**.

The screenshot shows the 'MyTo-dos' page with a list of tasks. The tasks are: 'GMAApproval' (Priority 5, State Ready, Kind To-do Task) and 'SubmitRequisition' (Priority 5, State Ready, Kind To-do Task). At the bottom, it says 'Items found: 2 Items selected: 0'.

15. Select the task instance at the **SubmitRequisition** activity and click **Work on**.

16. Set outputPosition > positionType to: Existing

17. Click **Complete**.

When you refresh again, both tasks arrive at the **GMAccrual** activity.

18. Select the first task that is named **GMAccrual** and click **Work on**.

Because the position was listed as **Existing**, the **approvalNeeded** field is enabled.

19. In the output message, next to the **gmApproval** field, click **Add**.

A check box is added for you to toggle.

gmApproval	<input checked="" type="checkbox"/>	Remove
gmComments		
instanceId		

20. Leave the check box selected and click **Complete** to indicate that the general manager approved the job requisition.

21. Examine the **Server Logs** view.

A message indicates that the general manager approved the job requisition.

22. Return to the Business Process Choreographer Explorer. Select **My To-Dos**.

23. Select the remaining task and click **Work on**.

24. In the output message, next to the **gmApproval** field, click **Add**.

A check box is added for you to toggle.

25. Clear the **gmApproval** check box.

This action signals that the general manager denied the job requisition.

26. Click **Complete**.

27. Examine the **Server Logs** view again.

A message indicates that the general manager rejected the job requisition.

28. Log out of the Business Process Choreographer Explorer and close the browser.

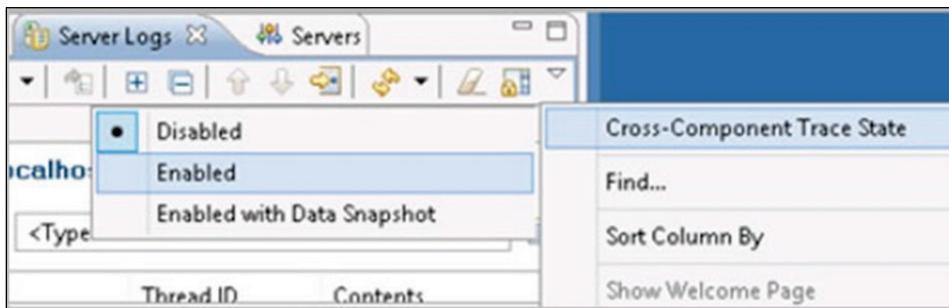
## Part 6. Use cross-component testing to examine a running process application.

Cross-component testing is a feature of the IBM Process Server test environment, which traces the individual components in a process application. Without the use of breakpoints or debugging, a developer can use cross-component tracing to follow the path of a running business process and examine its flow in the **Server Logs** view.

1. Open the **Server Logs** view. In the **Server Logs** toolbar, an icon is named **View Menu**.



2. Click **View Menu** and click **Cross-Component Trace State > Enabled**.



Information: The three settings for cross-component tracing are:

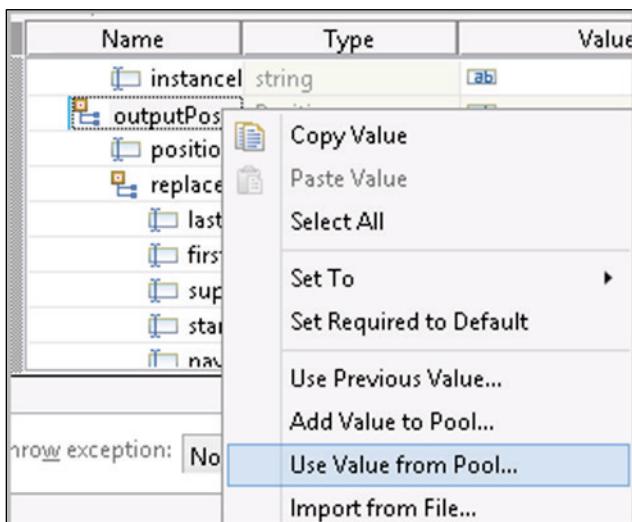
- **Disabled**: Tracing is not sent to the server log.
- **Enabled**: Tracing is sent to the server log.
- **Enabled with Data Snapshot**: Tracing is sent to the server log along with capturing data that is sent in and passed between components.

3. Clear the contents of the server log by clicking the **Clear Server Console** icon.



4. In the Business Integration view, expand **HiringSampleModule**.
5. Double-click **Assembly Diagram** to open it in the editor.
6. Right-click the **OpenNewPosition** business process and click **Test Component in Isolation**.
7. In the integration test client, click **Continue**.
8. If you are prompted to do so, select **IBM Process Server v8.6 at localhost** from the list of deployment locations, and click **Finish**.
9. If you are prompted to do so, enter `admin` as the User ID to and `websphere` as the password.
10. Click **OK**.

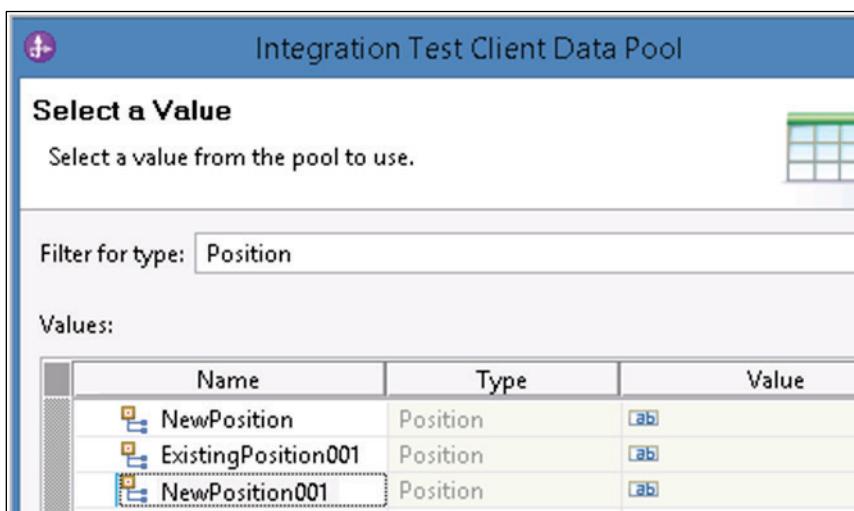
11. The process stops at emulating the **SubmitRequisition** human task.
12. Right-click the **outputPosition** parameter and click **Use Value from Pool**.



Important: Select the Position data object, not the Requisition data object.

Options are available for you to select from the pool of test data.

13. Select **NewPosition001** and click **OK**.



14. Click **Continue** when the sample data is populated.

The business process completes successfully.

## 15. Examine the **Server Logs** view.

You might choose to maximize the window so that it is easier to read.

Type	Time	Thread ID	Contents
Invocation sequence	Apr 15, 2016 14:52:43.632 EDT	00000247	
Start invoke (Op)	Apr 15, 2016 14:52:43.632 EDT	00000247	Start of the one-way invocation of operation OpenNewPos...
Start compo	Apr 15, 2016 14:52:43.710 EDT	00000059	Start of the component processing of operation OpenNew...
In BPEL ;	Apr 15, 2016 14:52:43.772 EDT	00000059	07e8a3dd-ddab-4640-be64-54345fc78d94 STATE 4fb376fb-...
Start BPEL	Apr 15, 2016 14:52:43.882 EDT	000002a2	Start of processing for BPEL process OpenNewPosition_Pl...
End BPEL	Apr 15, 2016 14:52:43.929 EDT	000002a2	End of processing for BPEL process OpenNewPosition_Pl:9...
Start BPEL	Apr 15, 2016 14:53:42.773 EDT	000002a2	Start of processing for BPEL process OpenNewPosition_Pl...
Log ;	Apr 15, 2016 14:53:42.788 EDT	000002a2	>>> Find Job Candidates --- No Approval Needed
End BPEL	Apr 15, 2016 14:53:42.819 EDT	000002a2	End of processing for BPEL process OpenNewPosition_Pl:9...
End compor	Apr 15, 2016 14:52:43.851 EDT	00000059	End of the component processing of operation OpenNewP...
End invoke (Op)	Apr 15, 2016 14:52:43.663 EDT	00000247	End of the one-way invocation of operation OpenNewPosi...

Cross-component tracing writes a trace to the Server Logs view, which displays the entire invocation sequence, from the time the SCA component was invoked at the top level to its BPEL implementation. Inside the BPEL process, you can follow how the BPEL process is calling each of the steps in turn. As each step completes, the invoking component that wraps it is closed in turn.

Hint: Cross-component tracing can be a valuable assistant when troubleshooting lab exercises. You might choose to leave cross-component tracing enabled for the labs that follow so that you can easily follow the flow of the business processes.

16. Close the integration test client. Do not save it.
17. In the **Servers view**, right-click **IBM Process Server v8.6 at localhost** and click **Add and Remove**.
18. Click **Remove All** and click **Finish** to remove the projects from the server.  
The status bar indicates when the publish (removal) action is complete. The projects are also removed from the **Servers view**.

## Part 7. Remove the applications from the server and stop the server.

1. In the Servers view, right-click IBM Process Server v8.6 at localhost and click Add and Remove from the menu.
2. Click **Remove All** and click **Finish** to remove the projects from the server.  
The status bar indicates when the publish (removal) action is complete. The projects are also removed from the Servers view.
3. Close any windows that might be open. Do not save the test trace.
4. Optionally, in the **Servers view**, right-click **IBM Process Server v8.6 at localhost** and click **Stop** from the menu.

5. Close IBM Integration Designer.

**Results**

**In this exercise, you assembled an SCA application and examined project components. You used the IBM Integration Designer Test environment to examine projects and run a test component.**

## **Unit 6**      SCA bindings

The slide features a blue header bar with 'IBM Training' on the left and the IBM logo on the right. The main content area has a light gray diagonal striped background. In the center, the text 'SCA bindings' is displayed in a large, bold, dark blue font. Below it, 'IBM Business Process Manager V8.6' is shown in a smaller, dark blue font. At the bottom of the slide, there is a copyright notice: '© Copyright IBM Corporation 2018' and 'Course materials may not be reproduced in whole or in part without the written permission of IBM.'

IBM Training

**SCA bindings**

IBM Business Process Manager V8.6

© Copyright IBM Corporation 2018  
Course materials may not be reproduced in whole or in part without the written permission of IBM.



## Unit objectives

- 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

SCA bindings

© Copyright IBM Corporation 2018

### *Unit objectives*

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.

## Topics

- SCA bindings
- Web services

SCA bindings

© Copyright IBM Corporation 2018

*Topics*

## SCA bindings

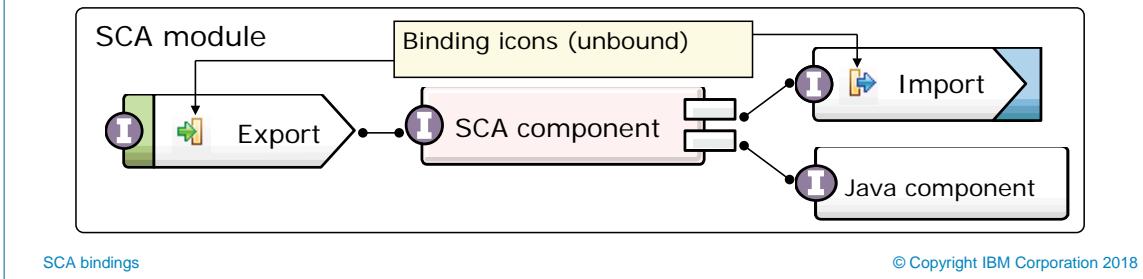
SCA bindings

© Copyright IBM Corporation 2018

*SCA bindings*

## 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)



*Incoming and outgoing interactions with external components*

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.

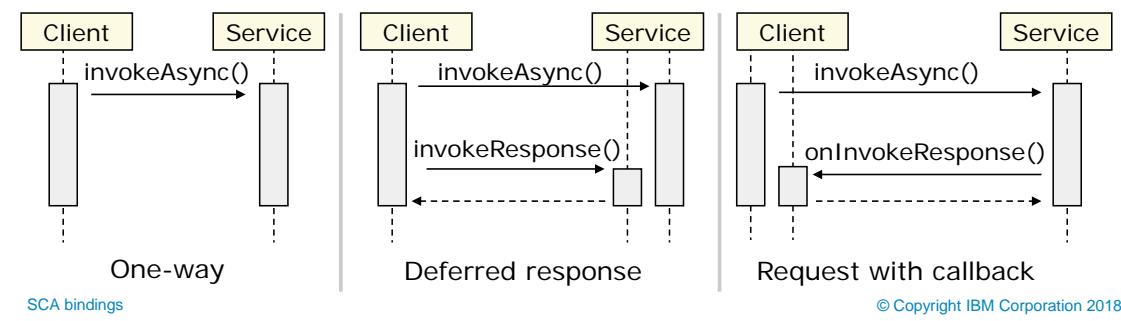
Imports and exports require binding information, which specifies the means of transporting data. Bindings for imports and exports have different purposes. An import binding describes the specific way that an external service is bound to an import component. An export binding describes how that export (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



### SCA invocation styles

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

Export: AccountVerificationExport1 (JMS Binding)

Description	Adapter type:	WebSphere Default Messaging Provider version 0.3
Details	Function selector:	* <a href="#">One Operation Function Selector</a>
Binding	Default data format:	UTF8XMLDataHandler
End-Point Configuration	Default message type:	Byte
Method Bindings		

SCA bindings © Copyright IBM Corporation 2018

### Import and export resources

IBM Process Server provides a technology that is called `DataBinding`. `DataBinding` is a Java interface that is called at run time in IBM Process Server. When IBM Process Server receives a message through an export component, the 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

SCA bindings

© Copyright IBM Corporation 2018

### Quality of service qualifiers

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

SCA bindings

© Copyright IBM Corporation 2018

### *Export binding types*

When an export component is added to the SCA assembly diagram, it must be bound to a particular transport binding. In the following pages, you examine each of the export binding types and options in detail. 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

SCA bindings

© Copyright IBM Corporation 2018

### *Import binding types*

An import component must also be bound to a particular transport binding. The import hides the mechanics of how the service call is implemented. All the caller sees is an exposed service interface. The actual network or invocation protocol is hidden. Import bindings are covered later in this unit. 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.

### *JCA adapter EIS bindings*

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.

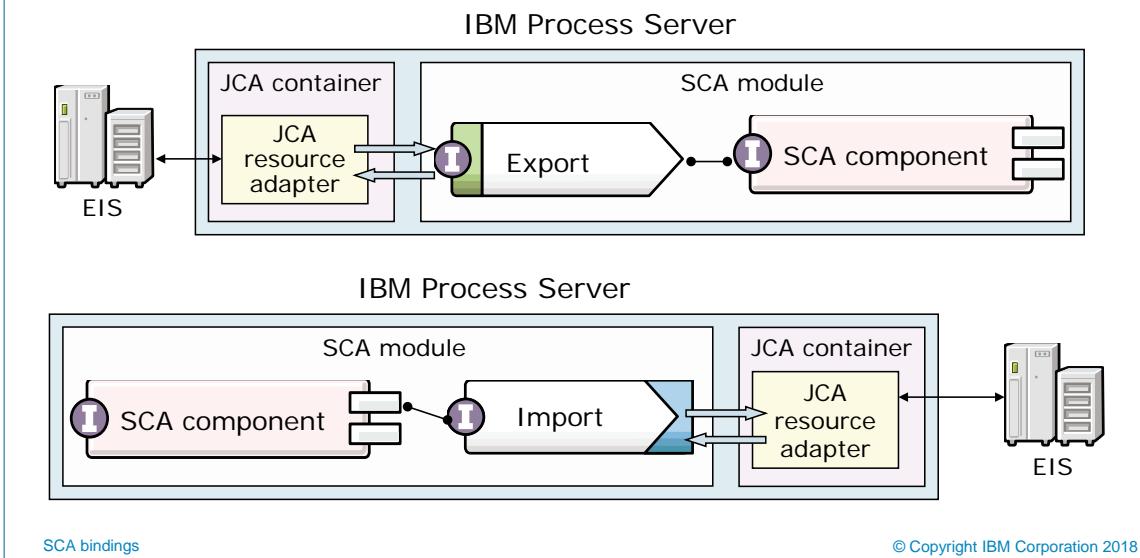
Your SCA components might require that data be transferred to or from an external EIS. When you create an SCA module that requires such connectivity, you include (in addition to your SCA component) an import or export with an EIS binding for communication with a specific external EIS.

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



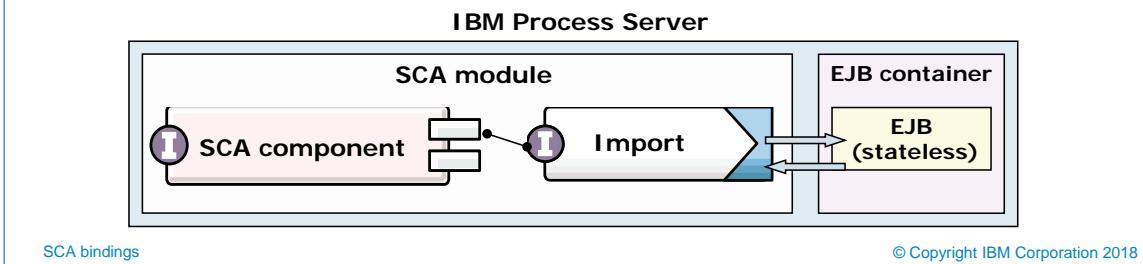
When an adapter is used, the module that contains that adapter has an export component that is bound to the adapter. The WebSphere Application Server JCA container that manages the instance of the adapter is cognizant of an external system update or request.

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

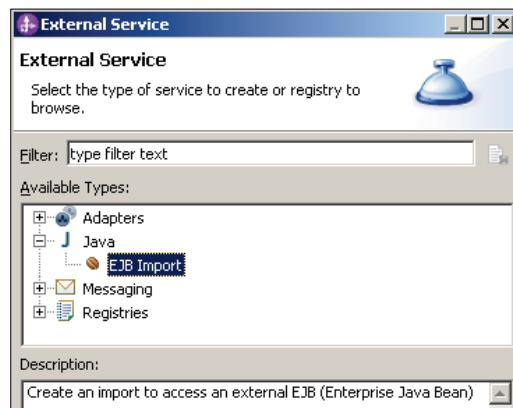


### *Enterprise JavaBeans (EJB) binding*

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

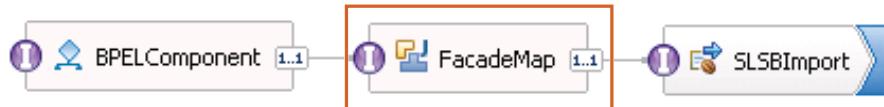


In IBM Integration Designer, you can drag a stateless session bean descriptor onto the SCA assembly diagram. This action creates an import component that, when called, calls the EJB bean.

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



SCA bindings

© Copyright IBM Corporation 2018

### WSDL to Java interface mapping for EJB beans

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

SCA bindings

© Copyright IBM Corporation 2018

### *SCA bindings*

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.