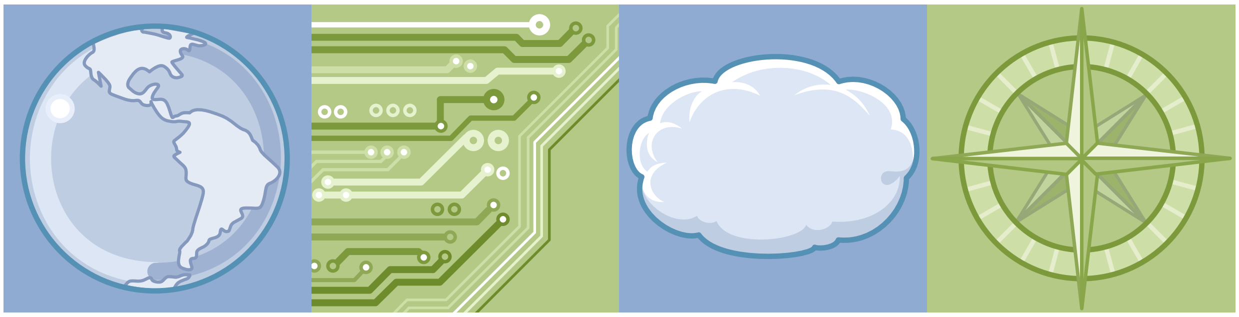
IBM Training

Student Notebook

IBM Integration Bus V10 System Administration

Course code WM646 ERC 1.0



IBM Systems
Middleware

Trademarks

December 2015 edition

Contents

Trademarks

Course description

IBM Integration Bus V10 System Administration

Duration: 5 days

Purpose

This course gives administrators training on IBM Integration Bus system administration. It is also relevant for IBM Integration Bus developers who also work in an administrative capacity. In this course, you learn how to administer IBM Integration Bus on distributed operating systems, such as Windows and AIX, by using the IBM Integration Bus administrative interfaces. These interfaces include the IBM Integration web user interface and the IBM Integration Bus commands. The course begins with an overview of IBM Integration Bus. Subsequent topics include initial product installation and maintenance, environment configuration, and basic administrative tasks such as backing up and recovering the environment. You learn about product requirements, securing access to IBM Integration Bus resources and message flows, problem determination and resolution, and performance monitoring and tuning. You also learn techniques for extending the capabilities of IBM Integration Bus. The course also covers the publish/subscribe model and reviews the Java Message Service (JMS) transport protocol and web services. In the course lab exercises, you gain hands-on experience with IBM Integration Bus administrative tasks such as managing IBM Integration Bus runtime components, monitoring message flow applications, and configuring security.

Audience

This course is designed for IBM Integration Bus administrators and developers who administer IBM Integration Bus.

Prerequisites

Before taking this course, you should have hands-on experience with the Windows operating system, and be familiar with basic security practices and protocols such as Secure Socket Layer (SSL) and Lightweight Directory Access Protocol (LDAP). Experience with IBM Integration Bus message flow development and IBM MQ system administration is helpful, but not required.

Objectives

After completing this course, you should be able to:

- Install and configure an IBM Integration Bus instance

- Establish, maintain, and manage an integration node
- Administer IBM Integration Bus components and message flow applications by using the IBM Integration web user interface and command interface
- Configure connectivity to IBM MQ to enable IBM Integration Bus to get messages from, or put messages to, queues on a local or remote queue manager
- Implement IBM Integration Bus administration and message flow security
- Use problem determination aids to diagnose and solve development and runtime errors
- Use the IBM Integration web user interface to generate and display message flow statistics
- Use IBM MQ or MQTT to publish and subscribe to IBM Integration Bus topics
- Implement an IBM Integration Bus global cache to store, reuse, and share data between integration nodes
- Use workload management policies to adjust the processing speed of messages and control the actions that are taken on unresponsive flows and threads
- Use the IBM Integration web user interface and a database to record events and replay messages
- Enable an integration node to connect to a database with ODBC and JDBC
- Configure a Java Message Services (JMS) provider for use with the JMS nodes
- Configure IBM Integration Bus for the secure file transfer protocol (SFTP)
- Find and install IBM Integration Bus SupportPac components

Curriculum relationship

Some knowledge of IBM MQ components and concepts is helpful for some functions. IBM MQ courses include WM101, *Technical Introduction to IBM MQ*, WM207/ZM207, WM207, *IBM MQ V8 System Administration (Windows Labs)*, and WM209, *IBM MQ V8 System Administration (Linux Labs).*

Agenda

Day 1

Course introduction
Unit 1. IBM Integration Bus overview
Unit 2. Product installation, configuration, and security planning
Exercise 1. Integration node setup and customization
Unit 3. Connecting to IBM MQ
Exercise 2. Connecting to IBM MQ

Day 2

Unit 4. Administration in the IBM Integration Toolkit
Exercise 3. Using the IBM Integration Toolkit
Unit 5. Administration basics
Exercise 4. Administering the IBM Integration Bus runtime components
Unit 6. Implementing IBM Integration Bus administration security

Day 3

Exercise 5. Using file-based security to control administration access
Exercise 6. Using queue-based security to control administration access
Unit 7. Implementing IBM Integration Bus message flow security
Unit 8. Administering web services and web service security
Exercise 7. Implementing web services and web services security

Day 4

Unit 9. Diagnosing problems
Exercise 8. Using problem diagnosis tools
Exercise 9. Identifying runtime problems
Unit 10. Monitoring the integration node and message flow performance
Unit 11. Publish/subscribe implementation overview
Exercise 10. Viewing runtime statistics

Day 5

Unit 12. Configuring IBM Integration Bus for high availability
Exercise 11. Administering workload management policies
Unit 13. Monitoring, recording, and replaying message flow events
Exercise 12. Recording and replaying message flow data
Unit 14. Extending IBM Integration Bus
Unit 15. Course summary

Unit 1. IBM Integration Bus overview

What this unit is about

This unit introduces the features and components of IBM Integration Bus. You also learn about IBM Integration Bus application connectivity.

What you should be able to do

After completing this unit, you should be able to:

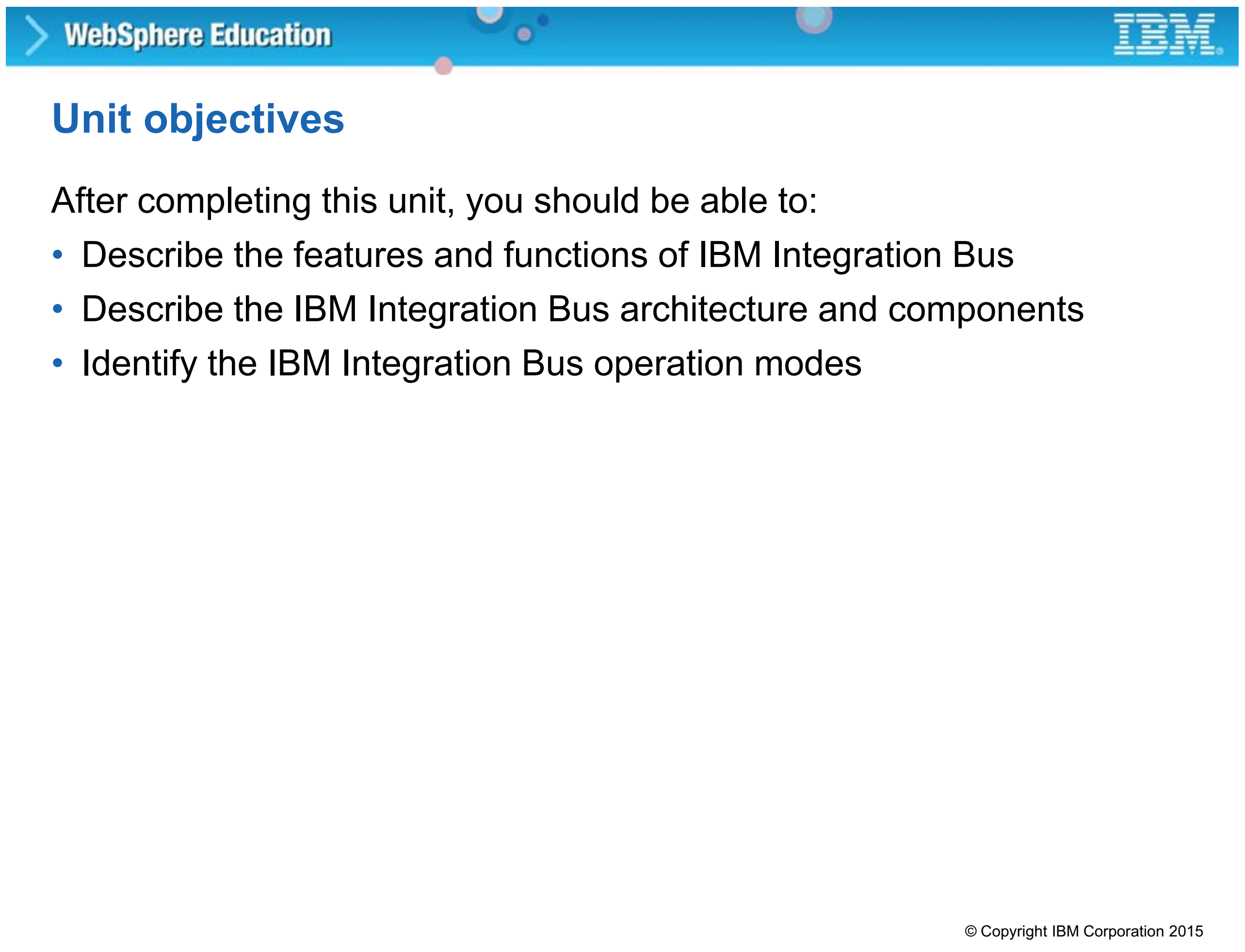
- Describe the features and functions of IBM Integration Bus
- Describe the IBM Integration Bus architecture and components
- Identify the IBM Integration Bus operation modes

How you will check your progress

- Checkpoint questions

References

IBM Knowledge Center



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the features and functions of IBM Integration Bus
- Describe the IBM Integration Bus architecture and components
- Identify the IBM Integration Bus operation modes

Figure 1-1. Unit objectives WM6461.0

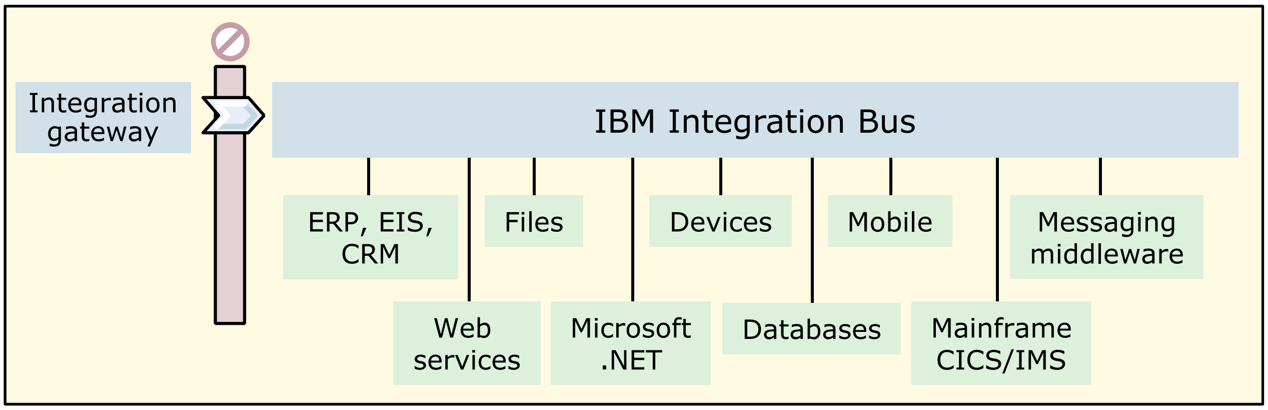
Notes:

This unit provides an overview to IBM Integration Bus V10. Each of the concepts and components that are introduced in this unit are described in more detail in later units.

1.1. IBM Integration Bus features and functions

WebSphere Education IBM

IBM Integration Bus



- IBM Integration Bus provides connectivity across enterprise systems, applications, and data
  - Avoids rewrites in response to new integration requirements
  - Simplifies maintenance by reducing expensive coupling
  - Provides flexibility, which adds anonymity between data producers and consumers
  - Adds insight into applications and the business value that they bring

Figure 1-2. IBM Integration Bus WM6461.0

Notes:

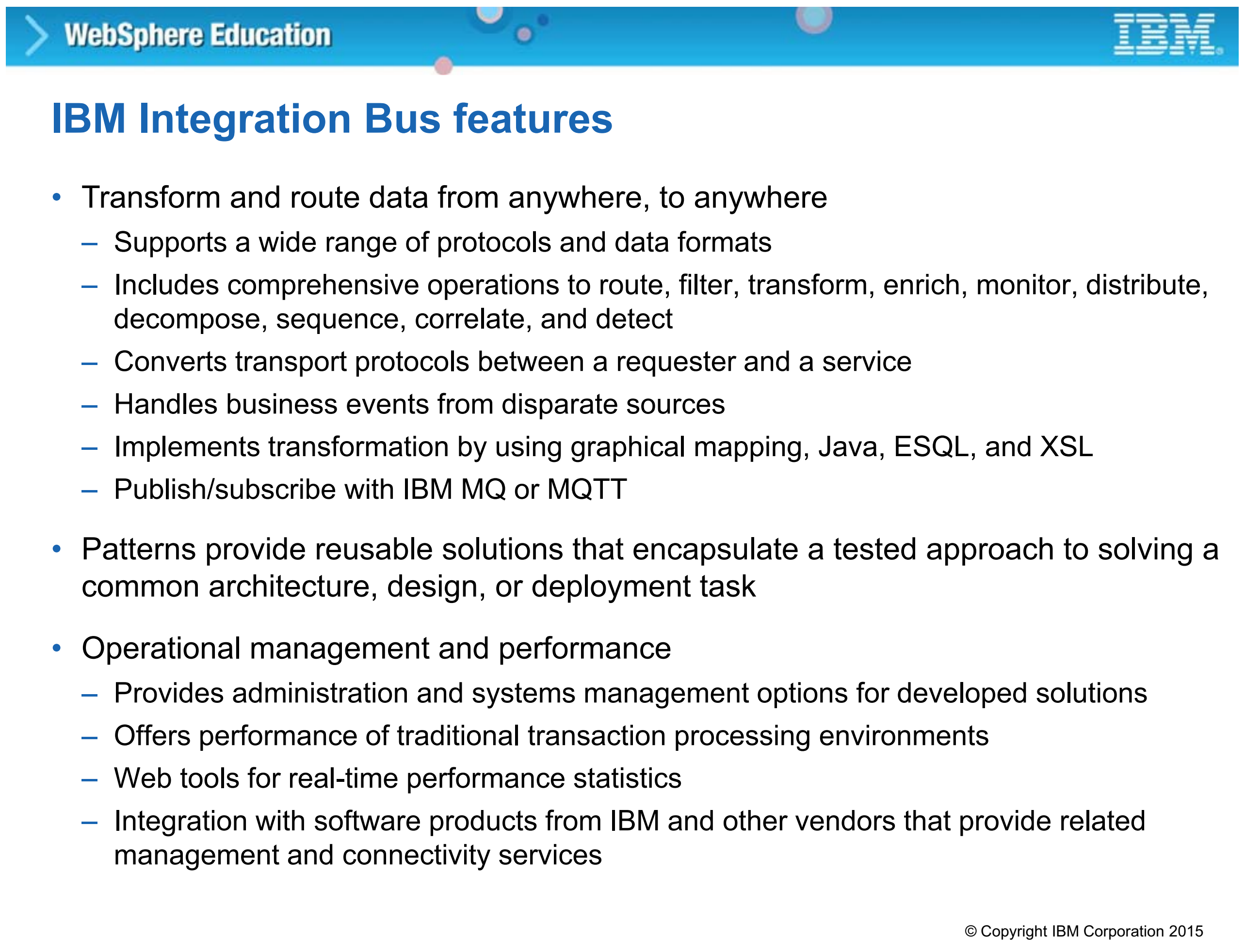
Enterprise systems consist of many logical endpoints such as off-the-shelf applications, services, packaged applications, web applications, devices, appliances, and custom built software.

Endpoints expose a set of inputs and outputs. Inputs and outputs include messaging middleware such as IBM MQ and JMS, protocols such as TCP/IP, HTTP, FTP, SMTP, and POP3, and databases such as DB2 and SQL Server. Endpoints also include formats such as flat files, COBOL, XML, industry standards (SWIFT, EDI, HL7), and user-defined data.

Integration Bus connects these endpoints in meaningful ways.

- Integration Bus avoids rewrites in response to new integration requirements
- Integration Bus simplifies maintenance by reducing expensive coupling
- Integration Bus provides flexibility that adds anonymity between producers and consumers of data
- Integration Bus adds insight into applications and business value they bring

IBM Integration Bus represents IBM's strategic ESB offering. IBM Integration Bus provides tools that helps convert WebSphere Enterprise Service Bus assets and so they can run on IBM Integration Bus.



WebSphere Education

IBM Integration Bus features

- Transform and route data from anywhere, to anywhere
  - Supports a wide range of protocols and data formats
  - Includes comprehensive operations to route, filter, transform, enrich, monitor, distribute, decompose, sequence, correlate, and detect
  - Converts transport protocols between a requester and a service
  - Handles business events from disparate sources
  - Implements transformation by using graphical mapping, Java, ESQL, and XSL
  - Publish/subscribe with IBM MQ or MQTT
- Patterns provide reusable solutions that encapsulate a tested approach to solving a common architecture, design, or deployment task
- Operational management and performance
  - Provides administration and systems management options for developed solutions
  - Offers performance of traditional transaction processing environments
  - Web tools for real-time performance statistics
  - Integration with software products from IBM and other vendors that provide related management and connectivity services

© Copyright IBM Corporation 2015

Figure 1-3. IBM Integration Bus features WM6461.0

Notes:

IBM Integration Bus simplifies application connectivity, routes, and transforms messages, simplifies programming, and provides operational management and performance tools.

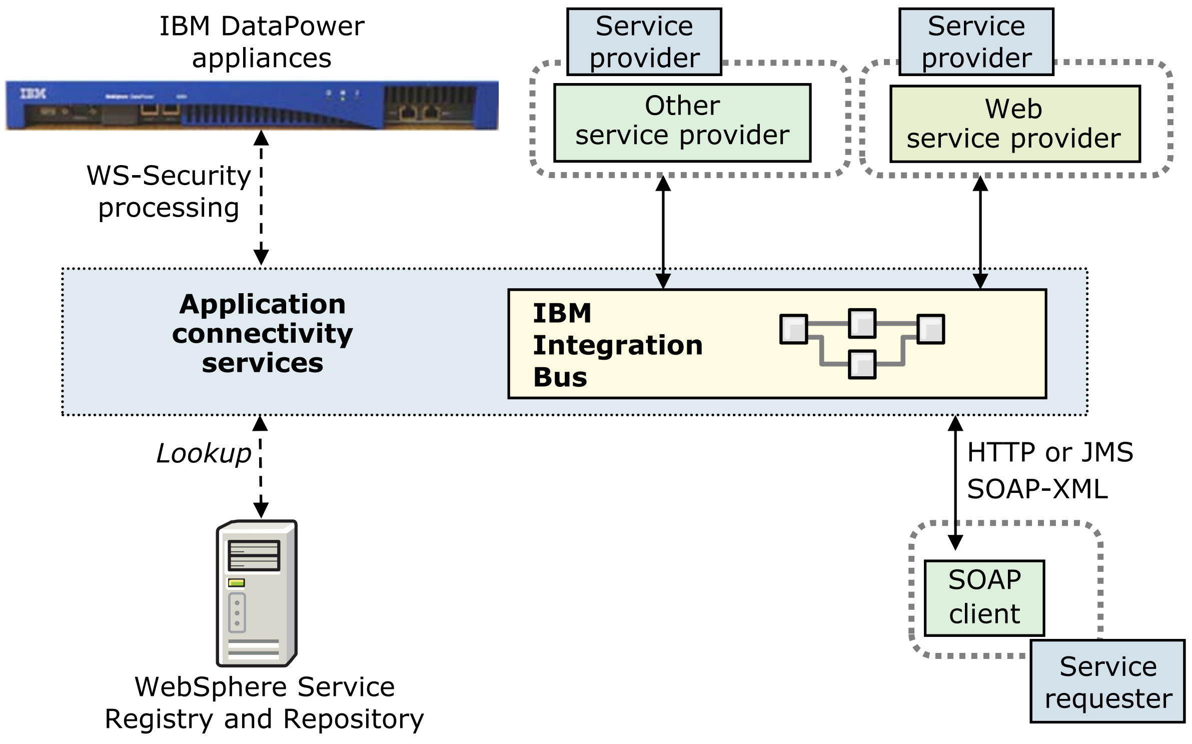
IBM Integration Bus includes features for universal connectivity to provide a flexible and dynamic infrastructure. These features include tools to help with the conversion of WebSphere Enterprise Service Bus assets so they can run on IBM Integration Bus.

IBM Integration Bus includes comprehensive operations and supports a wide range of protocols so that messages can be transformed and routed from anywhere, to anywhere. Predefined development patterns provide reusable solutions.

IBM Integration Bus has extensive administration and systems management facilities for real-time performance statistics.

WebSphere Education

IBM Integration Bus web services



© Copyright IBM Corporation 2015

Figure 1-4. IBM Integration Bus web services WM6461.0

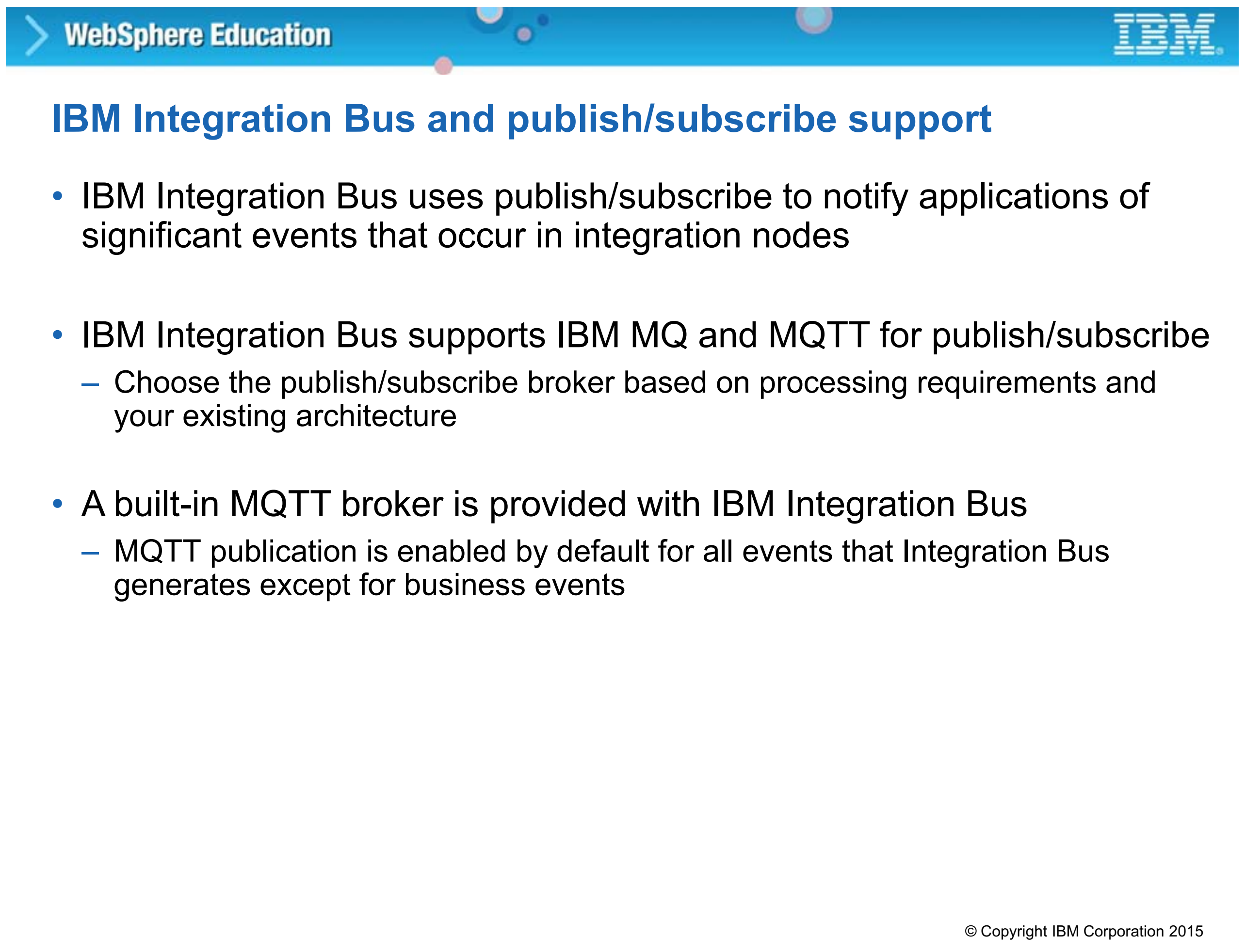
Notes:

Integration Bus can be used for connectivity to web services and web service appliances such as IBM DataPower appliances.

As shown in the figure, SOAP messages contain the data that is passed between the web service provider and the requester. SOAP messages are expressed in XML and can be delivered over HTTP or JMS transport. Integration Bus supports both protocols. So, in a message flow, web services can be requested and aggregated. A message flow can be deployed as a web service, listening to incoming SOAP requests on HTTP or JMS.

IBM Integration Bus can use the IBM DataPower appliance to handle its web services security (WS-Security) processing by providing HTTP/HTTPS encryption and decryption. DataPower SOA appliances are purpose-built, easy-to-deploy network appliances that help simplify, secure, and accelerate your XML and web service deployments while extending your SOA infrastructure. They can do XML schema validation, XSL transformations (XSLT), web service security, transformations between disparate message formats, and more.

Web services can be looked up dynamically in an external services repository such as IBM WebSphere Service Registry and Repository.



WebSphere Education

IBM Integration Bus and publish/subscribe support

- IBM Integration Bus uses publish/subscribe to notify applications of significant events that occur in integration nodes
- IBM Integration Bus supports IBM MQ and MQTT for publish/subscribe
  - Choose the publish/subscribe broker based on processing requirements and your existing architecture
- A built-in MQTT broker is provided with IBM Integration Bus
  - MQTT publication is enabled by default for all events that Integration Bus generates except for business events

© Copyright IBM Corporation 2015

Figure 1-5. IBM Integration Bus and publish/subscribe support WM6461.0

Notes:

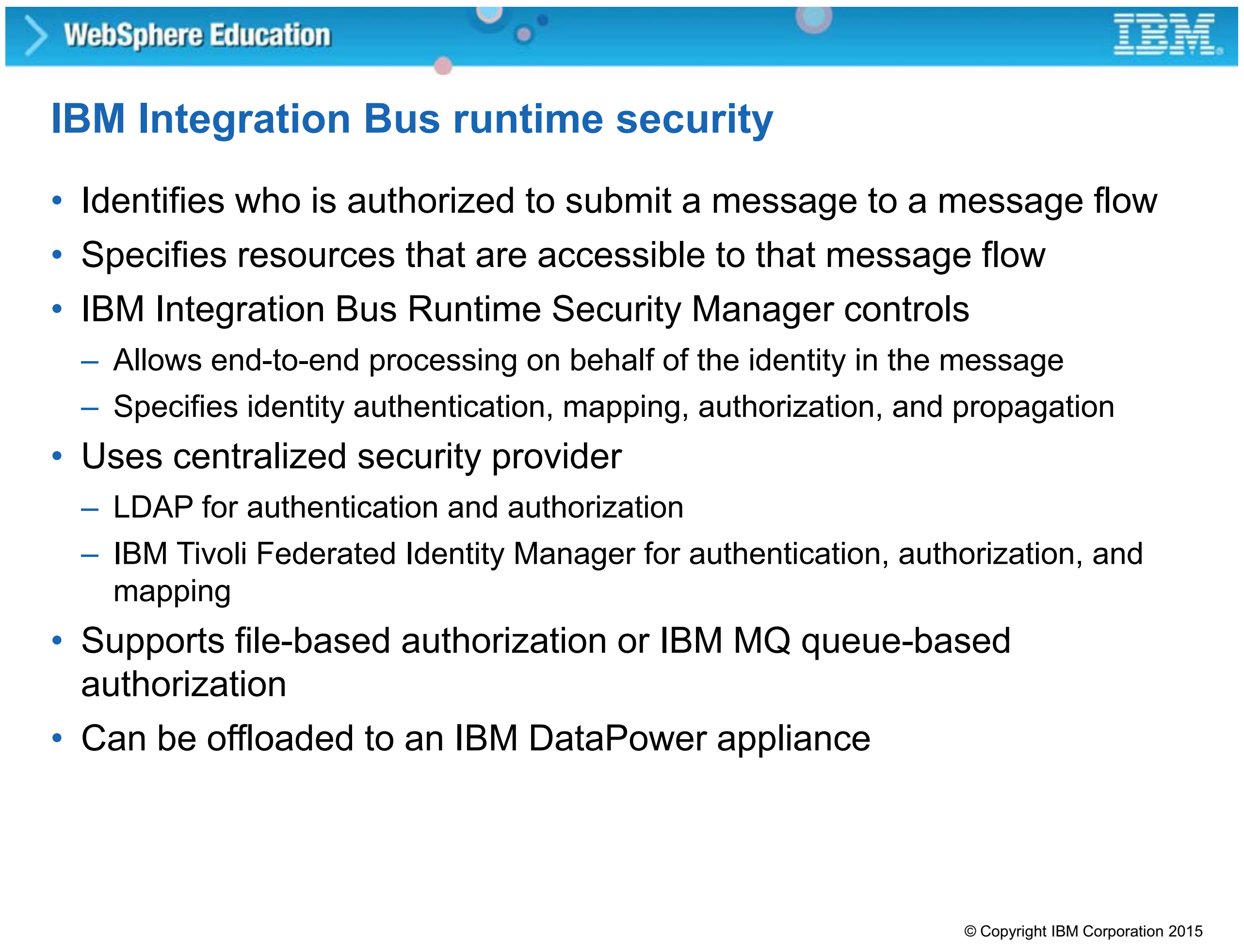
IBM Integration Bus supports two types of publish/subscribe broker: MQTT and IBM MQ. You can choose which type to use based on your processing requirements and your existing architecture.

A built-in MQTT broker is provided with IBM Integration Bus, and MQTT publication is enabled by default for all events that the integration node generates, except for business events. You can specify an alternative MQTT broker if you choose not to use the built-in MQTT broker.

If you installed IBM MQ, you can use the IBM MQ publish/subscribe broker that is provided with IBM MQ.

There are two situations where you can use publish/subscribe in an IBM Integration Bus message flow:

- To provide more transformation or routing function, or both, at publication time.
- To filter messages based on the content of the body of the message.



WebSphere Education

IBM Integration Bus runtime security

- Identifies who is authorized to submit a message to a message flow
- Specifies resources that are accessible to that message flow
- IBM Integration Bus Runtime Security Manager controls
  - Allows end-to-end processing on behalf of the identity in the message
  - Specifies identity authentication, mapping, authorization, and propagation
- Uses centralized security provider
  - LDAP for authentication and authorization
  - IBM Tivoli Federated Identity Manager for authentication, authorization, and mapping
- Supports file-based authorization or IBM MQ queue-based authorization
- Can be offloaded to an IBM DataPower appliance

Figure 1-6. IBM Integration Bus runtime security WM6461.0

Notes:

Security is an important consideration for both developers of IBM Integration Bus applications, and for system administrators that are configuring IBM Integration Bus authorities.

When a message arrives at an input node, a security profile indicates whether runtime security is configured. The security manager of the integration node is called to read the security profile. The security profile specifies the combination of authentication, authorization, and mapping that is done with the identity of the message, and by what external security provider.

The security manager can control access to message flows on a per-message basis by using the identity of the message. The security manager can:

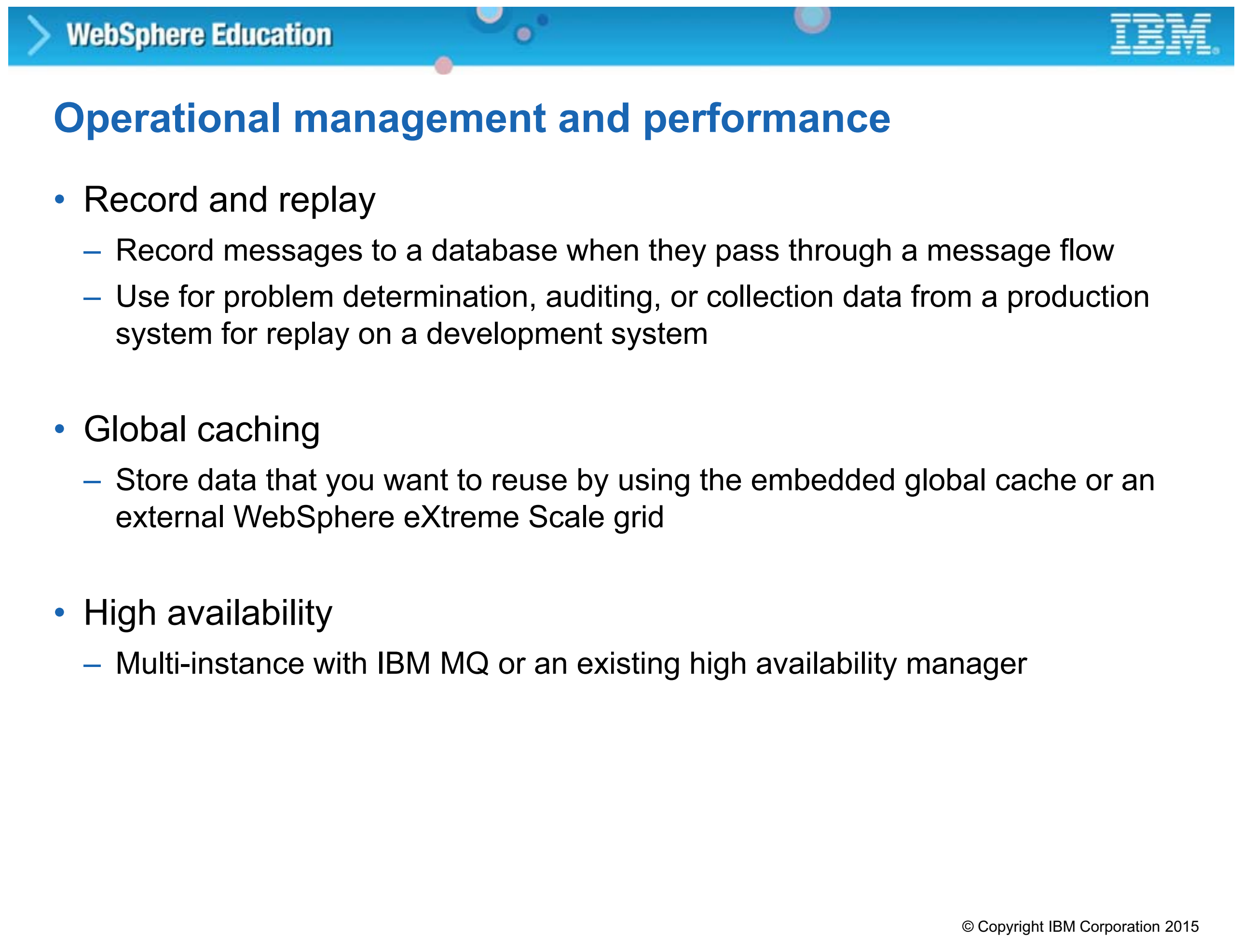
- Extract the identity from an inbound message.
- Authenticate it (using an external security provider).
- Map the identity to an alternative identity (by using an external security provider).
- Check that the alternative identity or the original identity is authorized to access the message flow (by using an external security provider).

- Propagate the alternative identity or the original identity with an outbound message. The actions to take for a message flow are controlled by using new security profiles. The integration node administrator creates security profiles; the security manager accesses them at run time.

Two external security providers are supported so that the Integration Bus can participate in a centralized security framework: LDAP for authentication and authorization and Tivoli Federated Identity Manager for authentication, mapping, and authorization.

Administration security can be implemented by using file-based authorization. If Integration Bus is configured connect to an IBM MQ queue manager, administration security can be implemented by using special system queues and queue-based authorization.

As an option, an IBM DataPower appliance can provide security.



WebSphere Education

Operational management and performance

- Record and replay
  - Record messages to a database when they pass through a message flow
  - Use for problem determination, auditing, or collection data from a production system for replay on a development system
- Global caching
  - Store data that you want to reuse by using the embedded global cache or an external WebSphere eXtreme Scale grid
- High availability
  - Multi-instance with IBM MQ or an existing high availability manager

Figure 1-7. Operational management and performance WM6461.0

Notes:

IBM Integration Bus includes many features for operational management, such as record and replay, and performance such as global caching.

For audit or problem determination purposes, you can record data to a database, view it, and replay it. The messages are stored in a database, which can be an Oracle, Microsoft SQL Server, or DB2. After you record the data, you can view it by using the web interface. You can replay a message to an IBM MQ queue by using the web interface or IBM Integration API. You can also replay the message to another message flow for testing or problem determination.

IBM Integration Bus contains an embedded global cache to store data that you want to reuse. As an option you can WebSphere eXtreme Scale for the cache.

With IBM Integration Bus, you can integrate Integration Bus with an IBM MQ multi-instance queue manager in a high availability IBM MQ configuration.

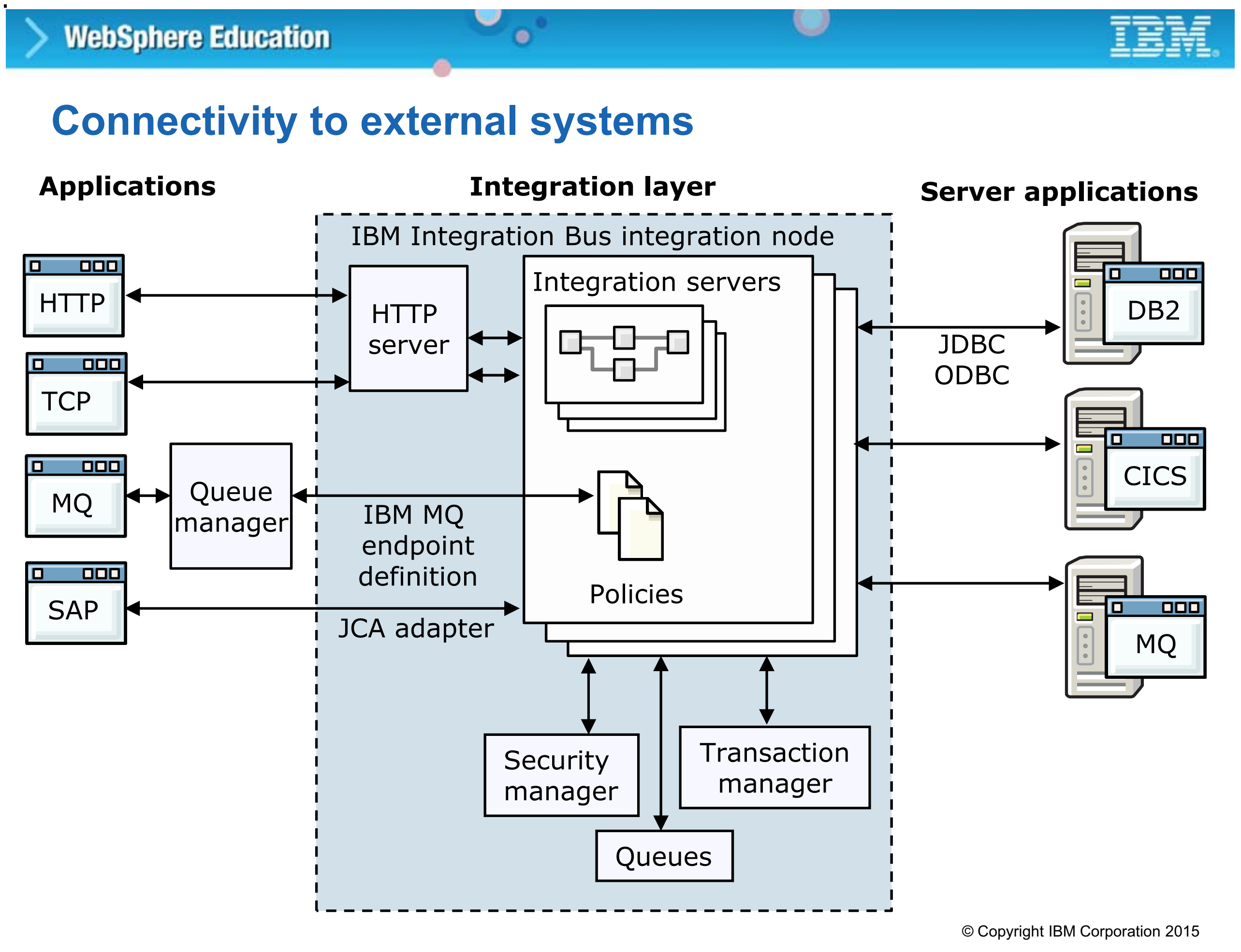


Figure 1-8. Connectivity to external systems WM6461.0

Notes:

IBM Integration Bus integrates many different products; you can also benefit from extra features by using the appropriate product or system with the product. The functions and features that you add by using supplementary products does not have an impact on your applications, but grants more options in the integration layer, which represents the operations of IBM Integration Bus.

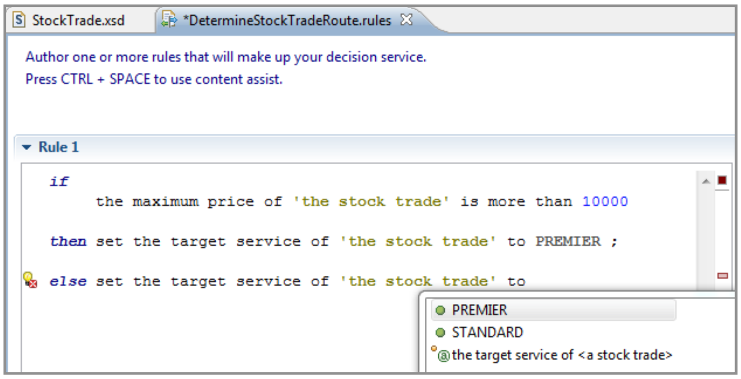
There are many different ways to connect into an IBM Integration Bus flow. Some of these connection methods include IBM MQ, JMS, HTTP, and web services.

In IBM Integration Bus, objects are defined in terms of message flows and message models. IBM Integration Bus has message flow nodes that are designed to allow interoperability with other IBM products such as WebSphere Process Server. So, for example, WebSphere Process Server can recognize the integration nodes as service component architecture (SCA) component by using specific message flow processing nodes. For example, if a message is received from a queue and the arrival of the message must drive a business process, an SCA input message processing node can be used to drive business process.

WebSphere Education

Decision management in IBM Integration Bus

- Provides business insight during integration data flows
  - Identifies inputs to business rules from in-flight data
  - Starts built-in rule engine to run business logic
  - Captures rules output for downstream processing


StockTrade.xsd | *DetermineStockTradeRoute.rules

Author one or more rules that will make up your decision service.
Press CTRL + SPACE to use content assist.

Rule 1

```
if
    the maximum price of 'the stock trade' is more than 10000
then set the target service of 'the stock trade' to PREMIER ;
else set the target service of 'the stock trade' to
```

- User can create rules in IBM Integration Toolkit or optionally refer to business rules on external IBM Operational Decision Manager decision server
- Embedded rules engine acts on business rules in the flow
  - Rule runs in the same operating system process as integration data flow
  - Rule update notification ensures consistent rule execution

Figure 1-9. Decision management in IBM Integration Bus WM6461.0

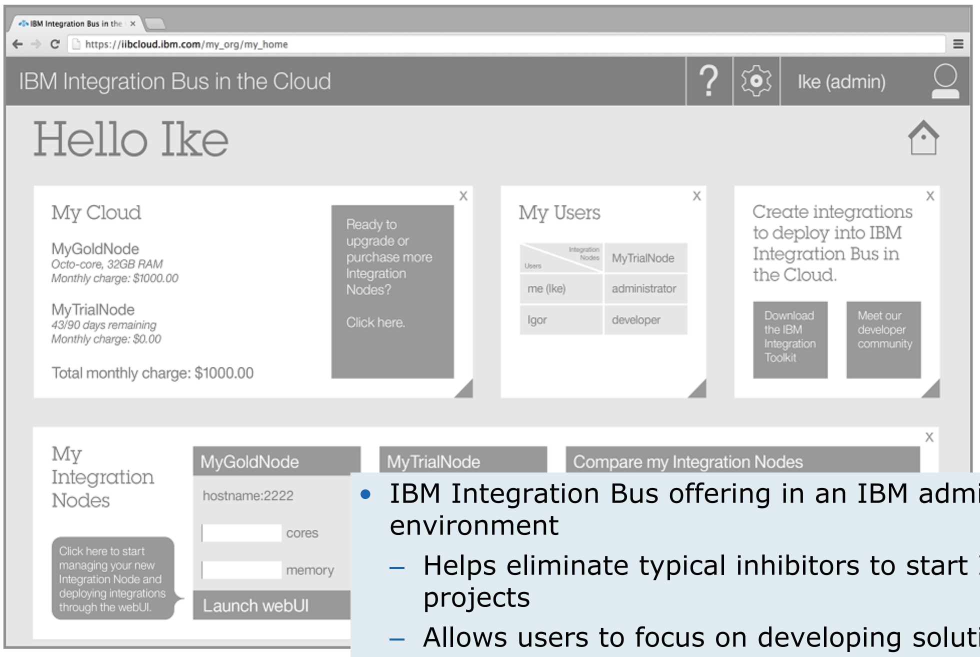
Notes:

The DecisionService node in the Integration Toolkit allows Integration Bus to call business rules that run on a component of IBM Decision Server.

The Integration Bus installation includes IBM Decision Server. The IBM Integration Bus license entitles you to use this component only through the DecisionService node and only for development and functional test. To use the IBM Decision Server component beyond development and functional test, you must purchase a separate license entitlement for either IBM Decision Server or IBM Decision Server Rules Edition for Integration Bus.

WebSphere Education

IBM Integration Bus in the Cloud


https://iibcloud.ibm.com/my_org/my_home
IBM Integration Bus in the Cloud
Ike (admin)
Hello Ike
My Cloud
MyGoldNode
Octo-core, 32GB RAM
Monthly charge: $1000.00
MyTrialNode
43/90 days remaining
Monthly charge: $0.00
Total monthly charge: $1000.00
Ready to upgrade or purchase more Integration Nodes?
Click here.
My Users
Users
Integration Nodes
MyTrialNode
me (Ike)
administrator
Igor
developer
Create integrations to deploy into IBM Integration Bus in the Cloud.
Download the IBM Integration Toolkit
Meet our developer community
My Integration Nodes
Click here to start managing your new Integration Node and deploying integrations through the webUI.
MyGoldNode
MyTrialNode
Compare my Integration Nodes
hostname:2222
cores
memory
Launch webUI

- IBM Integration Bus offering in an IBM administered cloud environment
  - Helps eliminate typical inhibitors to start Integration Bus projects
  - Allows users to focus on developing solutions
  - Within the constraints of a cloud environment, developers can use the same development tools for both cloud and on-premises software and assets that are generated can be deployed to either environment

Figure 1-10. IBM Integration Bus in the Cloud WM6461.0

Notes:

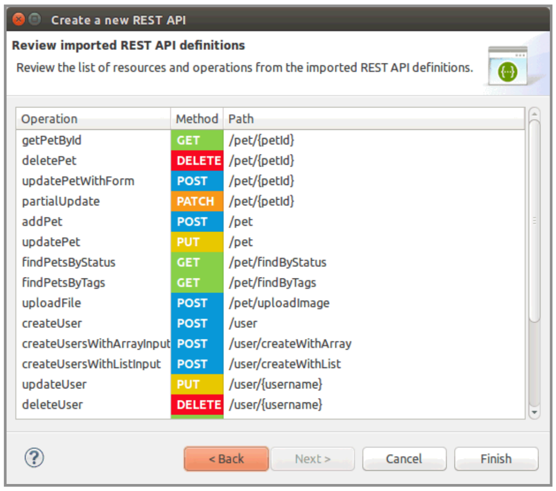
IBM Integration Bus in the Cloud helps eliminate typical inhibitors to start Integration Bus projects, such as capital expenditures, hardware availability, and the skills for managing an Integration Bus environment. It also allows users to focus on developing solutions rather than installing, configuring, and managing software.

Within the constraints of a cloud environment, developers can use the same development tools for both cloud and on-premises software, and the assets that are generated can be deployed to either environment.

WebSphere Education

Using IBM Integration Bus to provide a REST API

- Provides a way to receive JSON over HTTP/HTTPS and expose a REST API
  - Create a REST API in the IBM Integration Toolkit
  - Administer REST APIs as an IBM Integration Bus construct in the IBM Integration web user interface
- IBM Integration Toolkit REST API project
  - Import Swagger V2.0 JSON file to create the REST API project in the IBM Integration Toolkit
  - Defines a metadata format that is based on JSON schema to describe the REST APIs, their parameters, and the messages that are exchanged
  - Can use Swagger user interface to test and generate client code bindings



Review imported REST API definitions

Review the list of resources and operations from the imported REST API definitions.

Operation	Method	Path
getPetById	GET	/pet/{petId}
deletePet	DELETE	/pet/{petId}
updatePetWithForm	POST	/pet/{petId}
partialUpdate	PATCH	/pet/{petId}
addPet	POST	/pet
updatePet	PUT	/pet
findPetsByStatus	GET	/pet/findByStatus
findPetsByTags	GET	/pet/findByTags
uploadFile	POST	/pet/uploadImage
createUser	POST	/user
createUsersWithArrayInput	POST	/user/createWithArray
createUsersWithListInput	POST	/user/createWithList
updateUser	PUT	/user/{username}
deleteUser	DELETE	/user/{username}

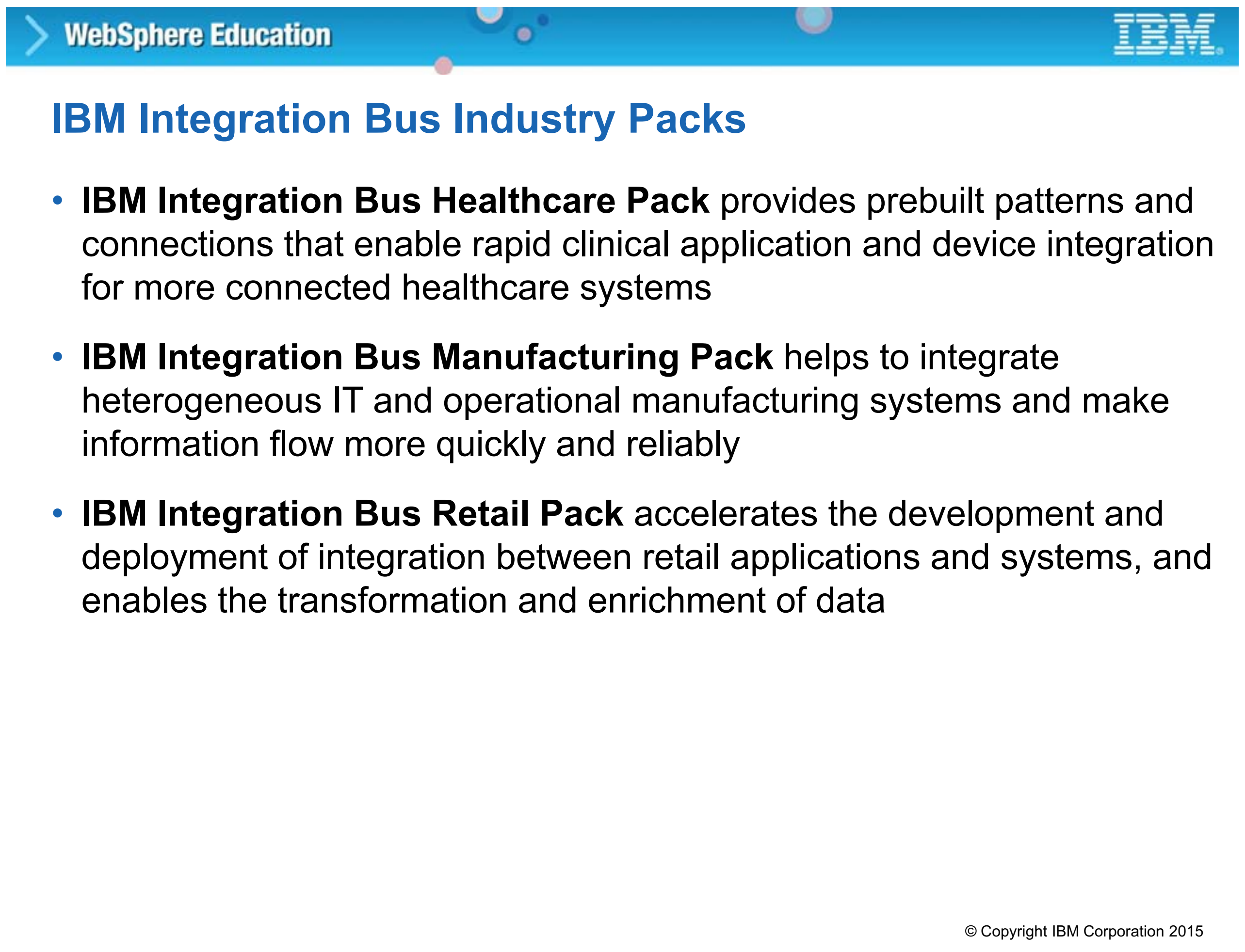
Figure 1-11. Using IBM Integration Bus to provide a REST API WM6461.0

Notes:

Representational State Transfer (REST) is a software architecture style that consists of guidelines and practices for creating scalable web services.

Swagger is a simple yet powerful representation of a RESTful API. Swagger provides a framework for describing, producing, using, and visualizing RESTful APIs.

With IBM Integration Bus V10, you can receive JSON over HTTP/HTTPS and expose a REST API. You can also use the IBM Integration Toolkit graphical data mapper capabilities for the transformation of unmodeled data structures, such as JSON messages.



WebSphere Education

IBM Integration Bus Industry Packs

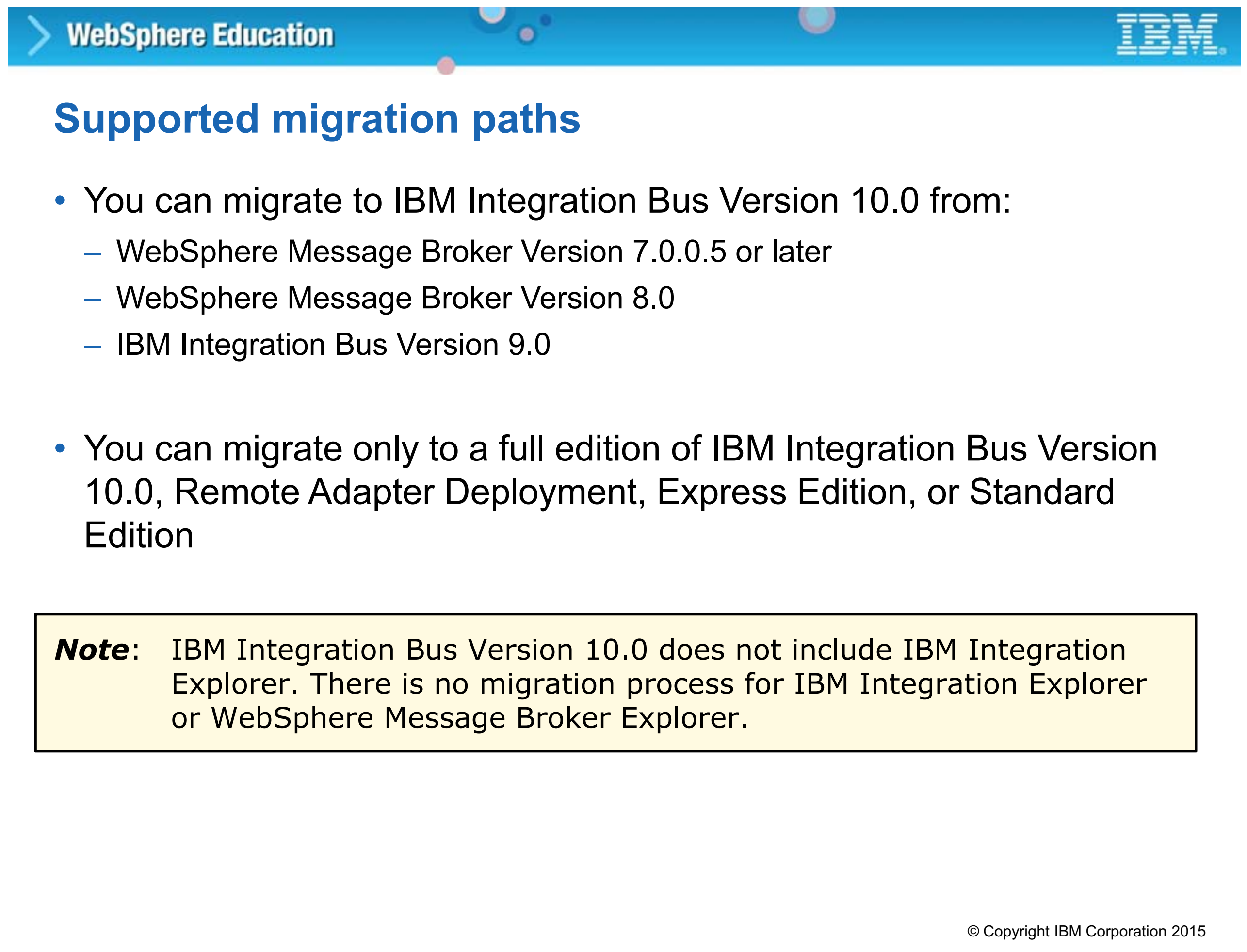
- **IBM Integration Bus Healthcare Pack** provides prebuilt patterns and connections that enable rapid clinical application and device integration for more connected healthcare systems
- **IBM Integration Bus Manufacturing Pack** helps to integrate heterogeneous IT and operational manufacturing systems and make information flow more quickly and reliably
- **IBM Integration Bus Retail Pack** accelerates the development and deployment of integration between retail applications and systems, and enables the transformation and enrichment of data

© Copyright IBM Corporation 2015

Figure 1-12. IBM Integration Bus Industry Packs WM6461.0

Notes:

The IBM Integration Bus Industry Packs provide targeted solutions and help to get you started quickly. The Industry Packs are described in more detail in course WM676, *IBM Integration Bus V10 Application Development II*.



WebSphere Education

Supported migration paths

- You can migrate to IBM Integration Bus Version 10.0 from:
  - WebSphere Message Broker Version 7.0.0.5 or later
  - WebSphere Message Broker Version 8.0
  - IBM Integration Bus Version 9.0
- You can migrate only to a full edition of IBM Integration Bus Version 10.0, Remote Adapter Deployment, Express Edition, or Standard Edition

Note: IBM Integration Bus Version 10.0 does not include IBM Integration Explorer. There is no migration process for IBM Integration Explorer or WebSphere Message Broker Explorer.

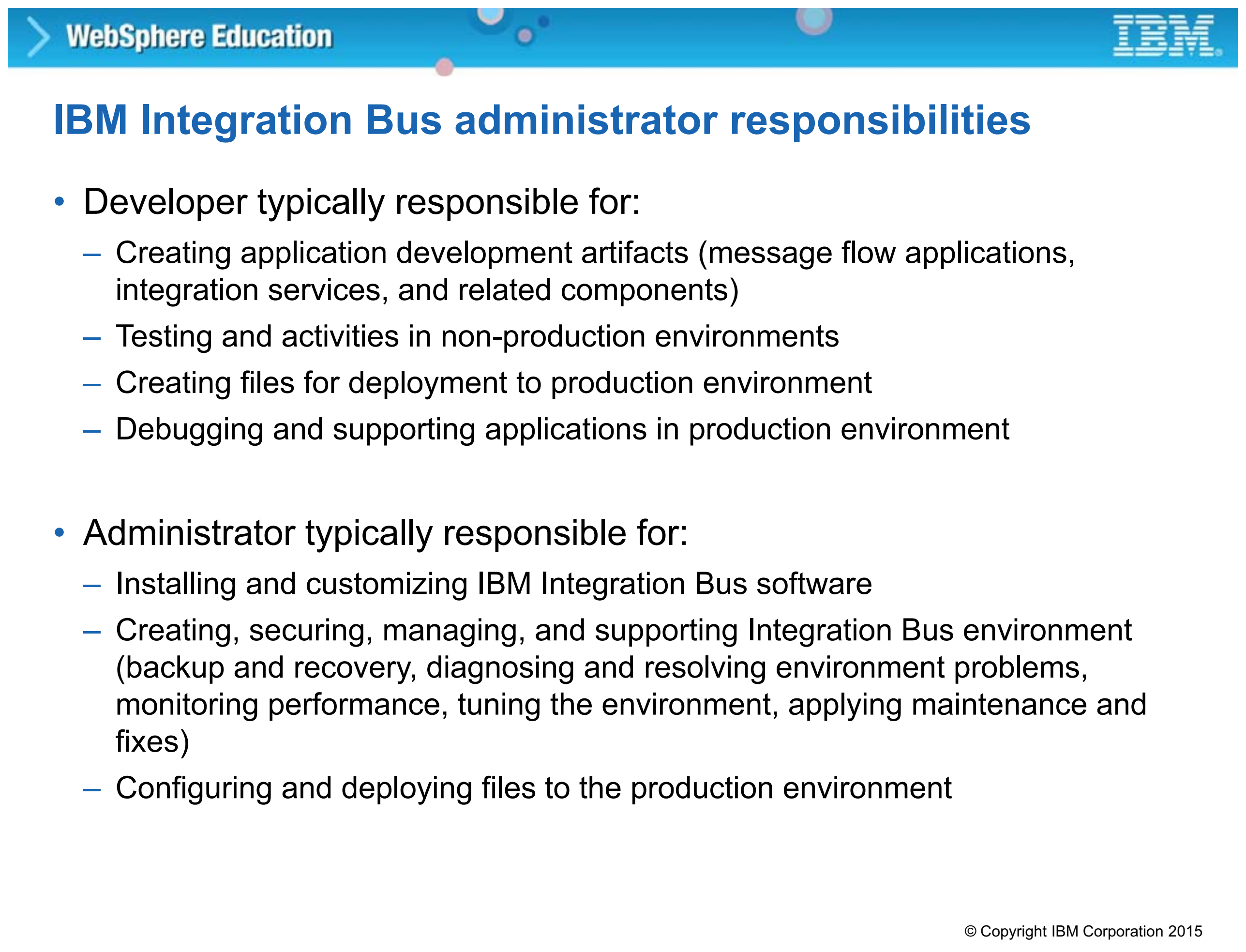
Figure 1-13. Supported migration paths WM6461.0

Notes:

IBM Integration Bus makes it simple to move applications from WebSphere Message Broker. All development assets such as message flows, ESQL and Java code, message models, and maps import directly into IBM Integration Bus.

You also have many options for moving your brokers and execution groups to integration nodes and integrations servers, including flexible co-existence options.

For more information about migration, see the IBM Knowledge Center.



WebSphere Education

IBM Integration Bus administrator responsibilities

- Developer typically responsible for:
  - Creating application development artifacts (message flow applications, integration services, and related components)
  - Testing and activities in non-production environments
  - Creating files for deployment to production environment
  - Debugging and supporting applications in production environment
- Administrator typically responsible for:
  - Installing and customizing IBM Integration Bus software
  - Creating, securing, managing, and supporting Integration Bus environment (backup and recovery, diagnosing and resolving environment problems, monitoring performance, tuning the environment, applying maintenance and fixes)
  - Configuring and deploying files to the production environment

© Copyright IBM Corporation 2015

Figure 1-14. IBM Integration Bus administrator responsibilities WM6461.0

Notes:

Many organizations that use IBM Integration Bus distinguish between the “developer” role and the “administrator” role.

Developers create message flow applications by using the facilities that the IBM Integration Bus environment provides. The development process includes testing the applications, locally on the developer workstation, and possibly in a dedicated testing environment.

Developers also prepare the applications for deployment to the production environment. When the applications are running in production, developers are also generally responsible for supporting those application components.

Administrators typically support the application development process by creating and maintaining the IBM Integration Bus environments. They install the product, and build and configure the environments in which the developer works. Administrators can be responsible for deploying components into the production environment after the developer packages those components.

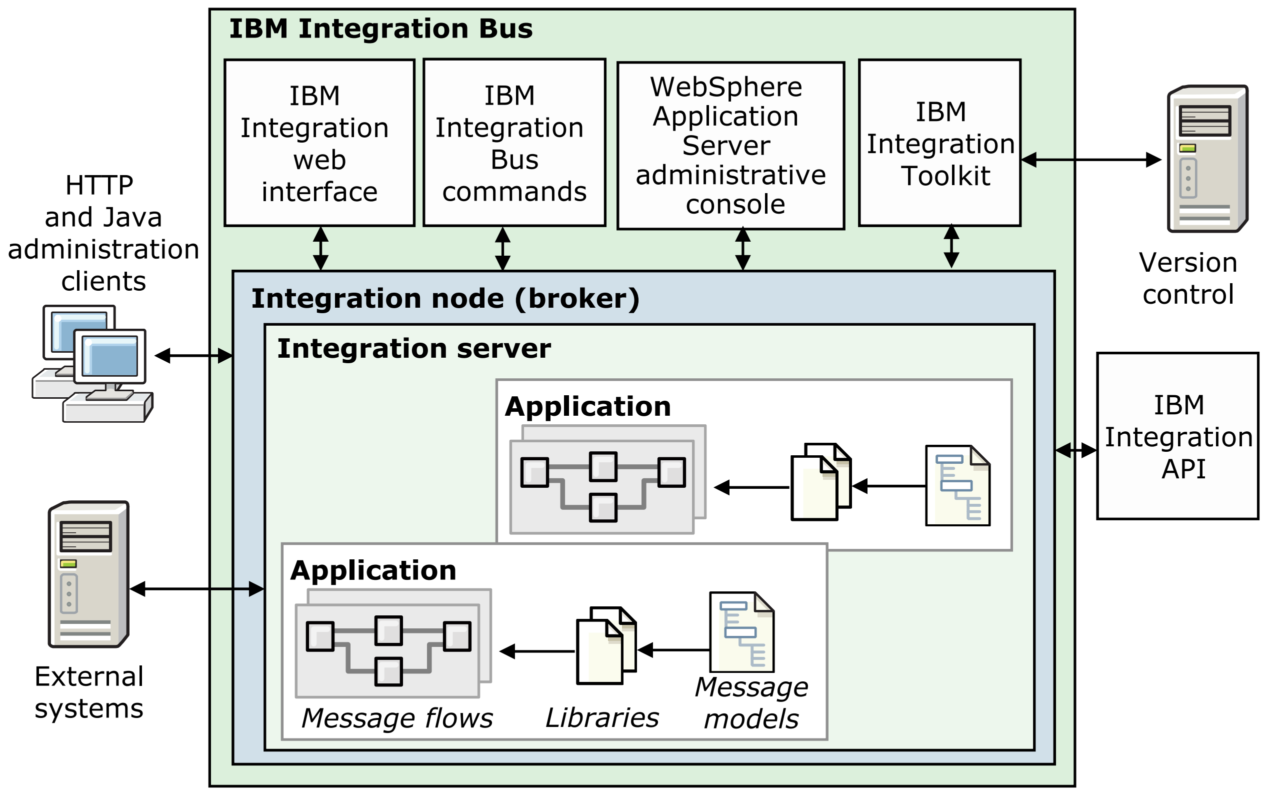
In many cases, there are overlaps between developer and administrator responsibilities. For example, creating an integration node or an integration server is usually considered to be an administrative responsibility. However, a developer might be granted the authority to create those components in a test environment, but not in a production environment. Likewise, diagnosing a problem might also require both application developers and administrators. Similarly, a change control team might be responsible for the migration of components into production, rather than the administrator. In some organizations, the same person might act both as developer and administrator.

As you review the content of this course, do not be overly concerned with “who does what.” This course focuses on “typical” administrator tasks, but the focus is on learning and understanding the tasks, rather than who has responsibility for them.

1.2. IBM Integration Bus architecture and components

WebSphere Education

IBM Integration Bus main components


IBM Integration Bus
IBM Integration web interface
IBM Integration Bus commands
WebSphere Application Server administrative console
IBM Integration Toolkit
Version control
HTTP and Java administration clients
Integration node (broker)
Integration server
Application
Application
IBM Integration API
External systems
Message flows
Libraries
Message models

© Copyright IBM Corporation 2015

Figure 1-15. IBM Integration Bus main components WM6461.0

Notes:

IBM Integration Bus includes components for development and runtime administration.

The runtime engine of IBM Integration Bus is the *integration node*, also known as the integration broker. The integration node processes in-flight messages that are based on *message flows* and messages. The message flow controls the type and sequence of operations on the incoming messages.

Message flows are run on *integration servers* to provide isolation and scalability. Message flows can interact with external systems such as web services and databases. You create message flows in the IBM Integration Toolkit, which is an integrated development and administration console. It can connect with external source control applications for team development.

A *library* is typically, a reusable, logical grouping of related code, data, or both that an application references.

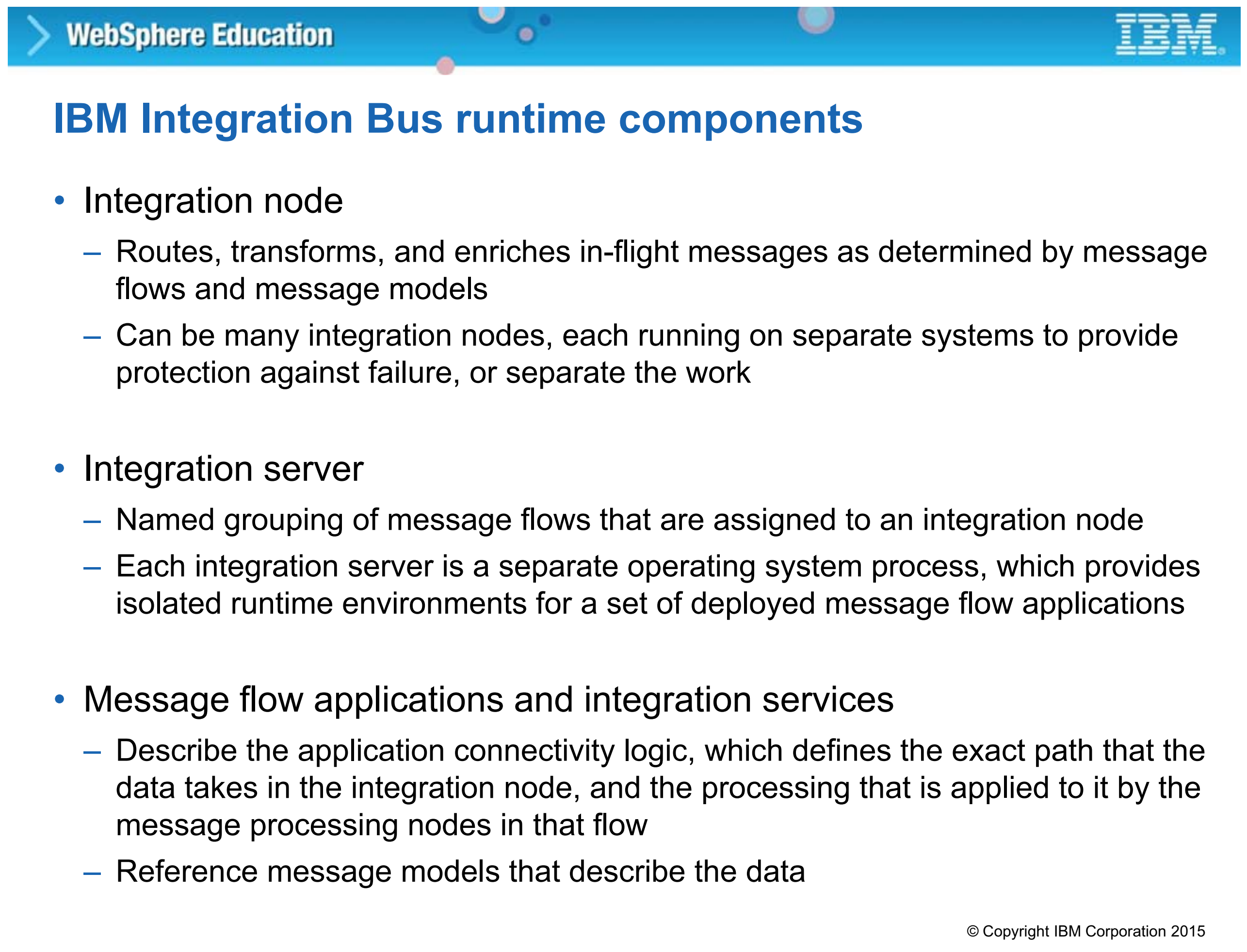
As shown in the figure, various administrative tools can be used to manage the integration node and monitor message flows and message content.

The IBM Integration web user interface is the primary administration interface. It can be used to manage integration nodes, integration servers, and message flows at run time.

The IBM Integration Toolkit is the development environment for applications, libraries, and message flows. The IBM Integration Toolkit can run only on Windows and Linux x86.

In some cases, it might be necessary or preferred to use a command interface. IBM Integration Bus supports IBM Integration administration and runtime commands.

The Integration API is a programming interface that your applications can use to control integration nodes and their resources through a remote interface. The Integration API classes and methods can also be used to develop custom applications.



WebSphere Education

IBM Integration Bus runtime components

- Integration node
  - Routes, transforms, and enriches in-flight messages as determined by message flows and message models
  - Can be many integration nodes, each running on separate systems to provide protection against failure, or separate the work
- Integration server
  - Named grouping of message flows that are assigned to an integration node
  - Each integration server is a separate operating system process, which provides isolated runtime environments for a set of deployed message flow applications
- Message flow applications and integration services
  - Describe the application connectivity logic, which defines the exact path that the data takes in the integration node, and the processing that is applied to it by the message processing nodes in that flow
  - Reference message models that describe the data

© Copyright IBM Corporation 2015

Figure 1-16. IBM Integration Bus runtime components WM6461.0

Notes:

An integration node is a system service on Windows, a daemon process on UNIX, or a started task on z/OS. It controls processes that run message flows. You can create integration nodes on every operating system that IBM Integration Bus supports.

There can be many integration nodes, and each can be running on a different system. This architecture provides protection against failure, and can separate work across different divisions in a business.

An integration node that is named TESTNODE_*UserName* is automatically created the first time that you start the IBM Integration Toolkit. This integration node is provided for development and testing.

An *integration server* is a named grouping of message flows that are assigned to an integration node. Each integration server is started as a separate operating system process, providing an isolated runtime environment for a set of deployed message flows.

Message flows describe the application connectivity logic and the path that the data takes through the integration node. The message flow requires a message model to parse or serialize the data.

WebSphere Education

IBM Integration Toolkit

- An integrated development environment and graphical user interface that is based on Eclipse
- A single perspective for compiling, testing, deploying, and fixing message flows
- Connects to one or more integration nodes to which the message flow applications are deployed
- Connects to patterns galleries for getting started quickly
- Runs on Microsoft Windows and Linux on x86

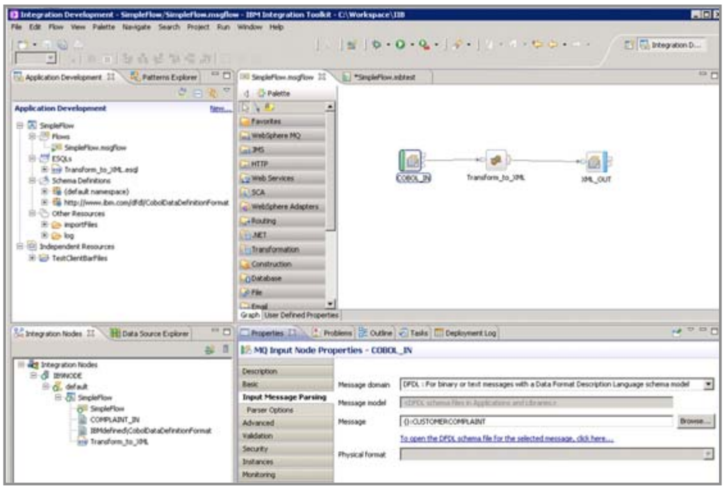


Figure 1-17. IBM Integration Toolkit WM6461.0

Notes:

The IBM Integration Toolkit is an integrated development environment and graphical user interface that is based on Eclipse. Application developers work in separate instances of the Integration Toolkit to develop applications, libraries, message models, and message flows.

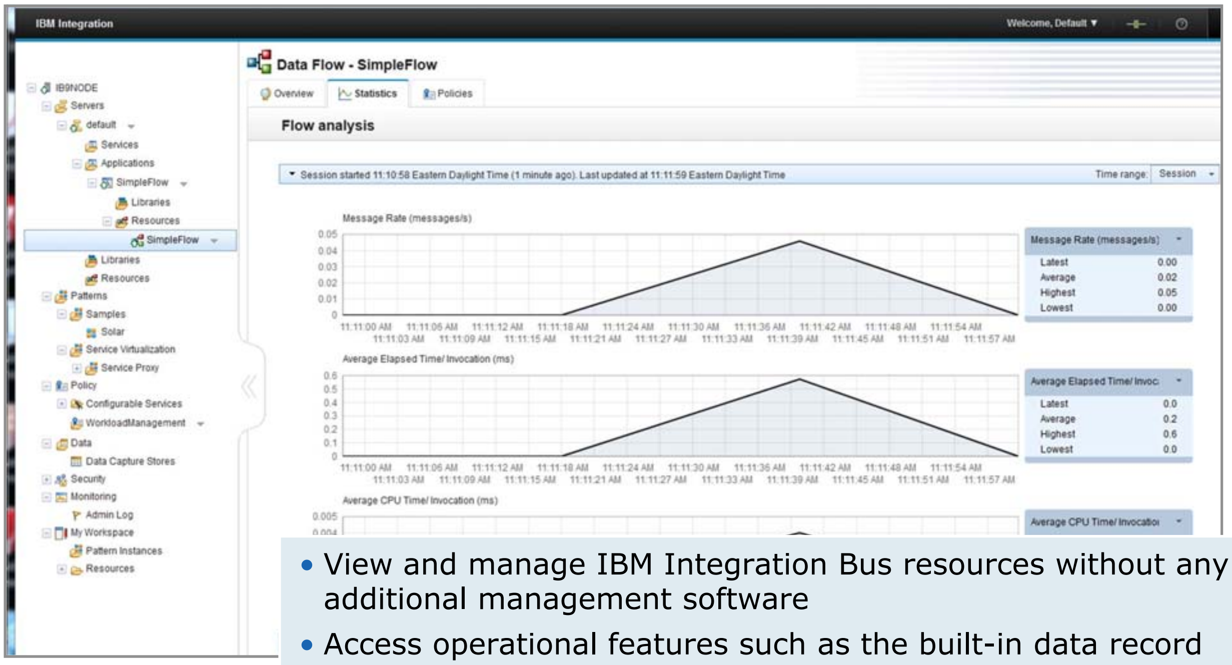
The Integration Toolkit runs on Windows and Linux and uses the local file system for its information storage. In the Integration Toolkit, you can:

- Define applications, message flows, and message flow components
- Define and import of message definitions
- Deploy message flows and message models to integration nodes
- Control log entries that are written during deployment
- Start, stop, and trace message flows that are running in integration nodes
- Connect to local and remote integration nodes

The IBM Integration Toolkit includes access to tutorials and patterns to help you get started with your application development.

WebSphere Education IBM

IBM Integration web user interface



Session started 11:10:58 Eastern Daylight Time (1 minute ago). Last updated at 11:11:59 Eastern Daylight Time

Message Rate (messages/s)	
Latest	0.00
Average	0.02
Highest	0.05
Lowest	0.00

Average Elapsed Time/ Invoc	
Latest	0.0
Average	0.2
Highest	0.6
Lowest	0.0

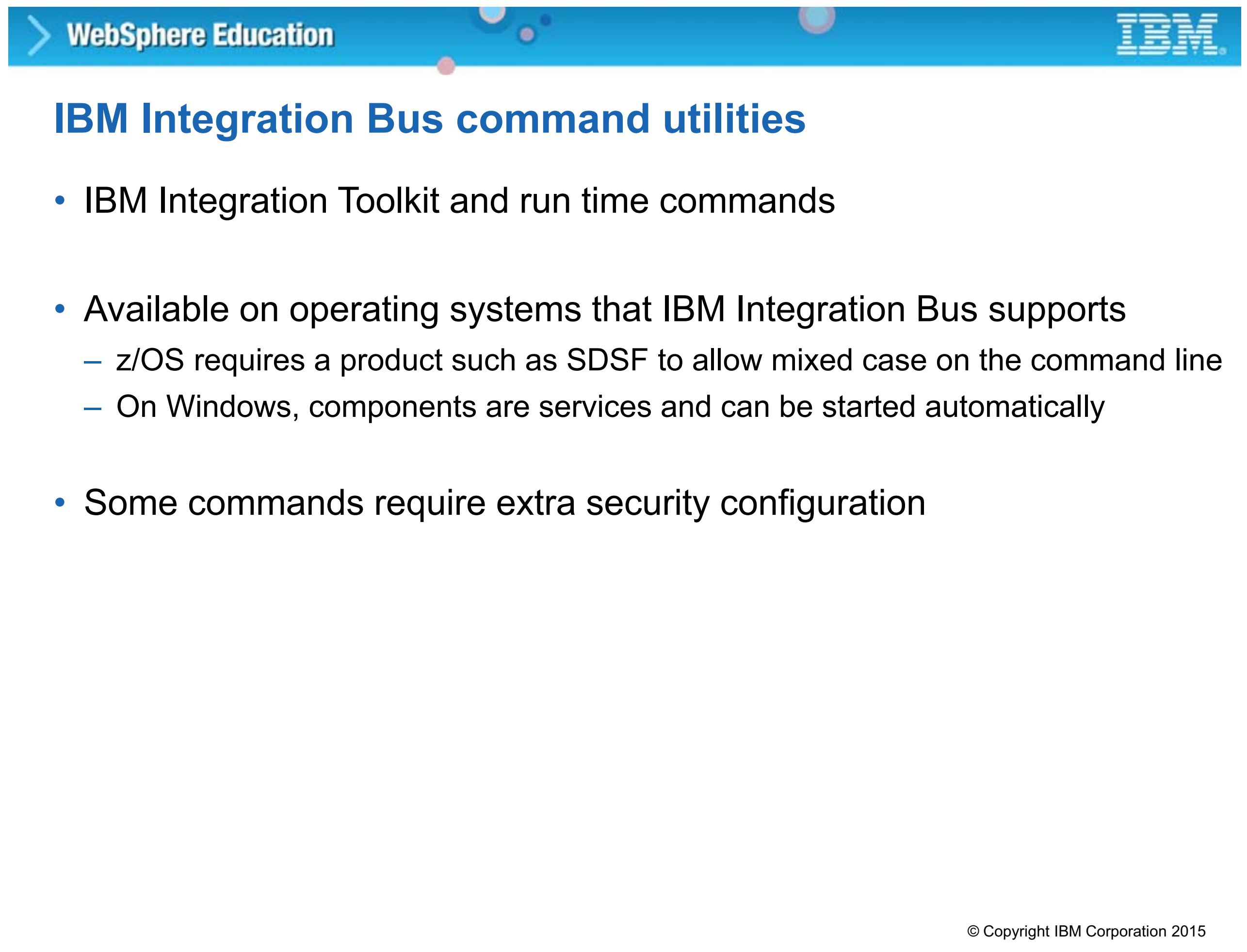
- View and manage IBM Integration Bus resources without any additional management software
- Access operational features such as the built-in data record and replay tool
- Configure policy-based workload management
- Deploy IBM Integration Bus applications

Figure 1-18. IBM Integration web user interface WM6461.0

Notes:

The IBM Integration Bus web user interface is the primary administrative interface for Integration Bus. You can view and manage Integration Bus resources without any extra management software. It provides a view of all deployed integration solutions, and gives you access to important operational features such as the built-in data record and replay tool.

IBM Integration Bus includes a set of performance monitoring tools that visually portray current server throughput rates. Tables and graphs show various metrics such as elapsed and CPU time in ways that immediately draw attention to performance bottlenecks and spikes in demand. You can get more detail, such as rates for individual connectors. You can use these tools to correlate performance information with configuration changes so that you can quickly determine the performance impact of specific configuration changes.



WebSphere Education

IBM Integration Bus command utilities

- IBM Integration Toolkit and run time commands
- Available on operating systems that IBM Integration Bus supports
  - z/OS requires a product such as SDSF to allow mixed case on the command line
  - On Windows, components are services and can be started automatically
- Some commands require extra security configuration

© Copyright IBM Corporation 2015

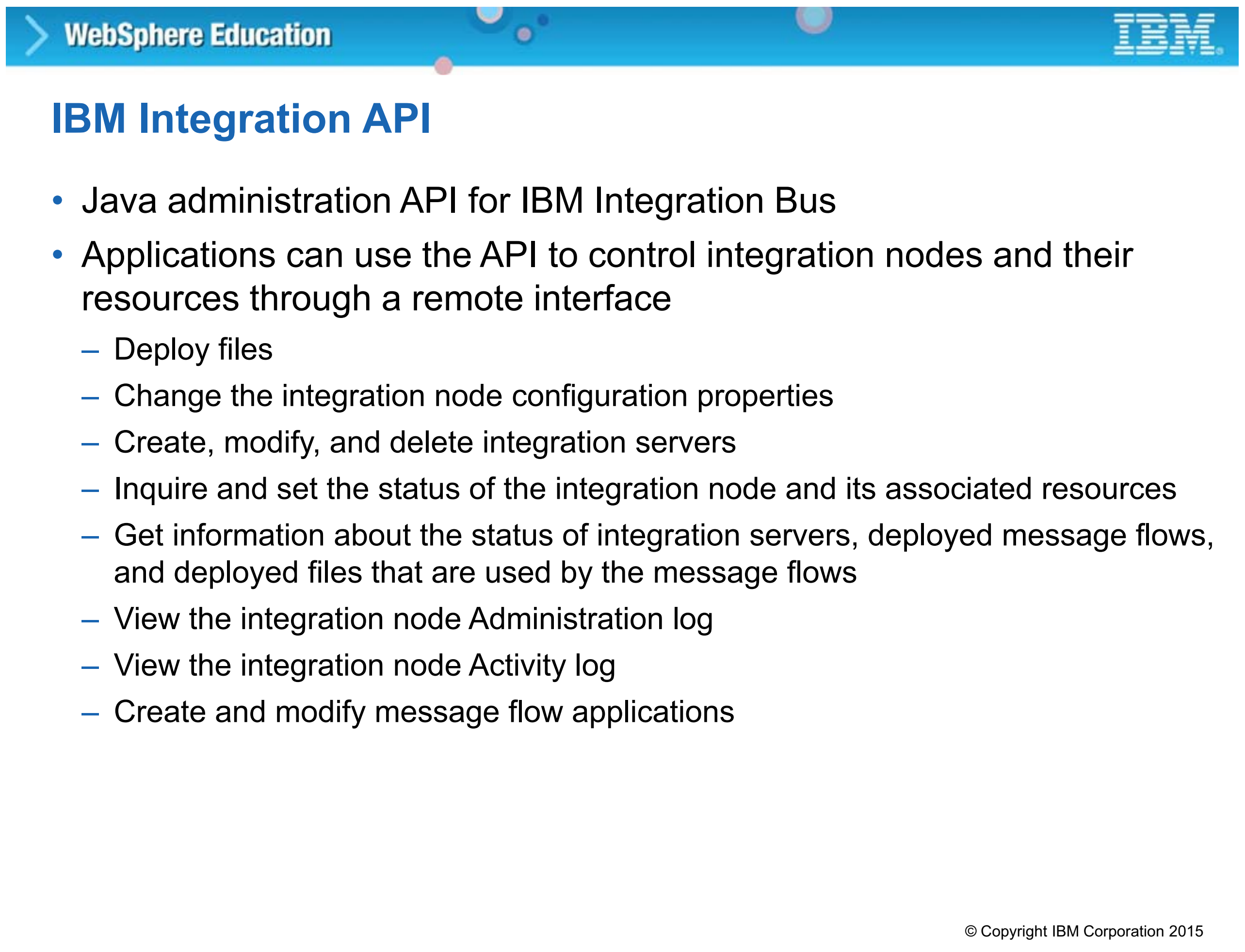
Figure 1-19. IBM Integration Bus command utilities WM6461.0

Notes:

In addition to using the IBM Integration web user interface to manage integration nodes, you can administer Integration Bus by using the command interface. You can also use the command interface to automate specific tasks. For example, you can write scripts to deploy applications to production integration servers on a schedule.

Some commands access the local components directly, others, need access to IBM Integration Toolkit resources and can run only on Windows or Linux.

Some commands require extra authorization. For example, on some operating systems, administrators must be a member of the `mqbrkrs` group to run administrative commands. For security requirements, see the IBM Knowledge Center for IBM Integration Bus.



IBM Integration API

- Java administration API for IBM Integration Bus
- Applications can use the API to control integration nodes and their resources through a remote interface
  - Deploy files
  - Change the integration node configuration properties
  - Create, modify, and delete integration servers
  - Inquire and set the status of the integration node and its associated resources
  - Get information about the status of integration servers, deployed message flows, and deployed files that are used by the message flows
  - View the integration node Administration log
  - View the integration node Activity log
  - Create and modify message flow applications

Figure 1-20. IBM Integration API WM6461.0

Notes:

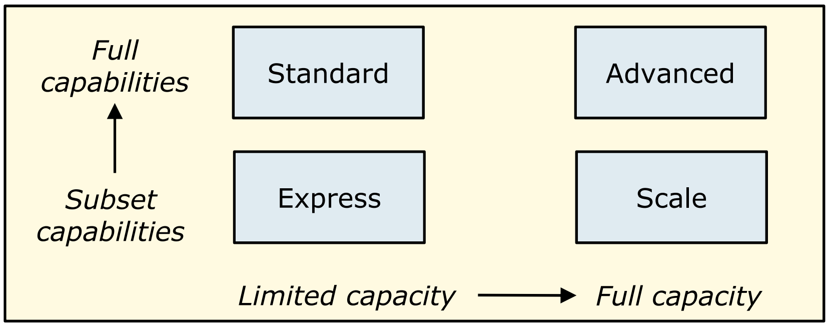
The Integration API is a set of lightweight Java classes that sit logically between the user application and the integration node. It is an alternative interface for administering integration nodes.

All IBM Integration Bus applications, such as the Integration Toolkit, use the API to communicate with the integration node environment.

IBM Integration Bus includes samples to learn the basic features that the Integration API provides.

WebSphere Education

IBM Integration Bus operation modes



Mode	Features	Integration servers	Deployed message flows
Express	Limited set of message flow nodes	One	Unlimited
Scale	Limited set of message flow nodes	Unlimited	Unlimited
Standard	All features are enabled	One	Unlimited
Advanced	All features are enabled	Unlimited	Unlimited

Figure 1-21. IBM Integration Bus operation modes WM6461.0

Notes:

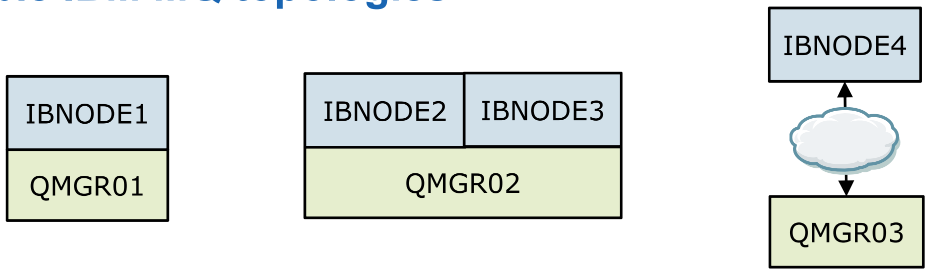
The capability and capacity that is provided by IBM Integration Bus varies according to the operation modes in which your integration nodes are running. Your entitlement to run in a particular mode depends on the edition of the product that you purchased.

Operation modes typically restrict the number of message flow nodes that are available, and integration server capacity, but many features are available across all modes of operation. This figure summarizes four of the IBM Integration Bus operation modes: Express, Scale, Standard, and Advanced. IBM Integration Bus is also available in Developer mode and Adapter mode.

- With **Developer mode**, all features are enabled, but you can use the product for evaluation, development, and test purposes only. Developer mode is available on Windows x86-64 and Linux x86-64 operating systems and is limited to one message per second at the message flow level.
- With **Adapter mode**, only adapter-related features are enabled, and the types of message flow node that you can use, and the number of integration servers that you can create, are limited.

WebSphere Education

Flexible IBM MQ topologies



- On distributed systems, IBM MQ is not a prerequisite in most implementations
  - Flexible topology options for IBM MQ access for simplicity, scalability, availability, and migration
- On z/OS, IBM MQ is required for installation

***Note**: On z/OS, only local connections to queue managers are supported.

Figure 1-22. Flexible IBM MQ topologies WM6461.0

Notes:

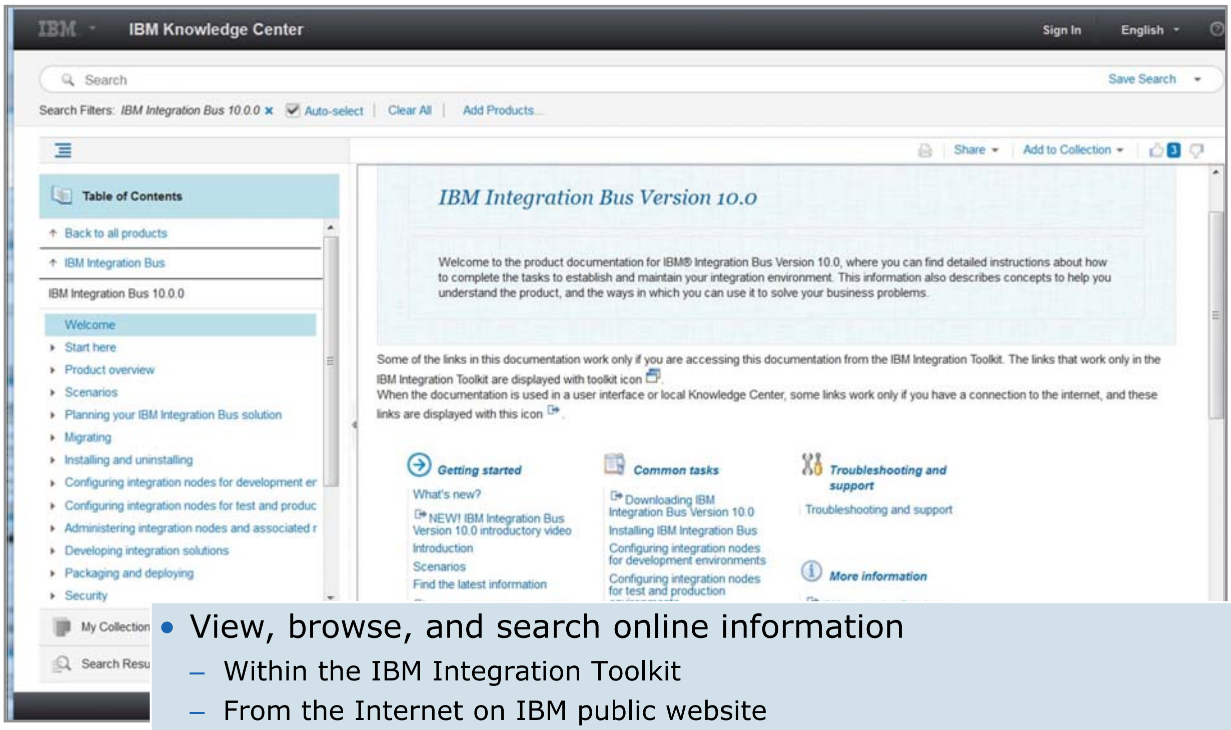
With IBM Integration Bus, you can configure local or client connections to IBM MQ, enabling your integration nodes to get messages from, or put messages to, queues on a local or remote queue manager.

You can configure either a local or client connection between your integration node and your queue manager, depending on the configuration of your existing architecture. If your IBM MQ queue manager is running on the same computer as your integration node, you can specify a local connection to the queue manager. Alternatively, if the IBM MQ queue manager that you want to connect to is hosted on a separate computer from IBM Integration Bus, you can configure a client connection from your integration node. The integration node with the client connection can access the messages on the remote queue manager.

You can define IBM MQ endpoint policies to create reusable run time resource and connection definitions.

WebSphere Education

IBM Knowledge Center for IBM Integration Bus



IBM Knowledge Center — Sign In — English

Search — Save Search

Search Filters: IBM Integration Bus 10.0.0 × — Auto-select — Clear All — Add Products...

Share — Add to Collection

Table of Contents
- Back to all products
- IBM Integration Bus
- IBM Integration Bus 10.0.0
- Welcome
- Start here
- Product overview
- Scenarios
- Planning your IBM Integration Bus solution
- Migrating
- Installing and uninstalling
- Configuring integration nodes for development er
- Configuring integration nodes for test and produc
- Administering integration nodes and associated r
- Developing integration solutions
- Packaging and deploying
- Security

My Collection

Search Resu

IBM Integration Bus Version 10.0

Welcome to the product documentation for IBM® Integration Bus Version 10.0, where you can find detailed instructions about how to complete the tasks to establish and maintain your integration environment. This information also describes concepts to help you understand the product, and the ways in which you can use it to solve your business problems.

Some of the links in this documentation work only if you are accessing this documentation from the IBM Integration Toolkit. The links that work only in the IBM Integration Toolkit are displayed with toolkit icon.
When the documentation is used in a user interface or local Knowledge Center, some links work only if you have a connection to the internet, and these links are displayed with this icon.

Getting started
- What's new?
- NEW! IBM Integration Bus Version 10.0 introductory video
- Introduction
- Scenarios
- Find the latest information

Common tasks
- Downloading IBM Integration Bus Version 10.0
- Installing IBM Integration Bus
- Configuring integration nodes for development environments
- Configuring integration nodes for test and production

Troubleshooting and support
- Troubleshooting and support

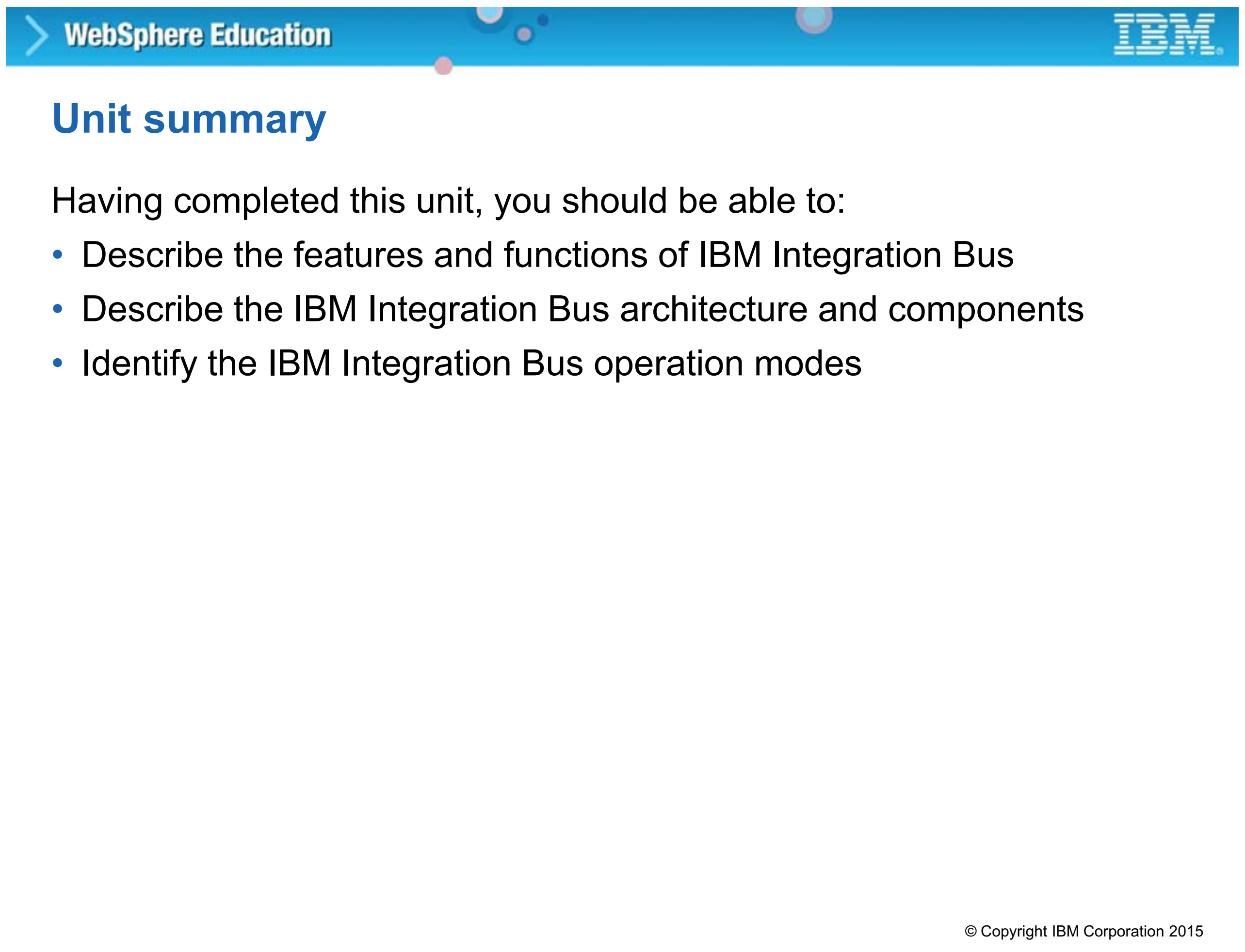
More information

- View, browse, and search online information
  - Within the IBM Integration Toolkit
  - From the Internet on IBM public website
    `www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/`
    `com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm/`

Figure 1-23. IBM Knowledge Center for IBM Integration Bus WM6461.0

Notes:

It is important to know where to go to get more information about IBM Integration Bus and its application. The IBM Knowledge Center for IBM Integration Bus Version 10 contains a complete set of application documentation. It is available online and with the IBM Integration Toolkit.



WebSphere Education

Unit summary

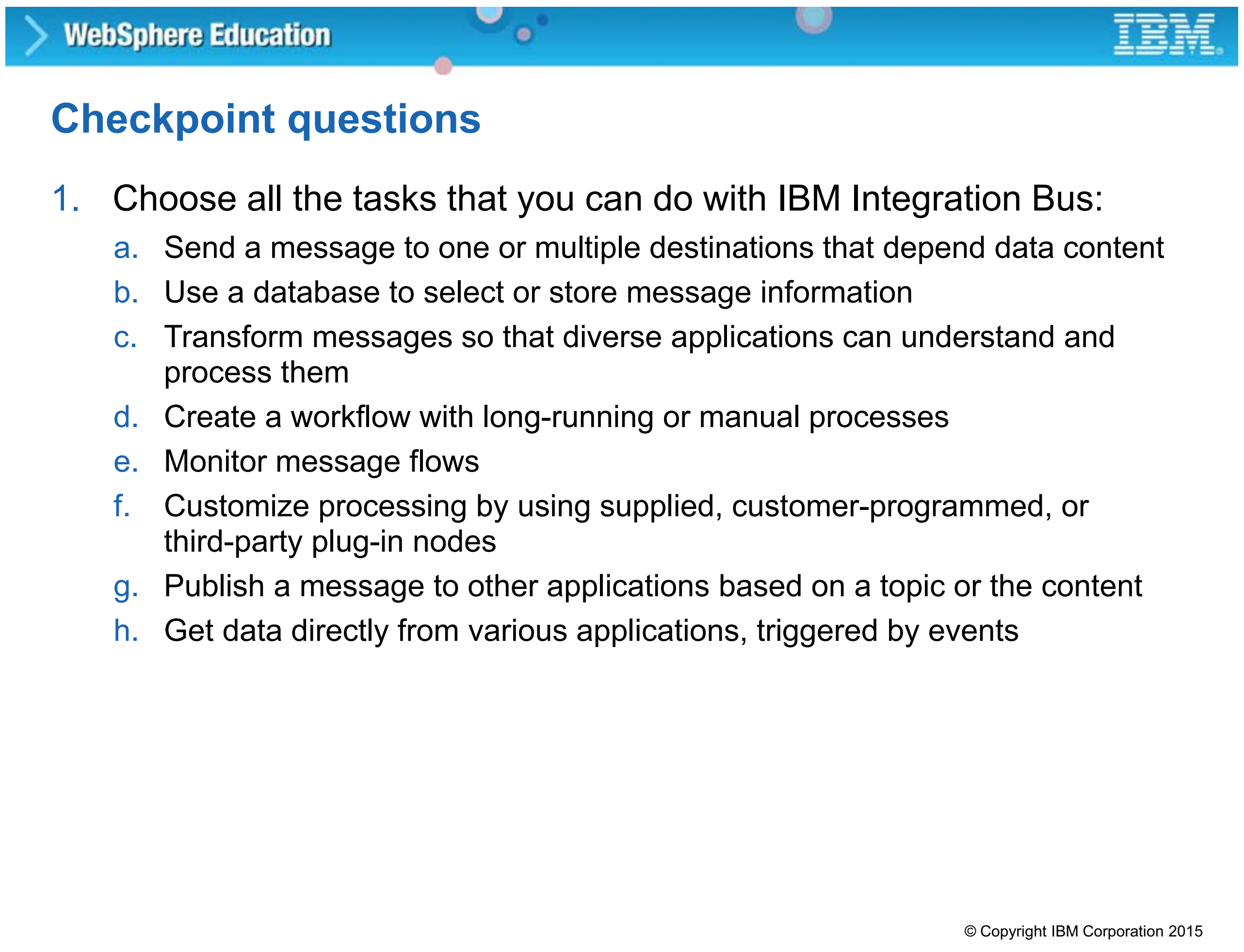
Having completed this unit, you should be able to:

- Describe the features and functions of IBM Integration Bus
- Describe the IBM Integration Bus architecture and components
- Identify the IBM Integration Bus operation modes

© Copyright IBM Corporation 2015

Figure 1-24. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

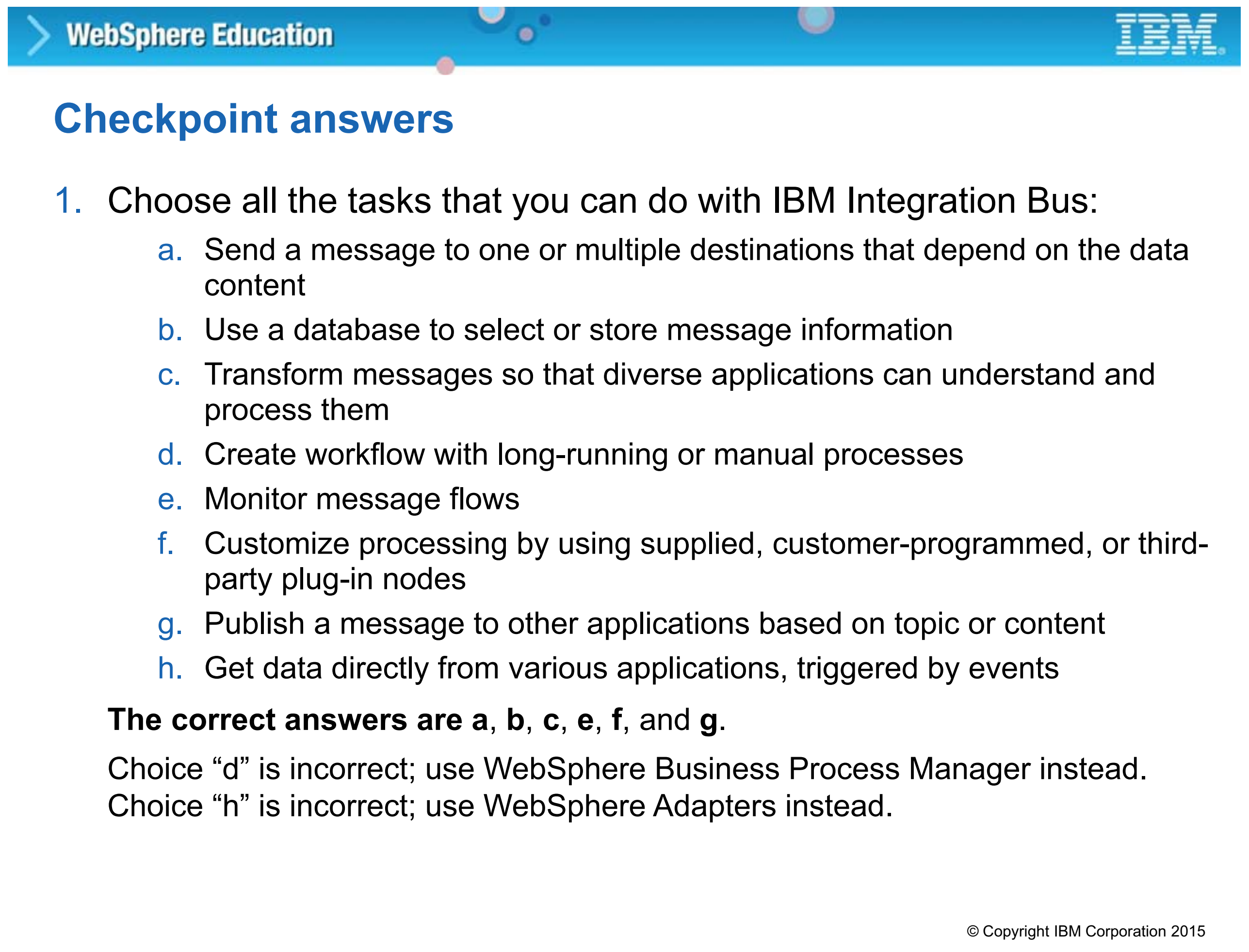
1. Choose all the tasks that you can do with IBM Integration Bus:
   a. Send a message to one or multiple destinations that depend data content
   b. Use a database to select or store message information
   c. Transform messages so that diverse applications can understand and process them
   d. Create a workflow with long-running or manual processes
   e. Monitor message flows
   f. Customize processing by using supplied, customer-programmed, or third-party plug-in nodes
   g. Publish a message to other applications based on a topic or the content
   h. Get data directly from various applications, triggered by events

Figure 1-25. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.



WebSphere Education

Checkpoint answers

1. Choose all the tasks that you can do with IBM Integration Bus:
   a. Send a message to one or multiple destinations that depend on the data content
   b. Use a database to select or store message information
   c. Transform messages so that diverse applications can understand and process them
   d. Create workflow with long-running or manual processes
   e. Monitor message flows
   f. Customize processing by using supplied, customer-programmed, or third-party plug-in nodes
   g. Publish a message to other applications based on topic or content
   h. Get data directly from various applications, triggered by events

The correct answers are a, **b**, **c**, **e**, **f**, and **g.**

Choice “d” is incorrect; use WebSphere Business Process Manager instead.
Choice “h” is incorrect; use WebSphere Adapters instead.

Figure 1-26. Checkpoint answers WM6461.0

Notes:

Unit 2. Product installation, configuration, and security planning

What this unit is about

This unit describes the installation preparation and procedures for IBM Integration Bus. It also describes the runtime security considerations.

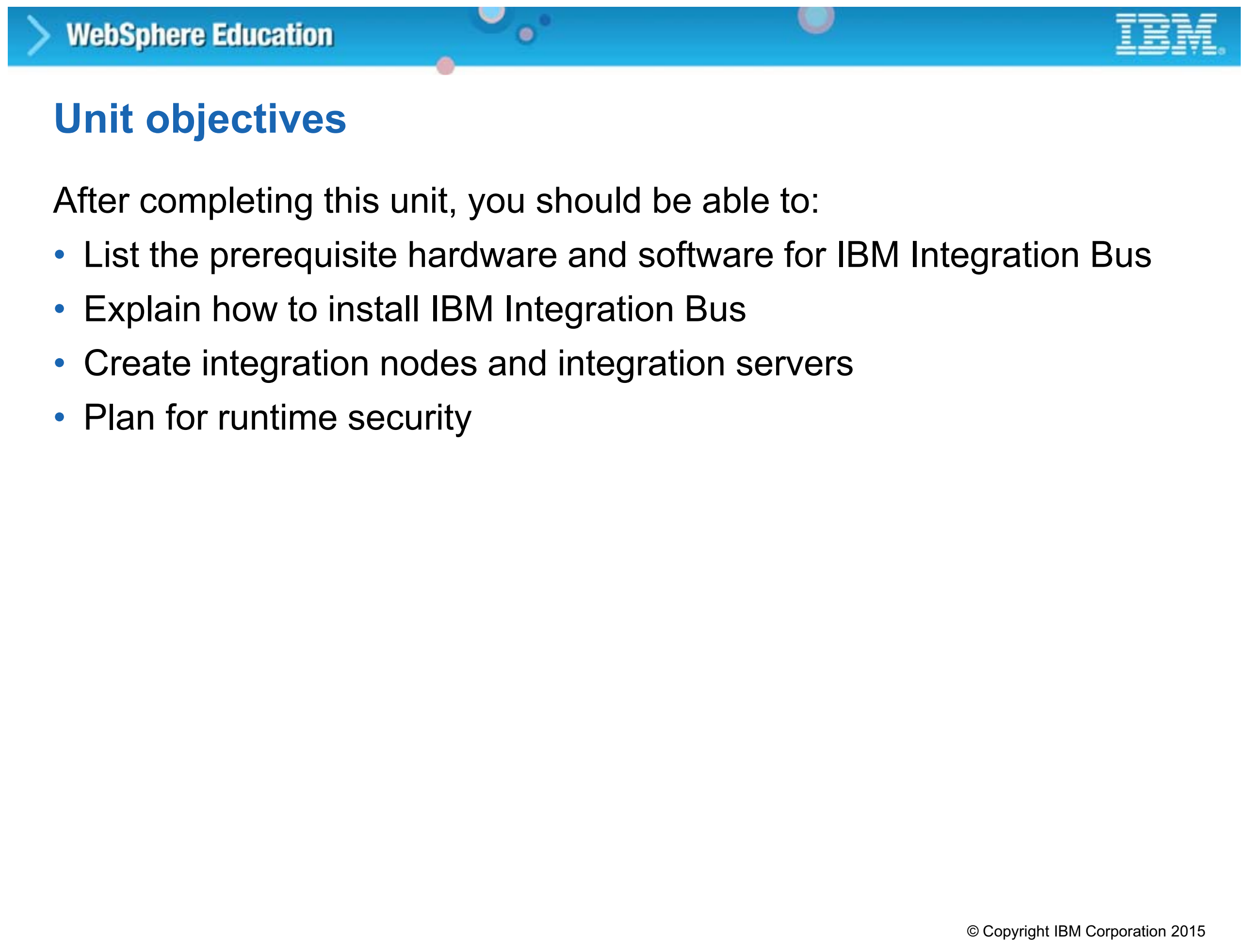
What you should be able to do

After completing this unit, you should be able to:

- List the prerequisite hardware and software for IBM Integration Bus
- Explain how to install IBM Integration Bus
- Create integration nodes and integration servers
- Plan for runtime security

How you will check your progress

- Checkpoint questions
- Exercise: Integration node setup and customization



WebSphere Education

Unit objectives

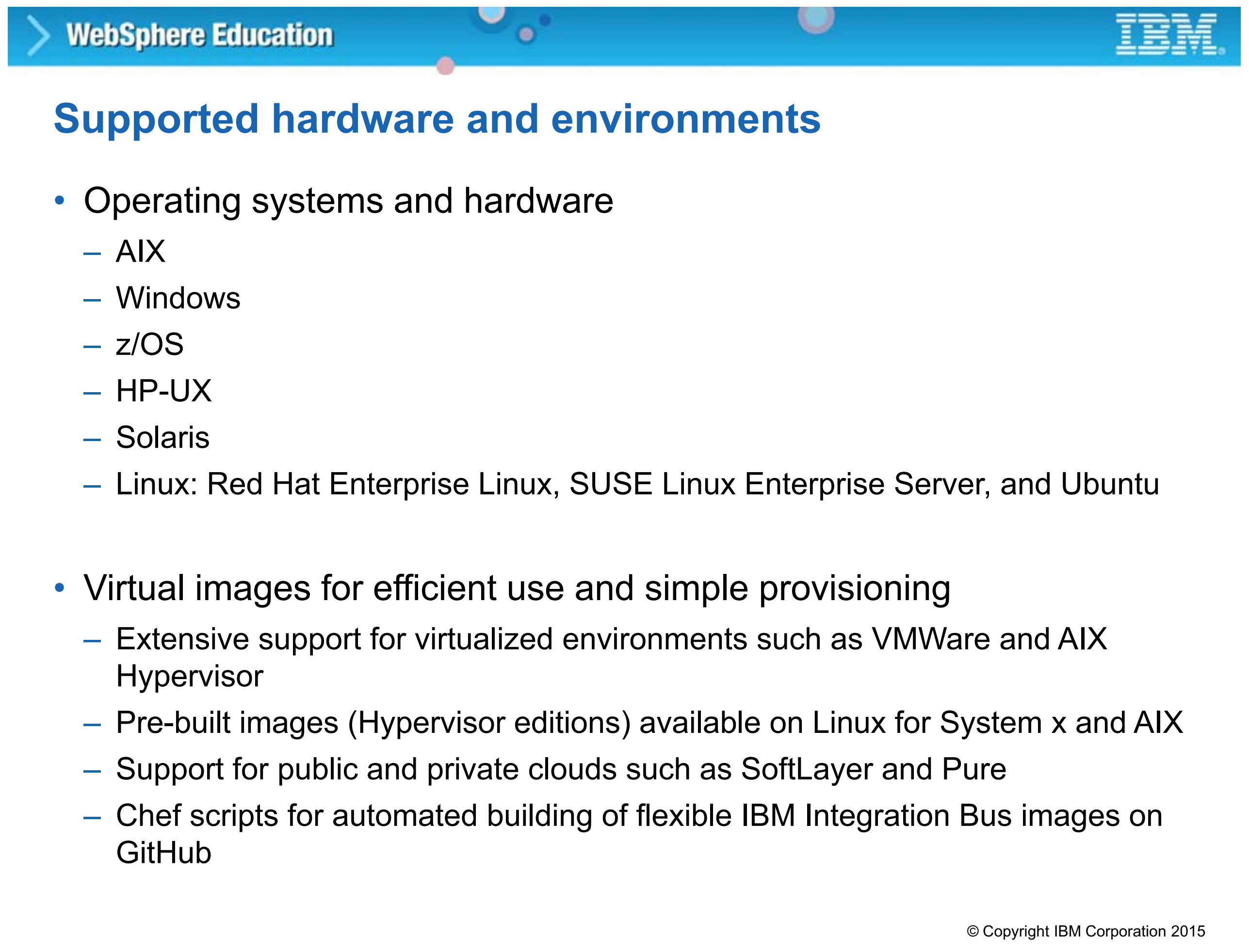
After completing this unit, you should be able to:

- List the prerequisite hardware and software for IBM Integration Bus
- Explain how to install IBM Integration Bus
- Create integration nodes and integration servers
- Plan for runtime security

Figure 2-1. Unit objectives WM6461.0

Notes:

2.1. Hardware and software requirements



WebSphere Education

Supported hardware and environments

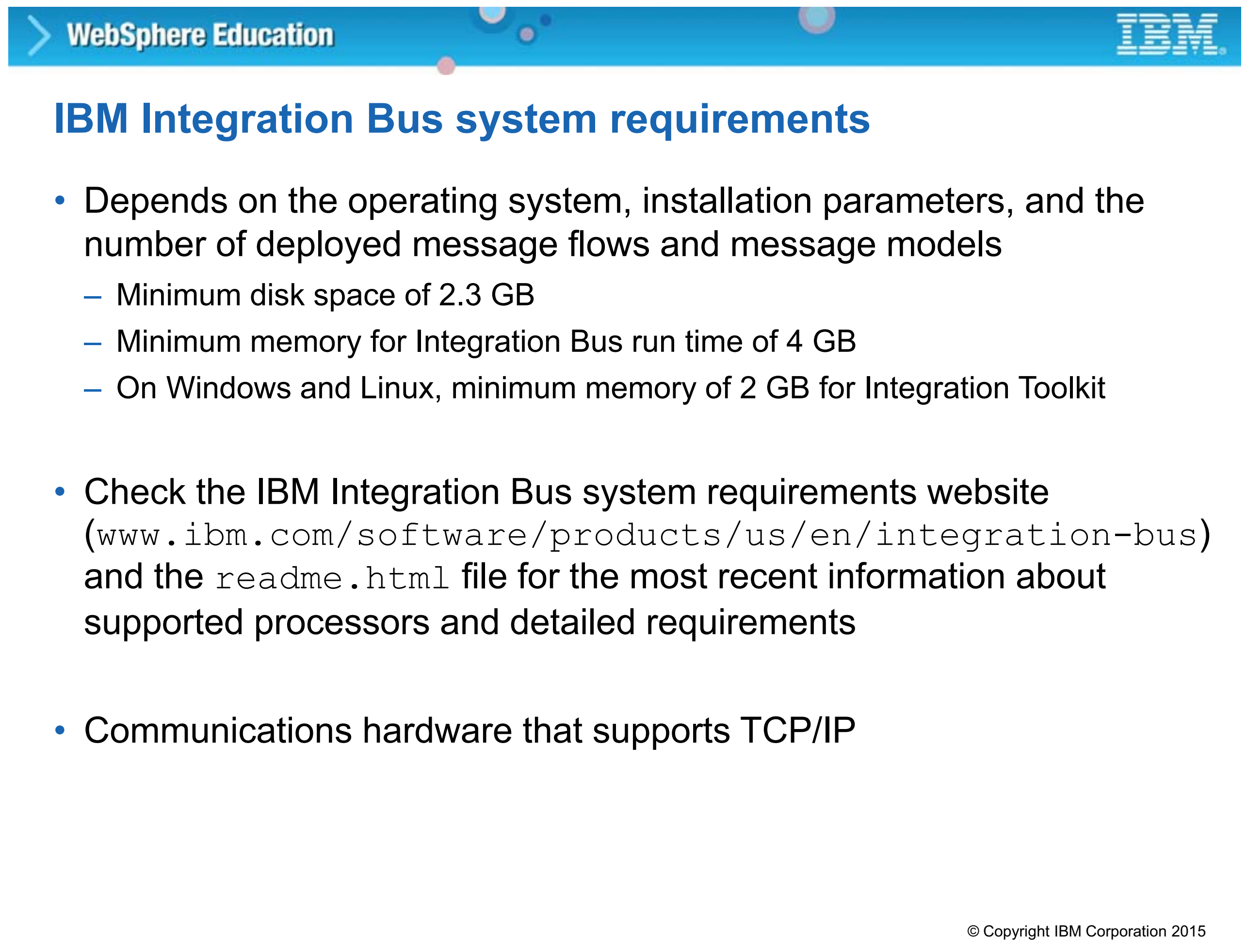
- Operating systems and hardware
  - AIX
  - Windows
  - z/OS
  - HP-UX
  - Solaris
  - Linux: Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu
- Virtual images for efficient use and simple provisioning
  - Extensive support for virtualized environments such as VMWare and AIX Hypervisor
  - Pre-built images (Hypervisor editions) available on Linux for System x and AIX
  - Support for public and private clouds such as SoftLayer and Pure
  - Chef scripts for automated building of flexible IBM Integration Bus images on GitHub

© Copyright IBM Corporation 2015

Figure 2-2. Supported hardware and environments WM6461.0

Notes:

This figure lists the supported and hardware and software environments. For a complete list of supported operating systems, hardware, virtual images, databases, and ERP systems, refer to the IBM Integration key prerequisites on the IBM website.



WebSphere Education

IBM Integration Bus system requirements

- Depends on the operating system, installation parameters, and the number of deployed message flows and message models
  - Minimum disk space of 2.3 GB
  - Minimum memory for Integration Bus run time of 4 GB
  - On Windows and Linux, minimum memory of 2 GB for Integration Toolkit
- Check the IBM Integration Bus system requirements website (`www.ibm.com/software/products/us/en/integration-bus`) and the `readme.html` file for the most recent information about supported processors and detailed requirements
- Communications hardware that supports TCP/IP

© Copyright IBM Corporation 2015

Figure 2-3. IBM Integration Bus system requirements WM6461.0

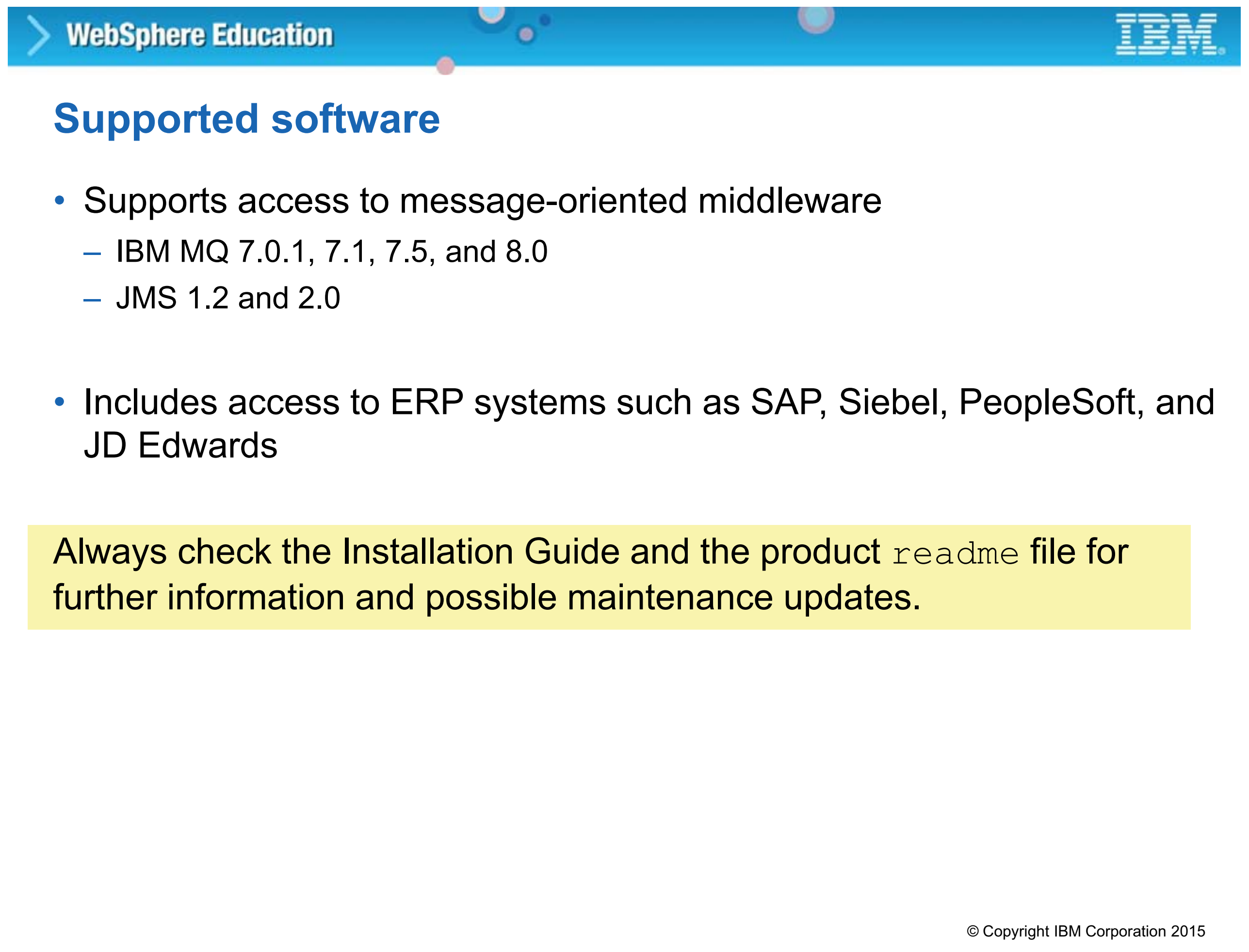
Notes:

The hardware requirements that are listed in this figure represent the *minimum* requirements. Your installation might require more capacity, depending on other applications that are running on your system and the amount of memory that is installed. In general, more memory provides better performance.

The installation of IBM Integration Bus for z/OS uses approximately 400 MB of disk space; plan on using 500 MB to allow for the component directories, and for new service fixes to be applied.

Disk speed is a factor for I/O-intensive message flows.

Recommendations, and monitoring and measuring techniques are available in a series of operating system-specific, performance-oriented SupportPacs that are accessible at:
`http://www.ibm.com/software/integration/wbimessagebroker/support`



WebSphere Education

Supported software

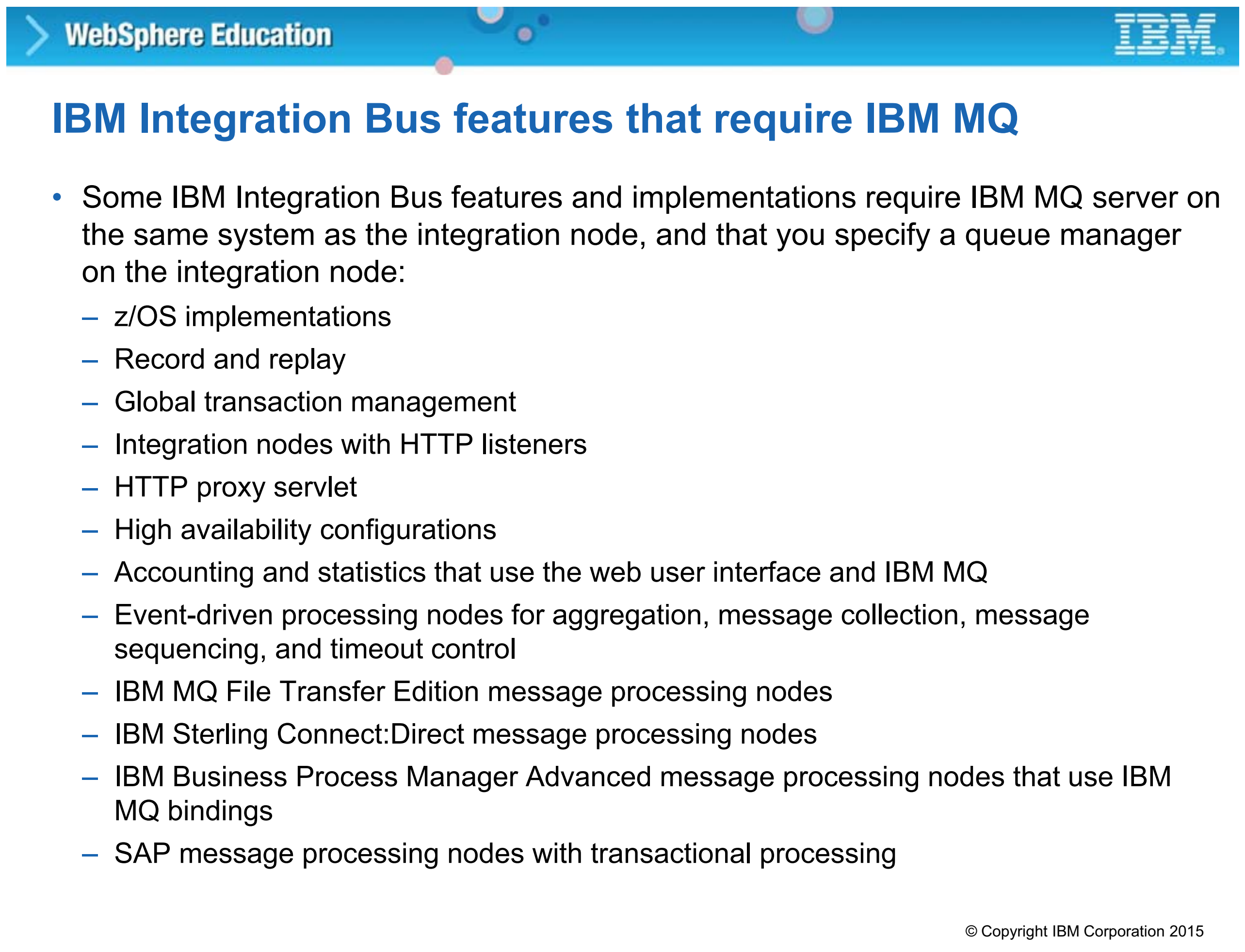
- Supports access to message-oriented middleware
  - IBM MQ 7.0.1, 7.1, 7.5, and 8.0
  - JMS 1.2 and 2.0
- Includes access to ERP systems such as SAP, Siebel, PeopleSoft, and JD Edwards

Always check the Installation Guide and the product `readme` file for further information and possible maintenance updates.

Figure 2-4. Supported software WM6461.0

Notes:

IBM Integration Bus supports message-oriented middleware such as IBM MQ and JMS. It also supports access to ERP systems though product-specific message processing nodes.



WebSphere Education

IBM Integration Bus features that require IBM MQ

- Some IBM Integration Bus features and implementations require IBM MQ server on the same system as the integration node, and that you specify a queue manager on the integration node:
  - z/OS implementations
  - Record and replay
  - Global transaction management
  - Integration nodes with HTTP listeners
  - HTTP proxy servlet
  - High availability configurations
  - Accounting and statistics that use the web user interface and IBM MQ
  - Event-driven processing nodes for aggregation, message collection, message sequencing, and timeout control
  - IBM MQ File Transfer Edition message processing nodes
  - IBM Sterling Connect:Direct message processing nodes
  - IBM Business Process Manager Advanced message processing nodes that use IBM MQ bindings
  - SAP message processing nodes with transactional processing

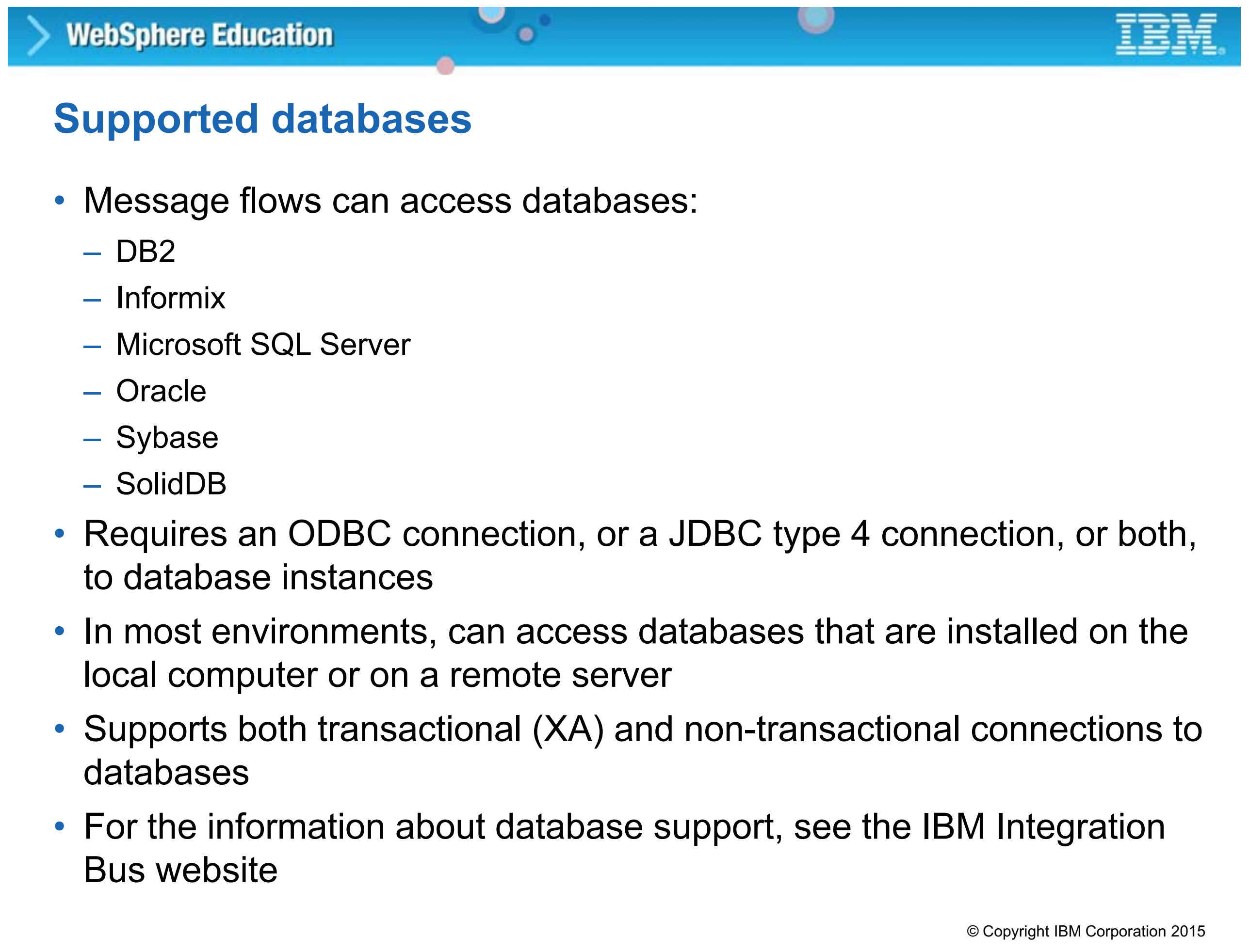
Figure 2-5. IBM Integration Bus features that require IBM MQ WM6461.0

Notes:

In previous versions of IBM Integration Bus and WebSphere Message Broker, the Integration Bus run time (the *integration node* or *broker*) required access to a dedicated IBM MQ queue manager for internal operations. In IBM Integration Bus V10, the requirement for an IBM MQ queue manager is not required for many implementations. This figure lists the implementations and message processing nodes that require an IBM MQ queue manager.

For example, the integration node HTTP listener requires access to an IBM MQ queue manager. You must install an IBM MQ server if you want to use an integration node listener to manage HTTP messages in your HTTP or SOAP flows. However, if you use HTTP nodes or SOAP nodes with the integration server embedded listener, the integration node does not require access to IBM MQ.

Many of these implementations and features are described in more detail later in this course.



WebSphere Education

Supported databases

- Message flows can access databases:
  - DB2
  - Informix
  - Microsoft SQL Server
  - Oracle
  - Sybase
  - SolidDB
- Requires an ODBC connection, or a JDBC type 4 connection, or both, to database instances
- In most environments, can access databases that are installed on the local computer or on a remote server
- Supports both transactional (XA) and non-transactional connections to databases
- For the information about database support, see the IBM Integration Bus website

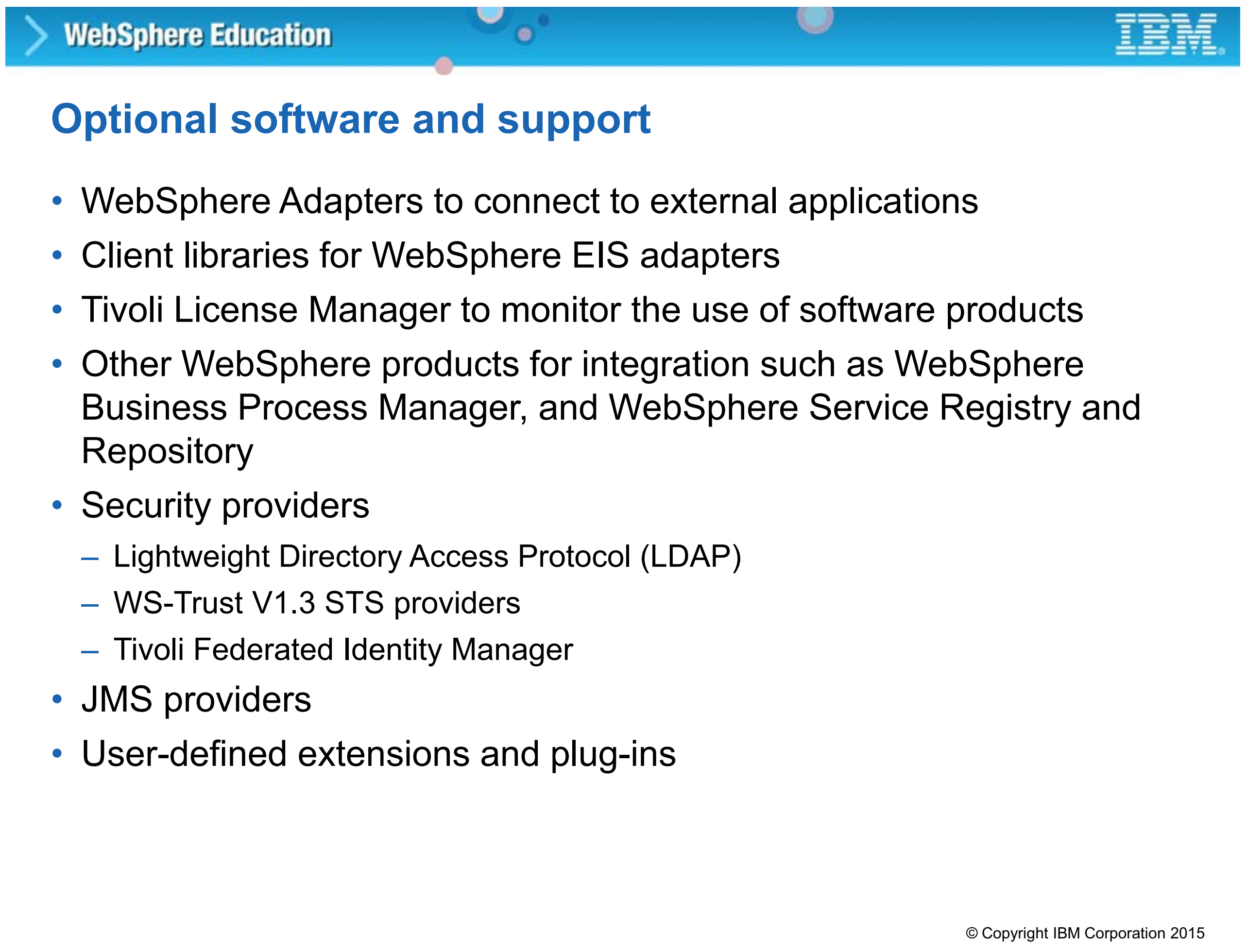
Figure 2-6. Supported databases WM6461.0

Notes:

IBM Integration Bus message flows can be configured to access databases at run time, provided there is an ODBC or JDBC type 4 connection.

Supported releases of DB2 UDB, Oracle, Sybase, and Informix can participate as a resource manager in a distributed XA transaction. IBM MQ can act as the XA transaction manager. This capability can be important depending on how the application developers use IBM Integration Bus.

The information that is shown in this figure indicates the support for databases valid when this course was published. For the latest information about database support, see the IBM Integration Bus *System Requirements* website.



WebSphere Education

Optional software and support

- WebSphere Adapters to connect to external applications
- Client libraries for WebSphere EIS adapters
- Tivoli License Manager to monitor the use of software products
- Other WebSphere products for integration such as WebSphere Business Process Manager, and WebSphere Service Registry and Repository
- Security providers
  - Lightweight Directory Access Protocol (LDAP)
  - WS-Trust V1.3 STS providers
  - Tivoli Federated Identity Manager
- JMS providers
- User-defined extensions and plug-ins

© Copyright IBM Corporation 2015

Figure 2-7. Optional software and support WM6461.0

Notes:

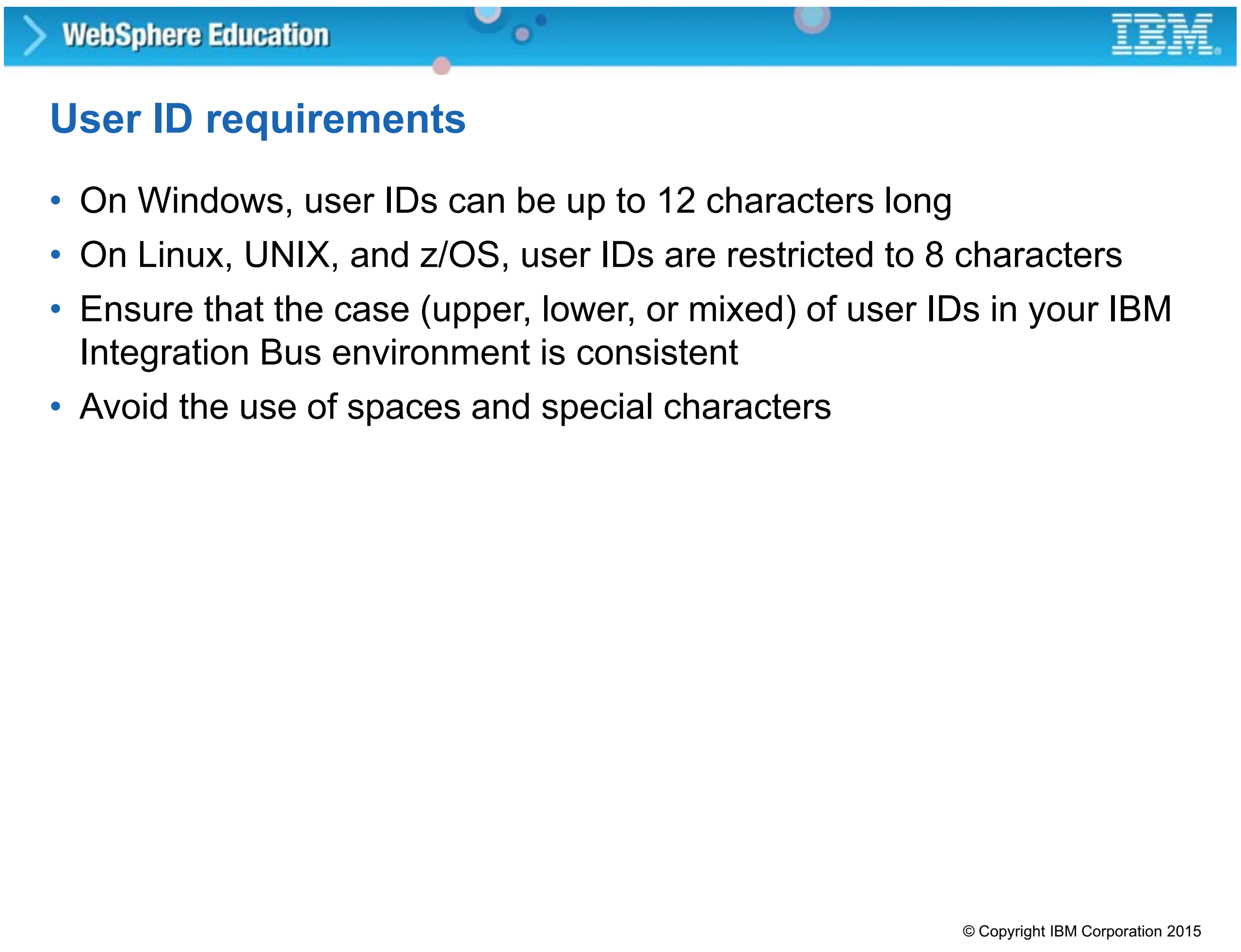
This figure lists other products that IBM Integration Bus supports. The products that are listed are not required, but might be useful. Except where stated, these products are not supplied with IBM Integration Bus.

WebSphere Adapters are required to connect to external applications. You must obtain the appropriate EIS client libraries from the EIS vendor. For details of the versions of WebSphere Adapters supported by IBM Integration Bus, see the IBM Integration Bus *System Requirements* website.

Integration Bus supports WebSphere Service Registry and Repository message processing nodes. These nodes allow a message flow to retrieve data dynamically from the repository at run time, and to use and access those resources in the message flow.

External security providers and repositories are required for some security implementations.

JMS providers are required if any message flows use the JMSInput and JMSOutput message processing nodes.



User ID requirements

- On Windows, user IDs can be up to 12 characters long
- On Linux, UNIX, and z/OS, user IDs are restricted to 8 characters
- Ensure that the case (upper, lower, or mixed) of user IDs in your IBM Integration Bus environment is consistent
- Avoid the use of spaces and special characters

© Copyright IBM Corporation 2015

Figure 2-8. User ID requirements WM6461.0

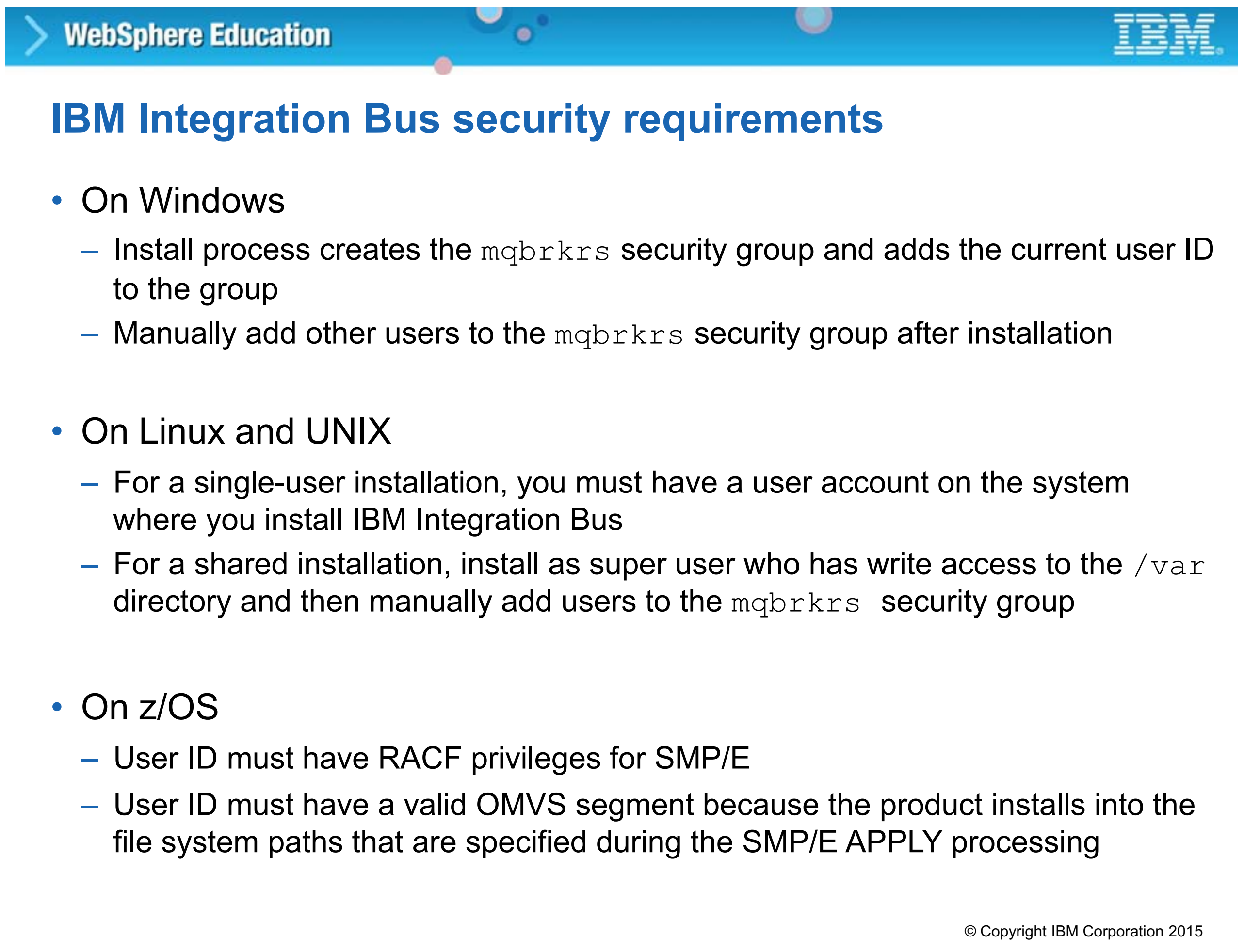
Notes:

Security control of Integration Bus components, resources, and tasks depend on the definition of users and groups of users (principals) to the security subsystem of the operating system. Check that you have the correct authority, and that the required principals are in place, before you install IBM Integration Bus.

There are platform-specific restrictions on IDs for users and services:

- Windows IDs: 12 characters maximum
- Linux, UNIX, z/OS IDs: 8 characters maximum
- Mixed-environment IDs: 8 characters maximum
- Ensure that case (uppercase or lowercase) is consistent. Some operating systems consider “User1” and “USER1” to be the same ID, others do not.

The Installation Guide describes the security tasks that you must complete before, during, and after installation for the supported operating systems.



WebSphere Education
IBM

IBM Integration Bus security requirements

- On Windows
  - Install process creates the `mqbrkrs` security group and adds the current user ID to the group
  - Manually add other users to the `mqbrkrs` security group after installation
- On Linux and UNIX
  - For a single-user installation, you must have a user account on the system where you install IBM Integration Bus
  - For a shared installation, install as super user who has write access to the `/var` directory and then manually add users to the `mqbrkrs` security group
- On z/OS
  - User ID must have RACF privileges for SMP/E
  - User ID must have a valid OMVS segment because the product installs into the file system paths that are specified during the SMP/E APPLY processing

Figure 2-9. IBM Integration Bus security requirements WM6461.0

Notes:

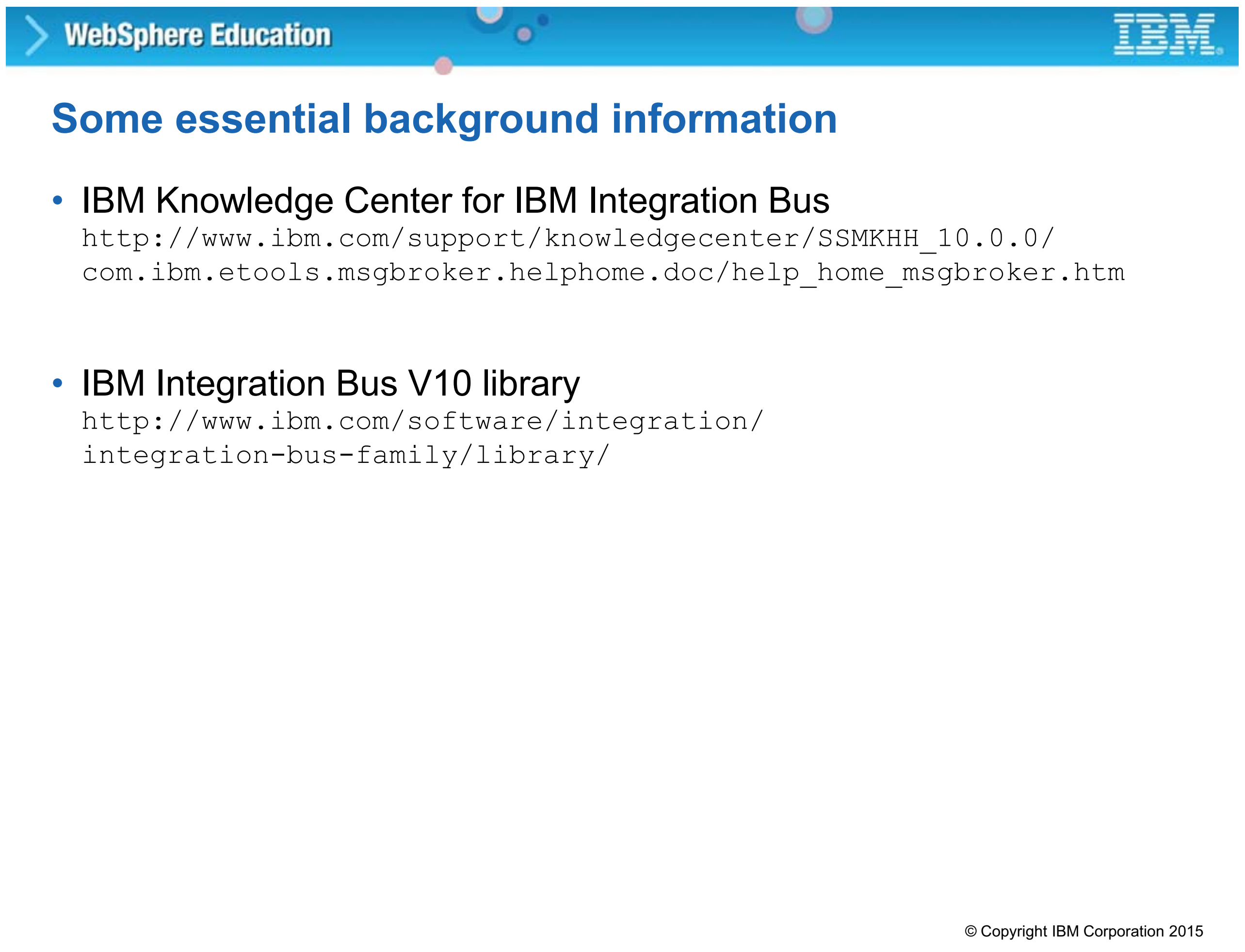
After you install Integration Bus, you must complete some extra configuration before users can access Integration Bus applications.

After you install IBM Integration Bus, a security group that is called `mqbrkrs` is created (if this group does not exist). To enable users to use Integration Bus, add the users to the `mqbrkrs` group by using the security facilities that are provided by your operating system.

To deploy a single-user installation of Integration Bus on Linux or UNIX systems, you must have a user account on the system where you install Integration Bus. The user account does not need super user access to the system. After the deployment, only you can use the Integration Bus installation.

To deploy a shared installation of Integration Bus on Linux or UNIX systems, you must log in as `root` or as a super user who has write access to the `/var` directory.

On z/OS, the user ID that you use during installation must also have suitable RACF privileges to install SMP/E in your environment. The user ID must have a valid OMVS segment because the product installs into the file system paths that are specified during the SMP/E APPLY processing.



Some essential background information

- IBM Knowledge Center for IBM Integration Bus
  `http://www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm`

- IBM Integration Bus V10 library
  `http://www.ibm.com/software/integration/integration-bus-family/library/`

Figure 2-10. Some essential background information WM6461.0

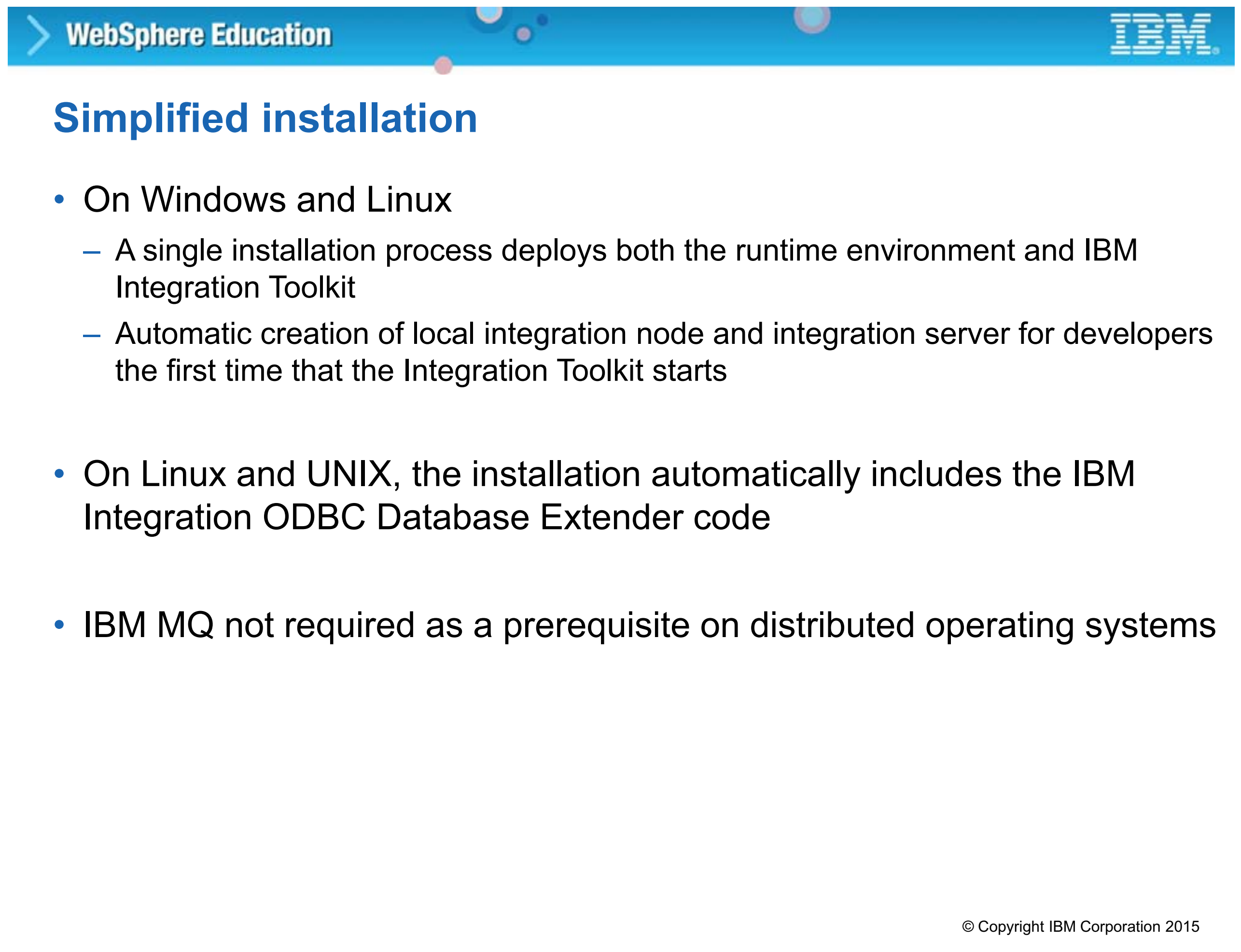
Notes:

IBM Integration Bus is constantly being improved. You should regularly monitor changes by checking the websites that are listed in the figure.

The product documentation can be installed on a Windows or Linux server, independent of the IBM Integration Toolkit.

2.2. Installation and configuration

This topic describes IBM Integration installation and basic configuration. It also describes commands that you can use to verify the installation.



WebSphere Education IBM

Simplified installation

- On Windows and Linux
  - A single installation process deploys both the runtime environment and IBM Integration Toolkit
  - Automatic creation of local integration node and integration server for developers the first time that the Integration Toolkit starts
- On Linux and UNIX, the installation automatically includes the IBM Integration ODBC Database Extender code
- IBM MQ not required as a prerequisite on distributed operating systems

© Copyright IBM Corporation 2015

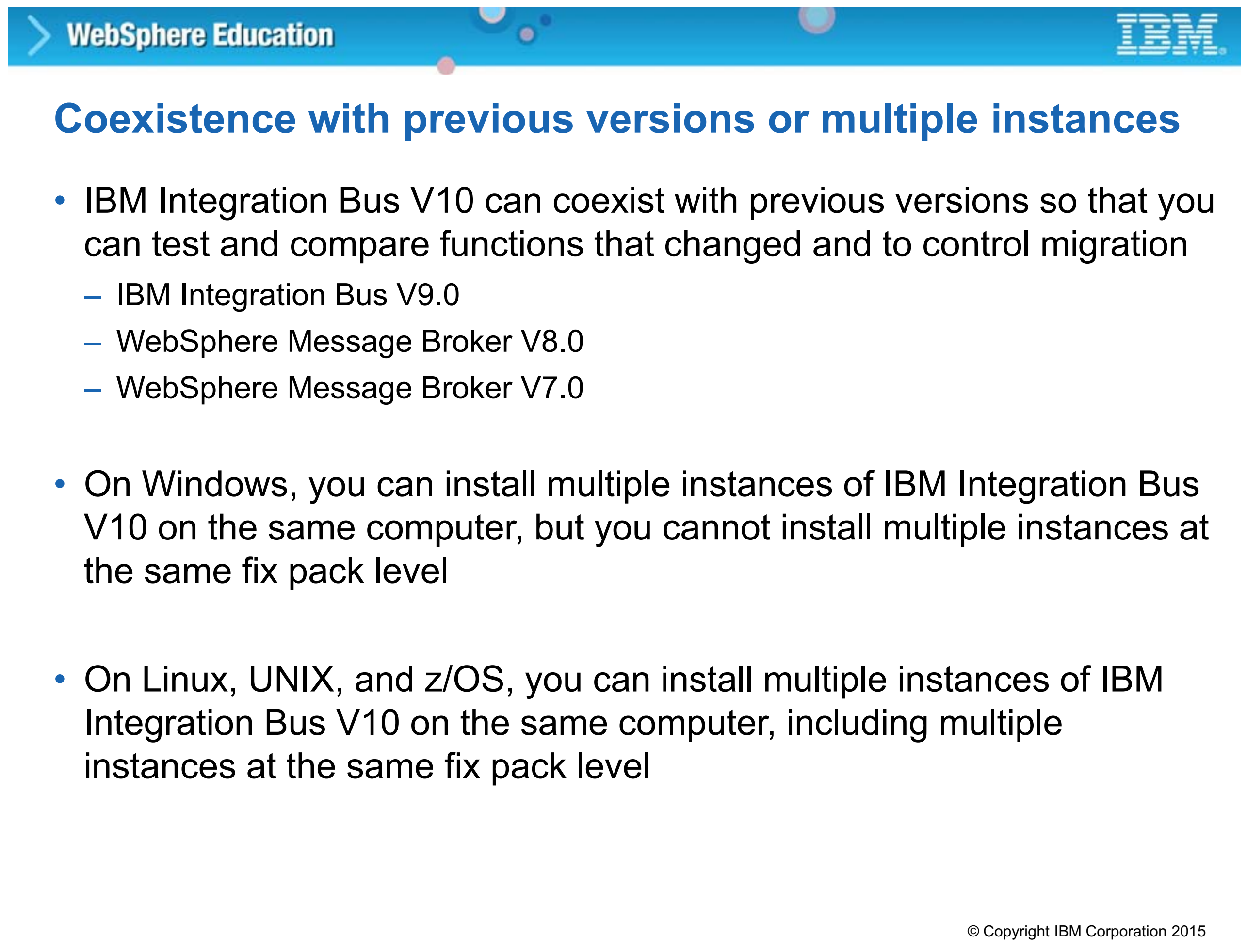
Figure 2-11. Simplified installation WM6461.0

Notes:

The process to install and configure IBM Integration Bus is simplified so that an integration developer can use the Integration Toolkit to start creating applications quicker.

The installation process includes the following improvements:

- On Windows and Linux, a single installation process deploys both the runtime environment and Integration Toolkit. On Windows, you run the installer, accept the license, and (optionally) change the location for the installation. On Linux, the installation is a simple unpack of the code into an appropriate directory.
- On Linux and UNIX, the installation automatically includes the IBM Integration ODBC Database Extender code; you do not have to install this SupportPac manually after you install IBM Integration Bus.
- IBM MQ is no longer a prerequisite for using Integration Bus on distributed operating system. You can develop and deploy applications independently of IBM MQ. You administer Integration Bus integration nodes by using the IBM Integration web user interface instead of the IBM MQ Explorer.



Coexistence with previous versions or multiple instances

- IBM Integration Bus V10 can coexist with previous versions so that you can test and compare functions that changed and to control migration
  - IBM Integration Bus V9.0
  - WebSphere Message Broker V8.0
  - WebSphere Message Broker V7.0
- On Windows, you can install multiple instances of IBM Integration Bus V10 on the same computer, but you cannot install multiple instances at the same fix pack level
- On Linux, UNIX, and z/OS, you can install multiple instances of IBM Integration Bus V10 on the same computer, including multiple instances at the same fix pack level

© Copyright IBM Corporation 2015

Figure 2-12. Coexistence with previous versions or multiple instances WM6461.0

Notes:

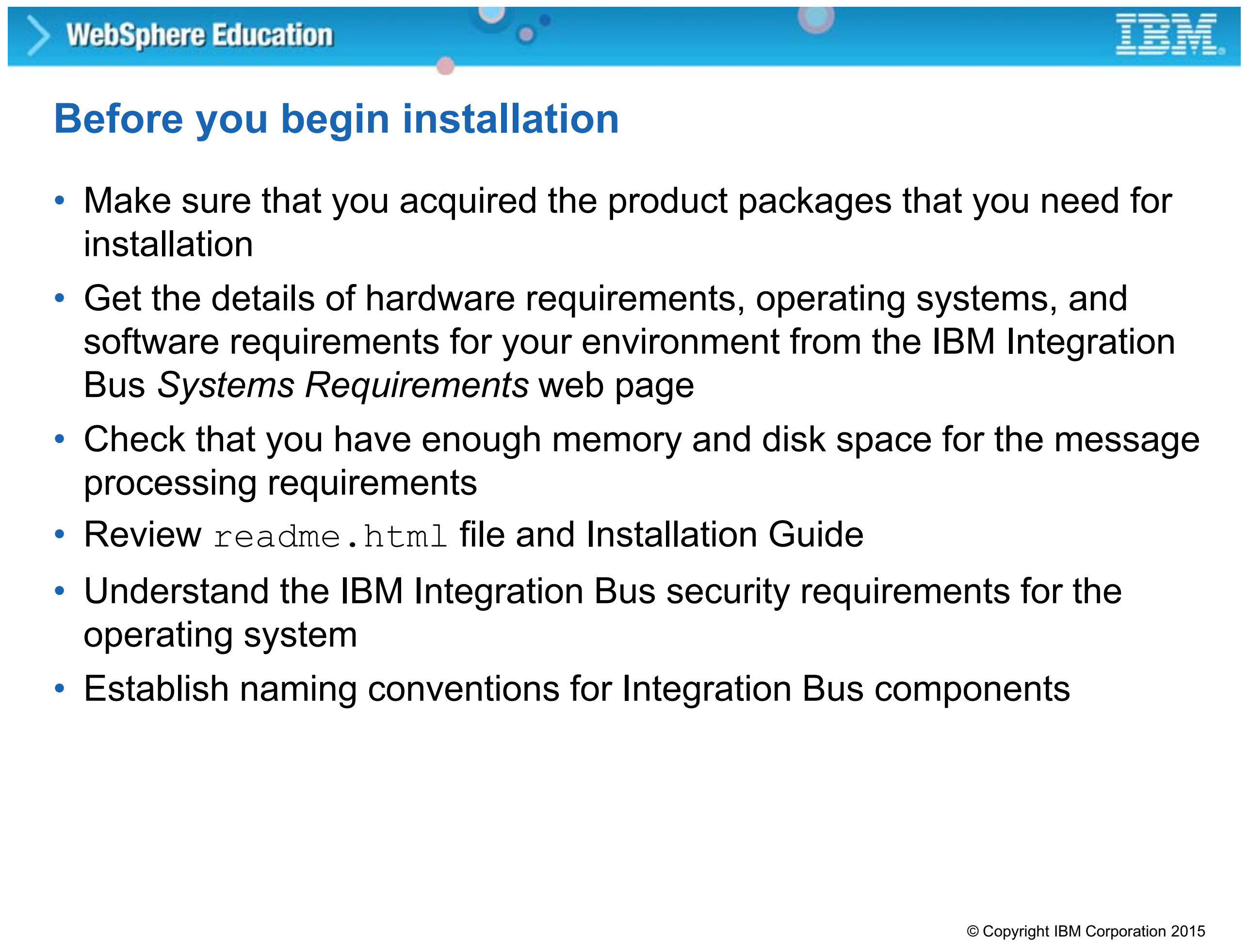
IBM Integration Bus Version V10.0 can coexist with previous versions so that you can test and compare functions that changed. You can then control your migration to the newer version of Integration Bus by migrating individual integration nodes one at a time, rather than by migrating all the integration nodes at the same time.

Different versions of IBM Integration Bus or WebSphere Message Broker can coexist and can run independently, and you can migrate integration nodes from one version to another.

The following restrictions apply to coexistence on Windows:

On Windows, Linux, and UNIX, you can install multiple instances of Integration Bus on the same computer. On Windows, you cannot install multiple instances of a product version at the same fix pack level. For example, you can install Version 10.0.0.1 on the same computer as Version 10.0.0.0, but you cannot install two instances of Version 10.0.0.1 on the same computer.

If you have more than one installation on a single computer, you must ensure that the commands that you issue to integration nodes on that computer are using the correct version of installed code.



WebSphere Education

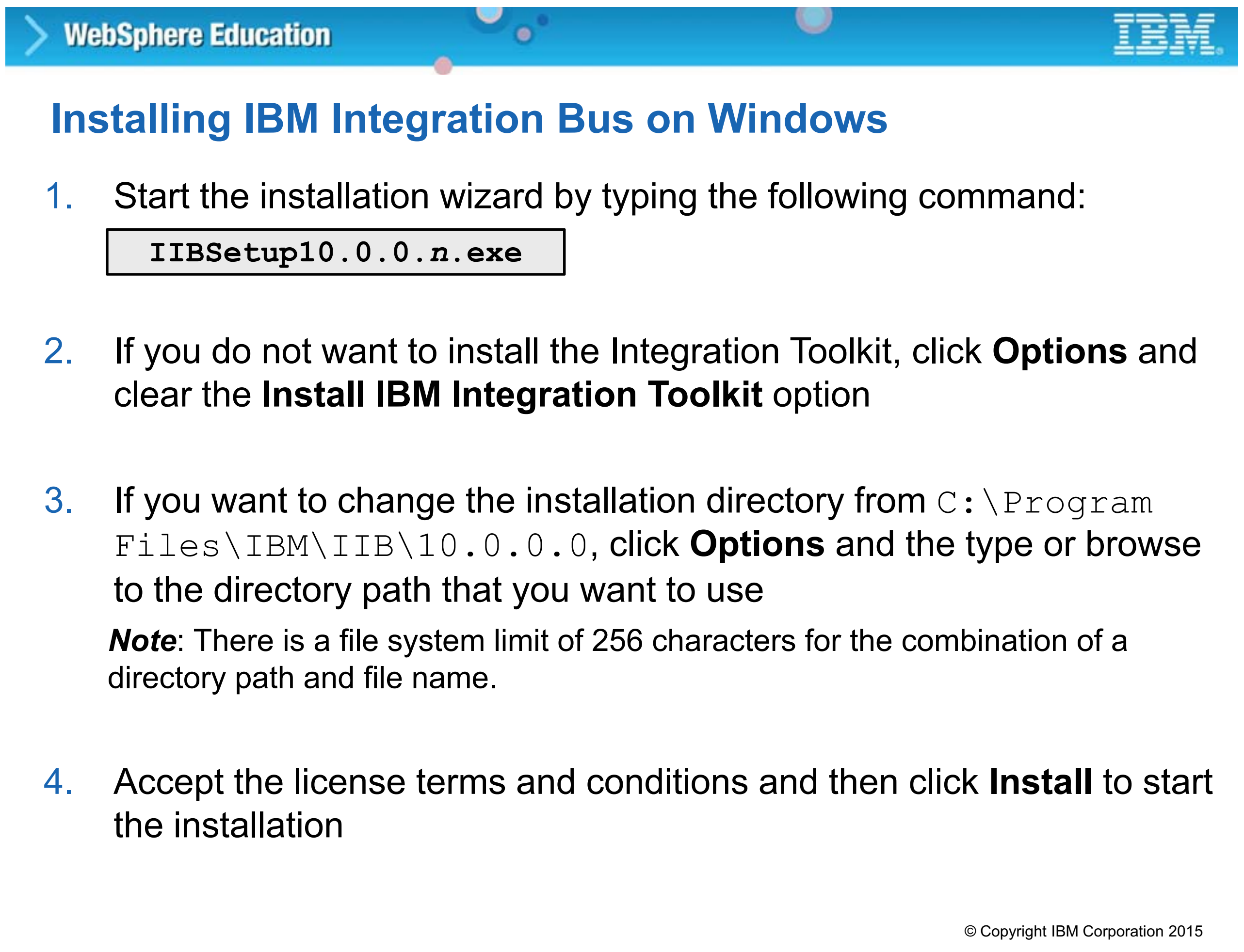
Before you begin installation

- Make sure that you acquired the product packages that you need for installation
- Get the details of hardware requirements, operating systems, and software requirements for your environment from the IBM Integration Bus *Systems Requirements* web page
- Check that you have enough memory and disk space for the message processing requirements
- Review `readme.html` file and Installation Guide
- Understand the IBM Integration Bus security requirements for the operating system
- Establish naming conventions for Integration Bus components

Figure 2-13. Before you begin installation WM6461.0

Notes:

Before you begin the Integration Bus installation, ensure that your environment meets the hardware and software prerequisites.



WebSphere Education

Installing IBM Integration Bus on Windows

1. Start the installation wizard by typing the following command:

```
IIBSetup10.0.0.n.exe
```

2. If you do not want to install the Integration Toolkit, click **Options** and clear the **Install IBM Integration Toolkit** option

3. If you want to change the installation directory from `C:\Program Files\IBM\IIB\10.0.0.0`, click **Options** and the type or browse to the directory path that you want to use

   Note: There is a file system limit of 256 characters for the combination of a directory path and file name.

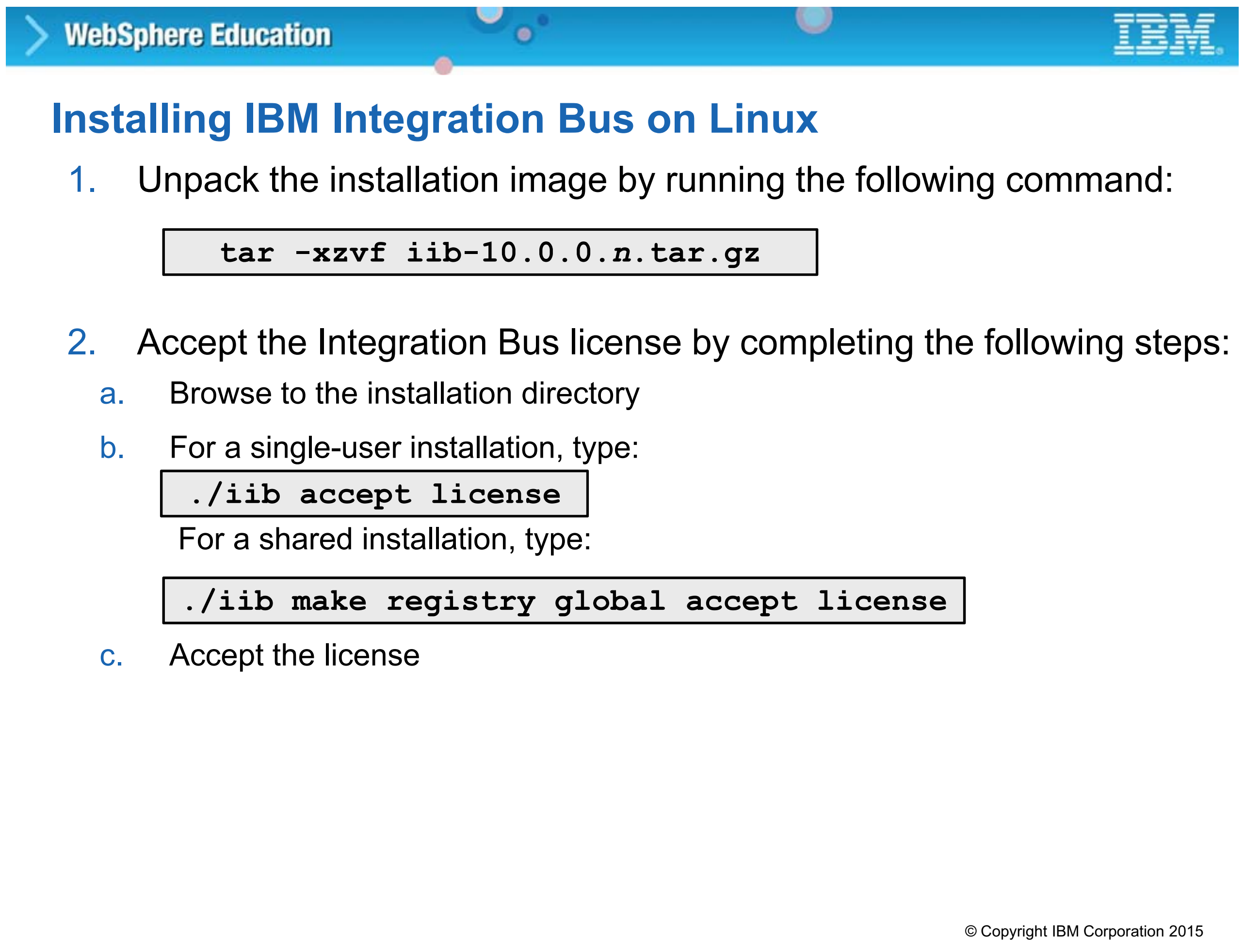
4. Accept the license terms and conditions and then click **Install** to start the installation

Figure 2-14. Installing IBM Integration Bus on Windows WM6461.0

Notes:

This figure lists the steps for installing Integration Bus on Windows.

By default, an installation on Windows includes the Integration Toolkit integrated development environment.



WebSphere Education IBM

Installing IBM Integration Bus on Linux

1. Unpack the installation image by running the following command:

```
tar -xzvf iib-10.0.0.n.tar.gz
```

2. Accept the Integration Bus license by completing the following steps:
   a. Browse to the installation directory
   b. For a single-user installation, type:

```
./iib accept license
```

   For a shared installation, type:

```
./iib make registry global accept license
```

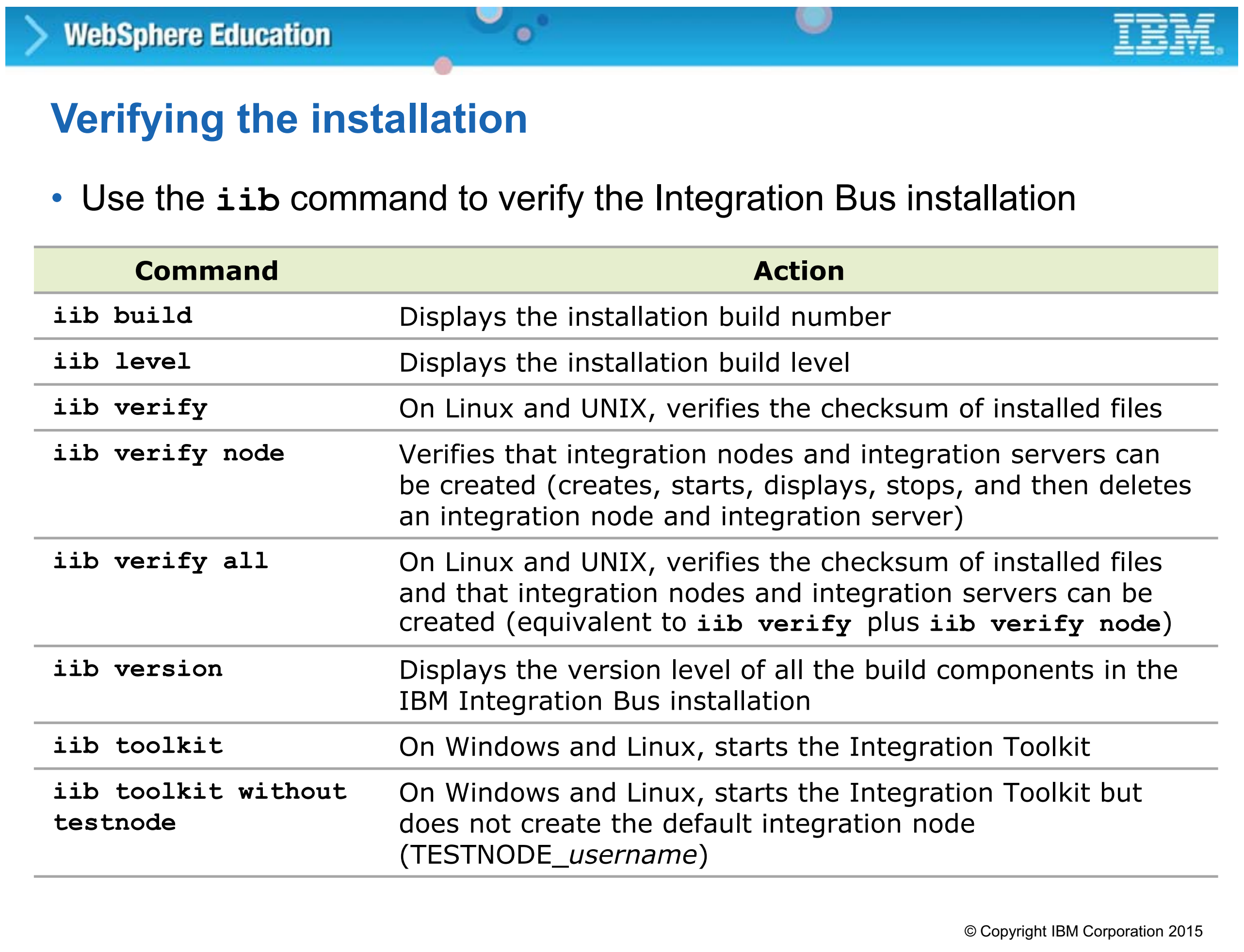
   c. Accept the license

Figure 2-15. Installing IBM Integration Bus on Linux WM6461.0

Notes:

This figure lists the steps for installing Integration Bus on Linux.

For Linux and UNIX, you can install Integration Bus as a single-user installation or a shared installation.



WebSphere Education IBM.

Verifying the installation

- Use the **`iib`** command to verify the Integration Bus installation

Command	Action
`iib build`	Displays the installation build number
`iib level`	Displays the installation build level
`iib verify`	On Linux and UNIX, verifies the checksum of installed files
`iib verify node`	Verifies that integration nodes and integration servers can be created (creates, starts, displays, stops, and then deletes an integration node and integration server)
`iib verify all`	On Linux and UNIX, verifies the checksum of installed files and that integration nodes and integration servers can be created (equivalent to **`iib verify`** plus **`iib verify node`**)
`iib version`	Displays the version level of all the build components in the IBM Integration Bus installation
`iib toolkit`	On Windows and Linux, starts the Integration Toolkit
`iib toolkit without testnode`	On Windows and Linux, starts the Integration Toolkit but does not create the default integration node (TESTNODE_*username*)

Figure 2-16. Verifying the installation WM6461.0

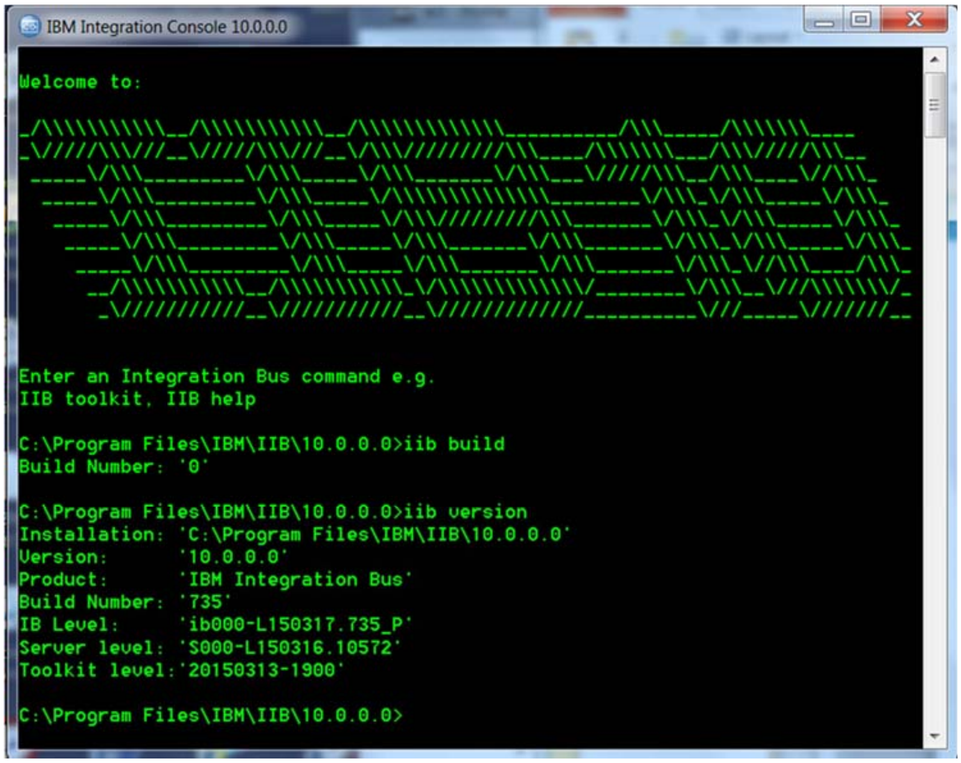
Notes:

You can use the **`iib`** command to complete tasks that are associated with the installation of Integration Bus such verifying build number and build levels.

You can run the **`iib`** command with the parameters that are listed in the table.

WebSphere Education

iib command examples



IBM Integration Console 10.0.0.0

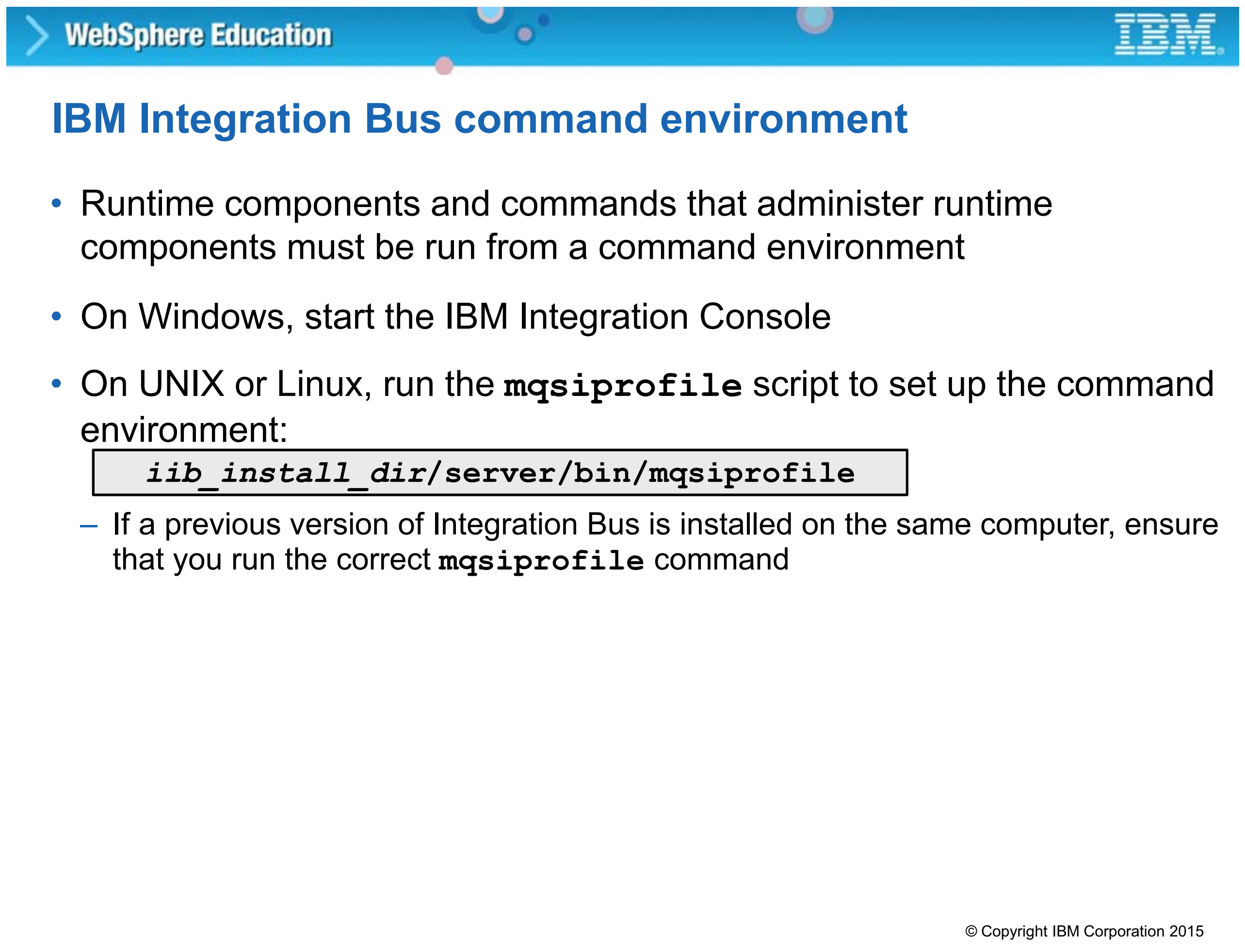
```
Welcome to:

_/\\\\\\\\\\\__/\\\\\\\\\\\__/\\\\\\\\\\\\\__________/\\\_____/\\\\\\\____
_\////\\\///__\////\\\///__\/\\\/////////\\\____/\\\\\\\___/\\\/////\\\__
 _____\/\\\_________\/\\\_____\/\\\_______\/\\\__\/////\\\__/\\\____\//\\\_
  _____\/\\\_________\/\\\_____\/\\\\\\\\\\\\\\__________\/\\\_\/\\\_____\/\\\_
   _____\/\\\_________\/\\\_____\/\\\/////////\\\_________\/\\\_\/\\\_____\/\\\_
    _____\/\\\_________\/\\\_____\/\\\_______\/\\\_________\/\\\_\/\\\_____\/\\\_
     _____\/\\\_________\/\\\_____\/\\\_______\/\\\_________\/\\\_\//\\\____/\\\_
      __/\\\\\\\\\\\__/\\\\\\\\\\\_\/\\\\\\\\\\\\\/__________\/\\\__\///\\\\\\\/_
       _\///////////__\///////////__\/////////////____________\///_____\///////__

Enter an Integration Bus command e.g.
IIB toolkit, IIB help

C:\Program Files\IBM\IIB\10.0.0.0>iib build
Build Number: '0'

C:\Program Files\IBM\IIB\10.0.0.0>iib version
Installation: 'C:\Program Files\IBM\IIB\10.0.0.0'
Version:      '10.0.0.0'
Product:      'IBM Integration Bus'
Build Number: '735'
IB Level:     'ib000-L150317.735_P'
Server level: 'S000-L150316.10572'
Toolkit level:'20150313-1900'

C:\Program Files\IBM\IIB\10.0.0.0>
```

Figure 2-17. `iib command examples` WM6461.0

Notes:

The figure shows examples of the **`iib build`** and the **`iib version`** commands.

Integration Bus commands must run in an IBM Integration Console environment. This example also shows the IBM Integration Console.



WebSphere Education

IBM Integration Bus command environment

- Runtime components and commands that administer runtime components must be run from a command environment
- On Windows, start the IBM Integration Console
- On UNIX or Linux, run the **`mqsiprofile`** script to set up the command environment:

```
iib_install_dir/server/bin/mqsiprofile
```

  - If a previous version of Integration Bus is installed on the same computer, ensure that you run the correct **`mqsiprofile`** command

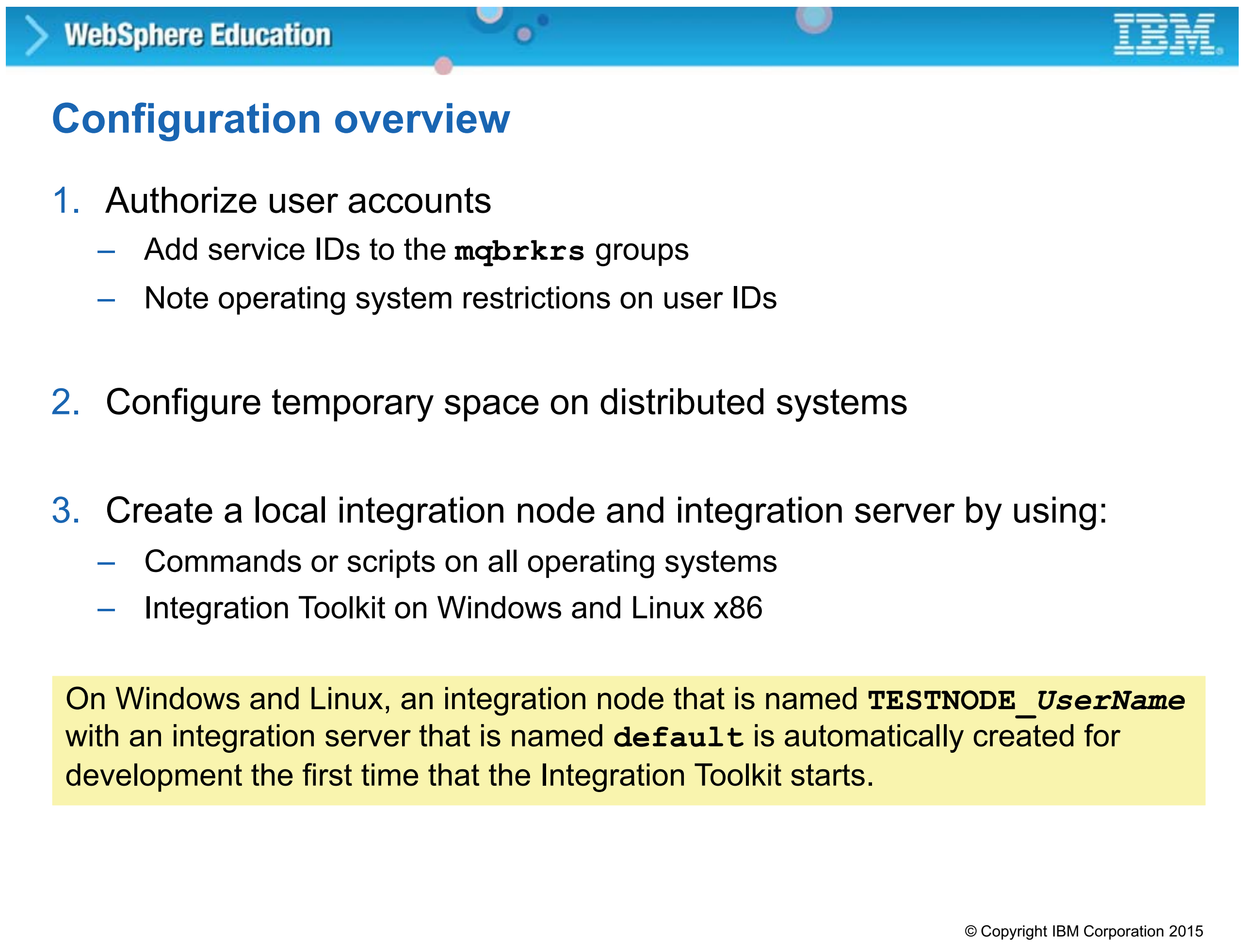
© Copyright IBM Corporation 2015

Figure 2-18. IBM Integration Bus command environment WM6461.0

Notes:

Runtime components, and commands that administer runtime components, must be run from a special command environment. This command environment is initialized by running the **`mqsiprofile`** command or on Windows by starting the IBM Integration Console.

If you have a previous version of Integration Bus on the same computer, ensure that you run the correct profile before you use Version 10.0. The **`mqsiprofile`** command places the Version 10.0 commands and libraries at the front of your search path, and can override any combination of PATH, CLASSPATH, or library PATH.



Configuration overview

1. Authorize user accounts
   – Add service IDs to the **`mqbrkrs`** groups
   – Note operating system restrictions on user IDs

2. Configure temporary space on distributed systems

3. Create a local integration node and integration server by using:
   – Commands or scripts on all operating systems
   – Integration Toolkit on Windows and Linux x86

On Windows and Linux, an integration node that is named **`TESTNODE_`*`UserName`*** with an integration server that is named **`default`** is automatically created for development the first time that the Integration Toolkit starts.

Figure 2-19. Configuration overview WM6461.0

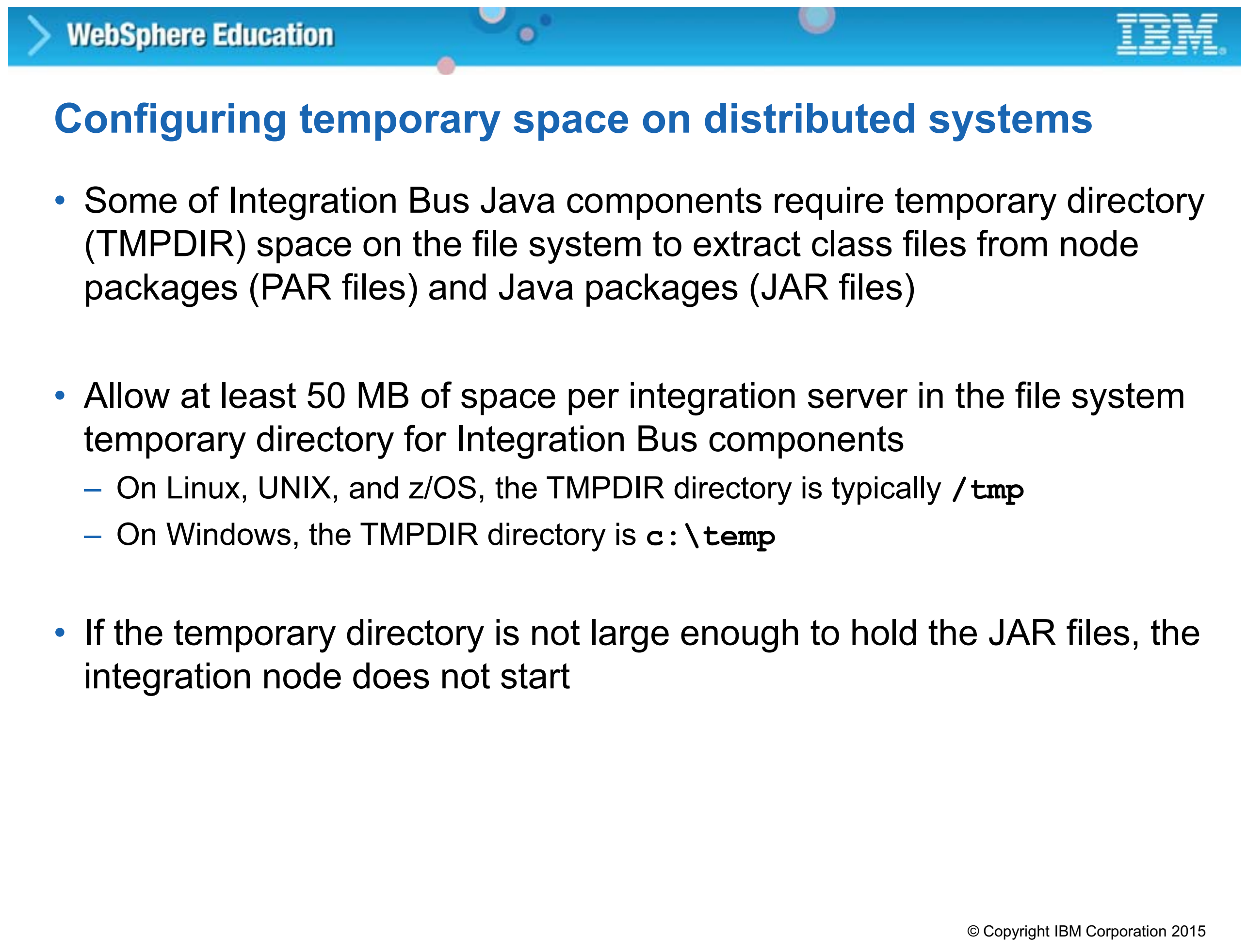
Notes:

More administrative actions must be completed before the Integration Bus can be used.

Administrators and authorized users must be added to the appropriate security groups and granted authorization to deploy and manage integration nodes and integration node artifacts. Some security setup steps occur automatically during installation. For example, on Windows the `mqbrkrs` security group is defined, and the installing user ID is added to that group.

Some Integration Bus components require temporary disk space when installed on distributed operating systems such as Windows and Linux. After installation, It might be necessary to configure temporary space.

After you complete the installation, you can define integration nodes and integration servers. If the Integration Toolkit is installed, an integration node and integration server are automatically created for the development environment.



Configuring temporary space on distributed systems

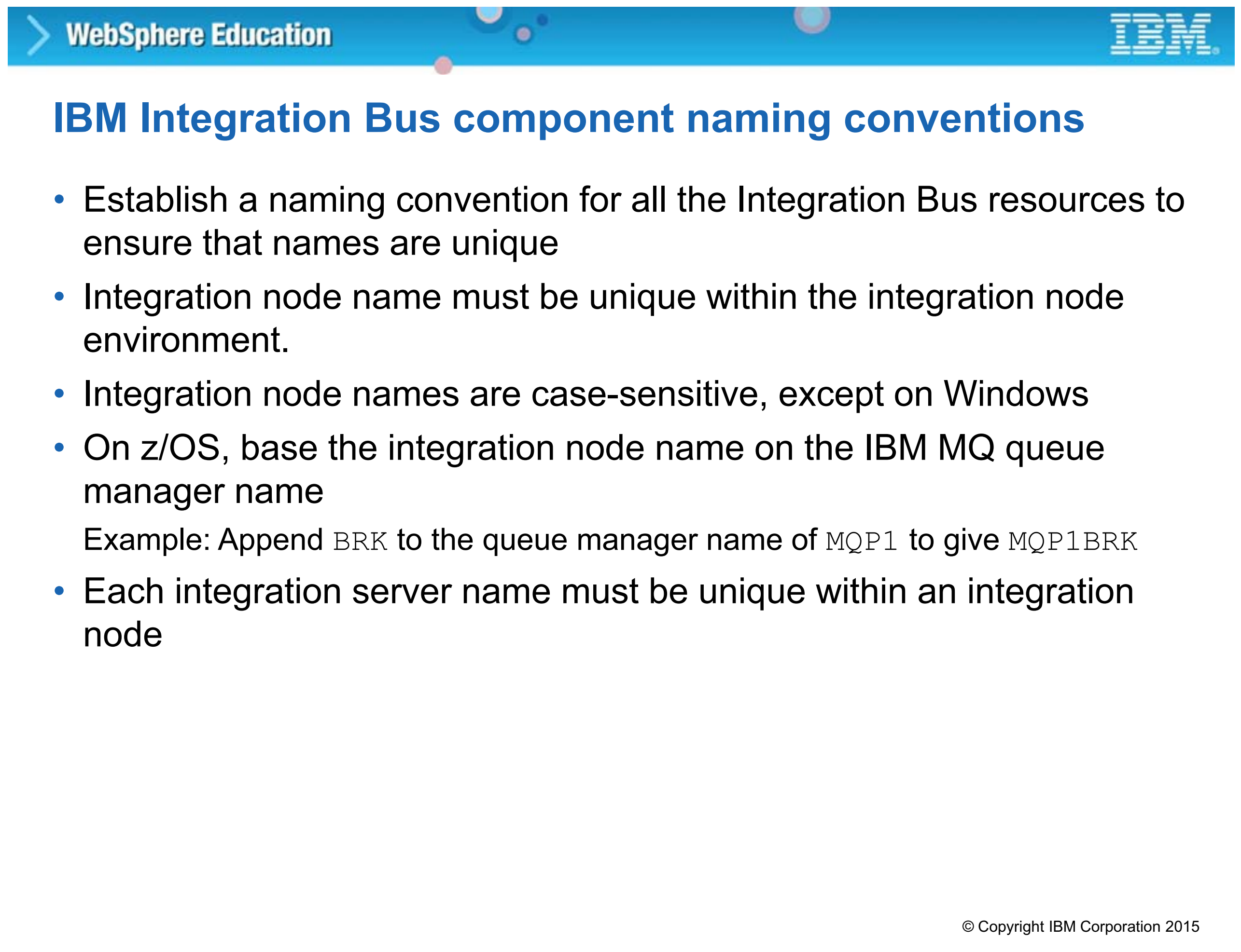
- Some of Integration Bus Java components require temporary directory (TMPDIR) space on the file system to extract class files from node packages (PAR files) and Java packages (JAR files)
- Allow at least 50 MB of space per integration server in the file system temporary directory for Integration Bus components
  - On Linux, UNIX, and z/OS, the TMPDIR directory is typically `/tmp`
  - On Windows, the TMPDIR directory is `c:\temp`
- If the temporary directory is not large enough to hold the JAR files, the integration node does not start

Figure 2-20. Configuring temporary space on distributed systems WM6461.0

Notes:

Integration Bus contains Java components. Some of these components require temporary directory space on the file system to extract class files from node packages (PAR files) and Java packages (JAR files). The Integration Bus Java Runtime Environment also creates files in the temporary directory to support debug facilities.

For correct operation, allow at least 50 MB of space per integration server in the file system temporary directory for Integration Bus components. This space is required in addition to any space required for operation of user developed artifacts. Use of JavaCompute nodes, Java user-defined nodes, or extra plug-in nodes that are implemented in Java might require further temporary directory space.



WebSphere Education

IBM Integration Bus component naming conventions

- Establish a naming convention for all the Integration Bus resources to ensure that names are unique
- Integration node name must be unique within the integration node environment.
- Integration node names are case-sensitive, except on Windows
- On z/OS, base the integration node name on the IBM MQ queue manager name

  Example: Append `BRK` to the queue manager name of `MQP1` to give `MQP1BRK`
- Each integration server name must be unique within an integration node

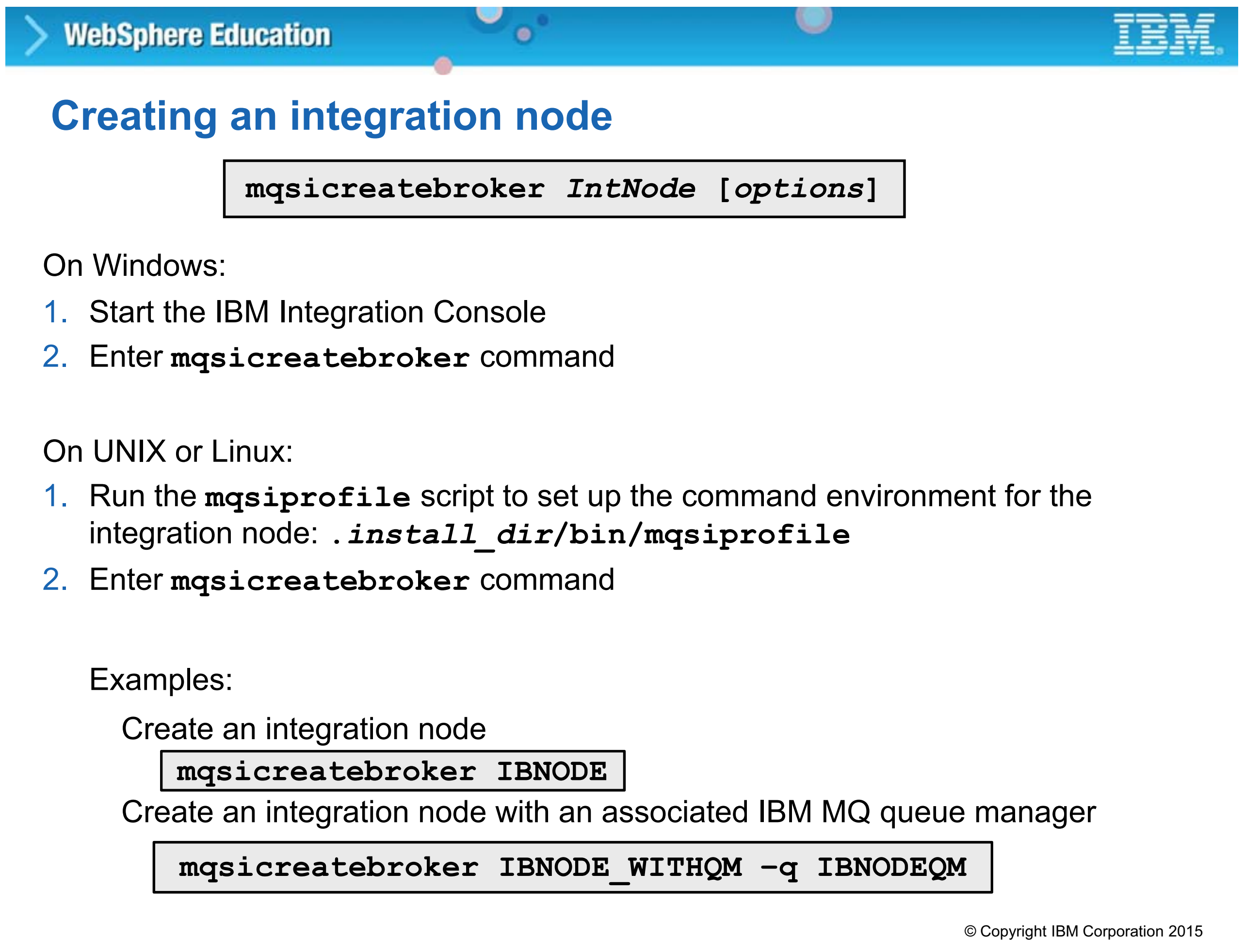
© Copyright IBM Corporation 2015

Figure 2-21. IBM Integration Bus component naming conventions WM6461.0

Notes:

Establish a naming convention for all the Integration Bus resources to ensure that names are unique, and that users that create resources can be confident of not introducing duplication or confusion.

On z/OS, each integration node requires its own queue manager. Base your integration node name on the queue manager name. For example, append `BRK` to the queue manager name of `MQP1`, to give `MQP1BRK`. This naming convention has the following advantages: It is easy to associate the integration node with the queue manager because they both begin with the same characters.



WebSphere Education

Creating an integration node

```
mqsicreatebroker IntNode [options]
```

On Windows:

1. Start the IBM Integration Console
2. Enter **`mqsicreatebroker`** command

On UNIX or Linux:

1. Run the **`mqsiprofile`** script to set up the command environment for the integration node: **`.`*`install_dir`*`/bin/mqsiprofile`**
2. Enter **`mqsicreatebroker`** command

Examples:

Create an integration node

```
mqsicreatebroker IBNODE
```

Create an integration node with an associated IBM MQ queue manager

```
mqsicreatebroker IBNODE_WITHQM -q IBNODEQM
```

© Copyright IBM Corporation 2015

Figure 2-22. Creating an integration node WM6461.0

Notes:

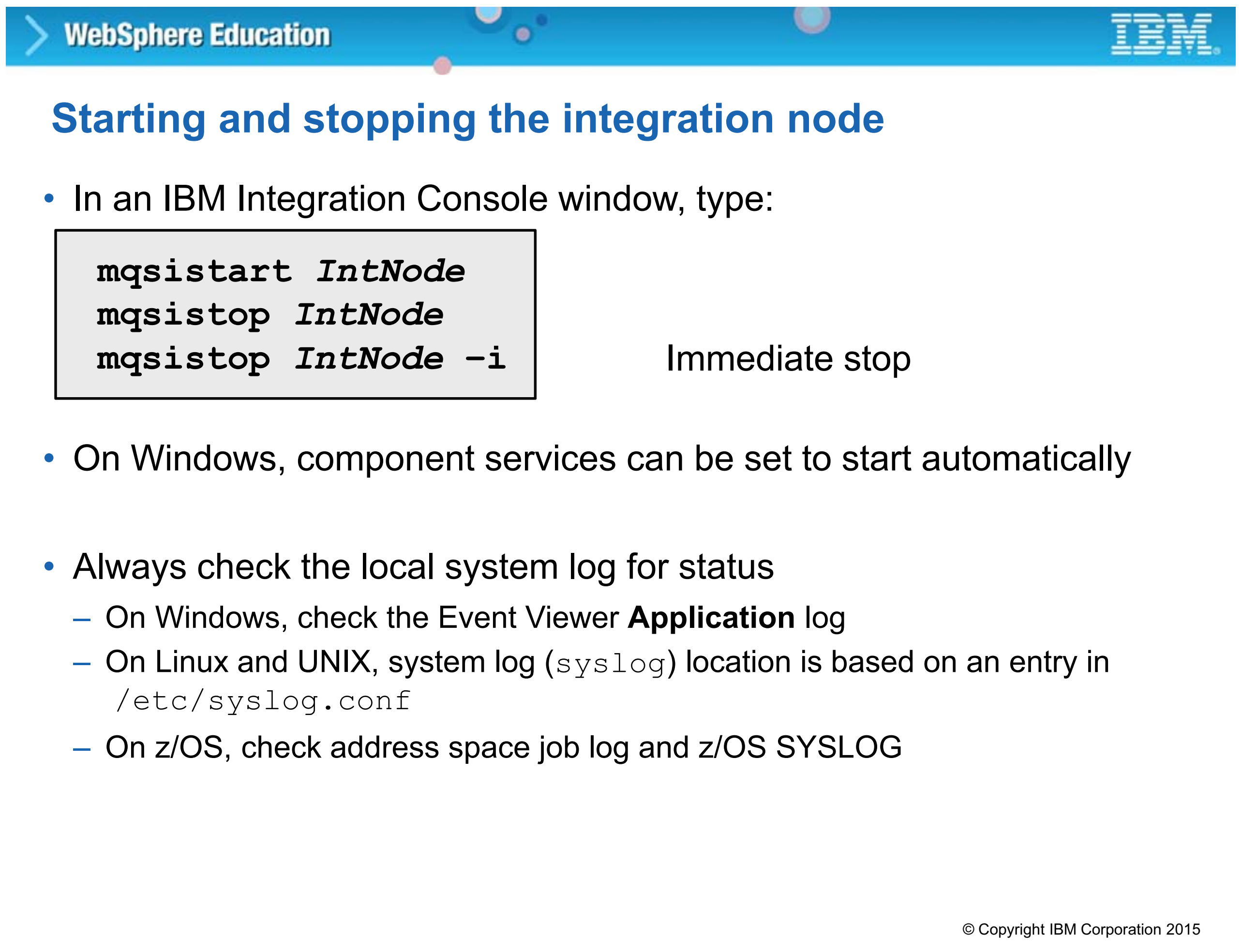
The `mqsicreatebroker` command can be used to create an integration node and its associated resources.

In the first example, the `mqsicreatebroker` command creates an integration node that is named `IBNODE`.

In the second example, the `mqsicreatebroker` command creates an integration node that is named `IBNODE_WITHQM` that is associated with an IBM MQ queue manager that is named `IBNODEQM` (`-q`).

For a detailed description of the `mqsicreatebroker` command and command syntax options, see the IBM Knowledge Center.

You can check the existence of the integration node with the `mqsilist` command. You can obtain a report of the details of the integration node with `mqsiservice` command. Both of these commands are described later in this unit.



WebSphere Education

Starting and stopping the integration node

- In an IBM Integration Console window, type:

```
mqsistart IntNode
mqsistop IntNode
mqsistop IntNode -i
```

Immediate stop

- On Windows, component services can be set to start automatically
- Always check the local system log for status
  - On Windows, check the Event Viewer **Application** log
  - On Linux and UNIX, system log (`syslog`) location is based on an entry in `/etc/syslog.conf`
  - On z/OS, check address space job log and z/OS SYSLOG

Figure 2-23. Starting and stopping the integration node WM6461.0

Notes:

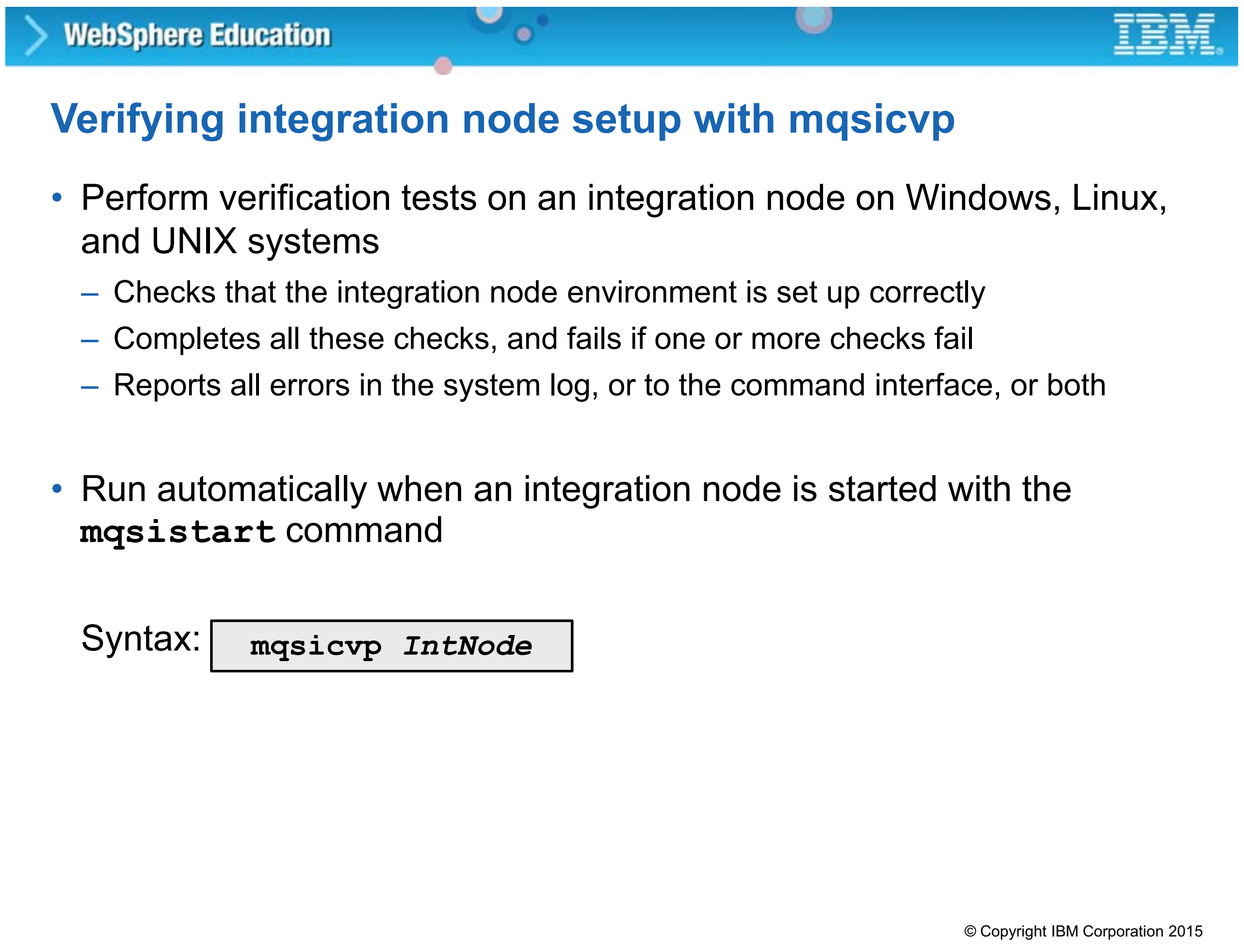
You can start and stop the integration node by using the `mqsistart` and `mqsistop` command. The `mqsistart` and `mqsistop` commands are supported on Windows, Linux, UNIX, and z/OS.

The `mqsistart` command starts the queue manager on distributed systems, and verifies that the integration node environment is set up correctly.

In Windows, the integration node can be started from the Windows **Services** control pane. It is useful to set it to start automatically when the Windows server starts. If you defined the integration node with a queue manager, verify that the *IBM MQSeries* service is active.

Use the `mqsistop` command to stop an integration node. You can specify the `-i` option if you already tried, and failed, to stop the integration node in a controlled fashion.

Review the system event log messages if any errors occur while processing the `mqsistart` or `mqsistop` commands.



WebSphere Education

Verifying integration node setup with mqsicvp

- Perform verification tests on an integration node on Windows, Linux, and UNIX systems
  - Checks that the integration node environment is set up correctly
  - Completes all these checks, and fails if one or more checks fail
  - Reports all errors in the system log, or to the command interface, or both
- Run automatically when an integration node is started with the **`mqsistart`** command

Syntax:

```
mqsicvp IntNode
```

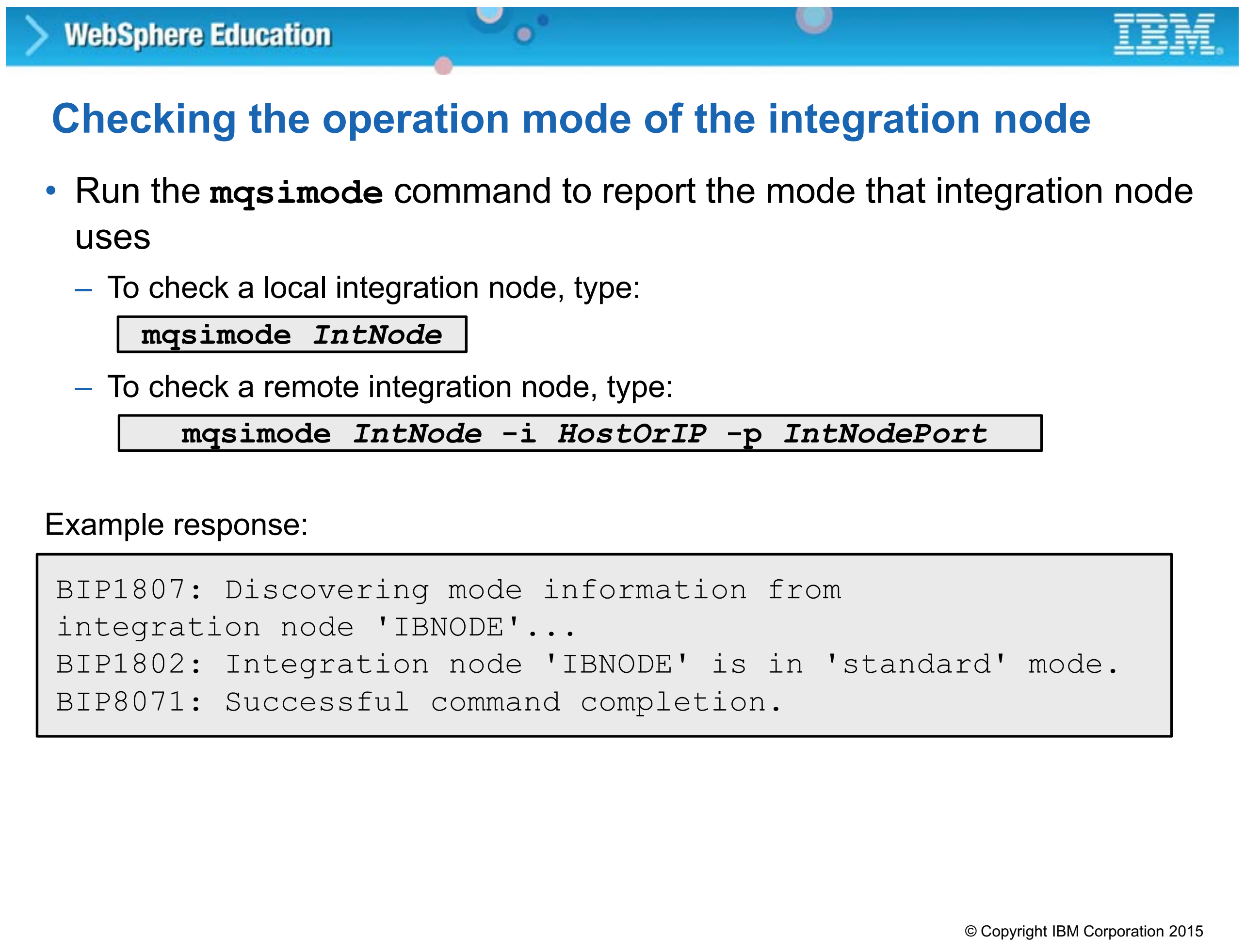
Figure 2-24. Verifying integration node setup with mqsicvp WM6461.0

Notes:

Use the `mqsicvp` command to run verification tests on an integration node before you create integration servers.

The `mqsicvp` command checks that the integration node environment is set up correctly; for example, that the installed level of Java is supported.

You can run this command against an integration node that is running, or is not running. If the integration node is not running, the verification tests are done, but the integration node is not started.



WebSphere Education IBM

Checking the operation mode of the integration node

- Run the **`mqsimode`** command to report the mode that integration node uses
  - To check a local integration node, type:

```
mqsimode IntNode
```

  - To check a remote integration node, type:

```
mqsimode IntNode -i HostOrIP -p IntNodePort
```

Example response:

```
BIP1807: Discovering mode information from
integration node 'IBNODE'...
BIP1802: Integration node 'IBNODE' is in 'standard' mode.
BIP8071: Successful command completion.
```

© Copyright IBM Corporation 2015

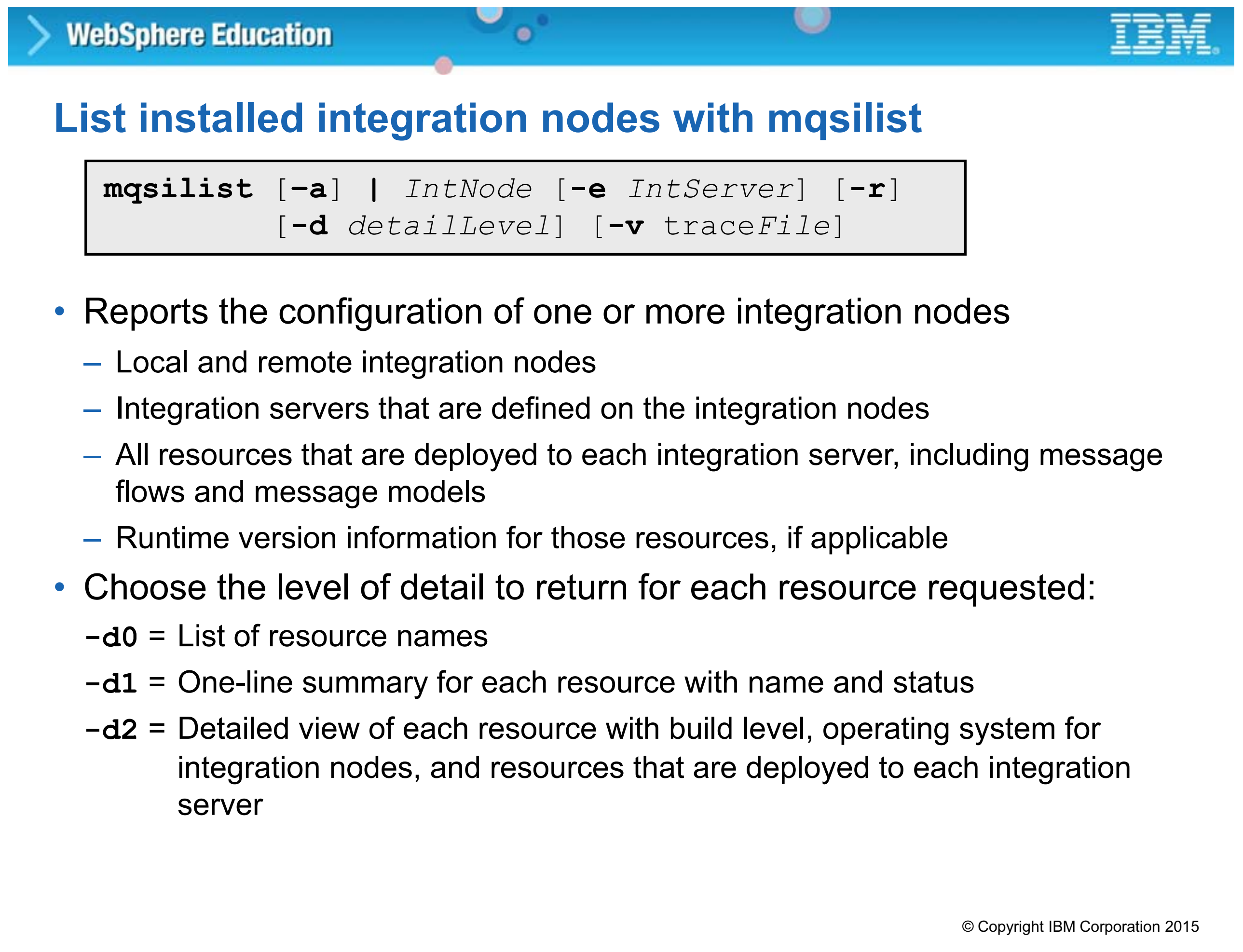
Figure 2-25. Checking the operation mode of the integration node WM6461.0

Notes:

Use the `mqsimode` command to configure and retrieve operation mode information.

You can also use the `mqsimode` command with the `-o` option to change the operation mode of an integration node. Valid values are advanced (the full edition), `standard` (Standard Edition), express (Express Edition), `scale` (Scale mode), and `adapter` (Remote Adapter Deployment mode). The default value is advanced, unless you have the Developer Edition, in which case the default value is `developer`.

Operation modes are described in detail in Unit 1.



WebSphere Education

List installed integration nodes with mqsilist

```
mqsilist [-a] | IntNode [-e IntServer] [-r]
         [-d detailLevel] [-v traceFile]
```

- Reports the configuration of one or more integration nodes
  - Local and remote integration nodes
  - Integration servers that are defined on the integration nodes
  - All resources that are deployed to each integration server, including message flows and message models
  - Runtime version information for those resources, if applicable
- Choose the level of detail to return for each resource requested:

  -d0 = List of resource names

  -d1 = One-line summary for each resource with name and status

  -d2 = Detailed view of each resource with build level, operating system for integration nodes, and resources that are deployed to each integration server

© Copyright IBM Corporation 2015

Figure 2-26. List installed integration nodes with mqsilist WM6461.0

Notes:

Use the `mqsilist` command to check for the existence of IBM Integration Bus components.

The figure shows the syntax for the `mqsilist` command. In the figure, optional parameters are enclosed in brackets [ ].

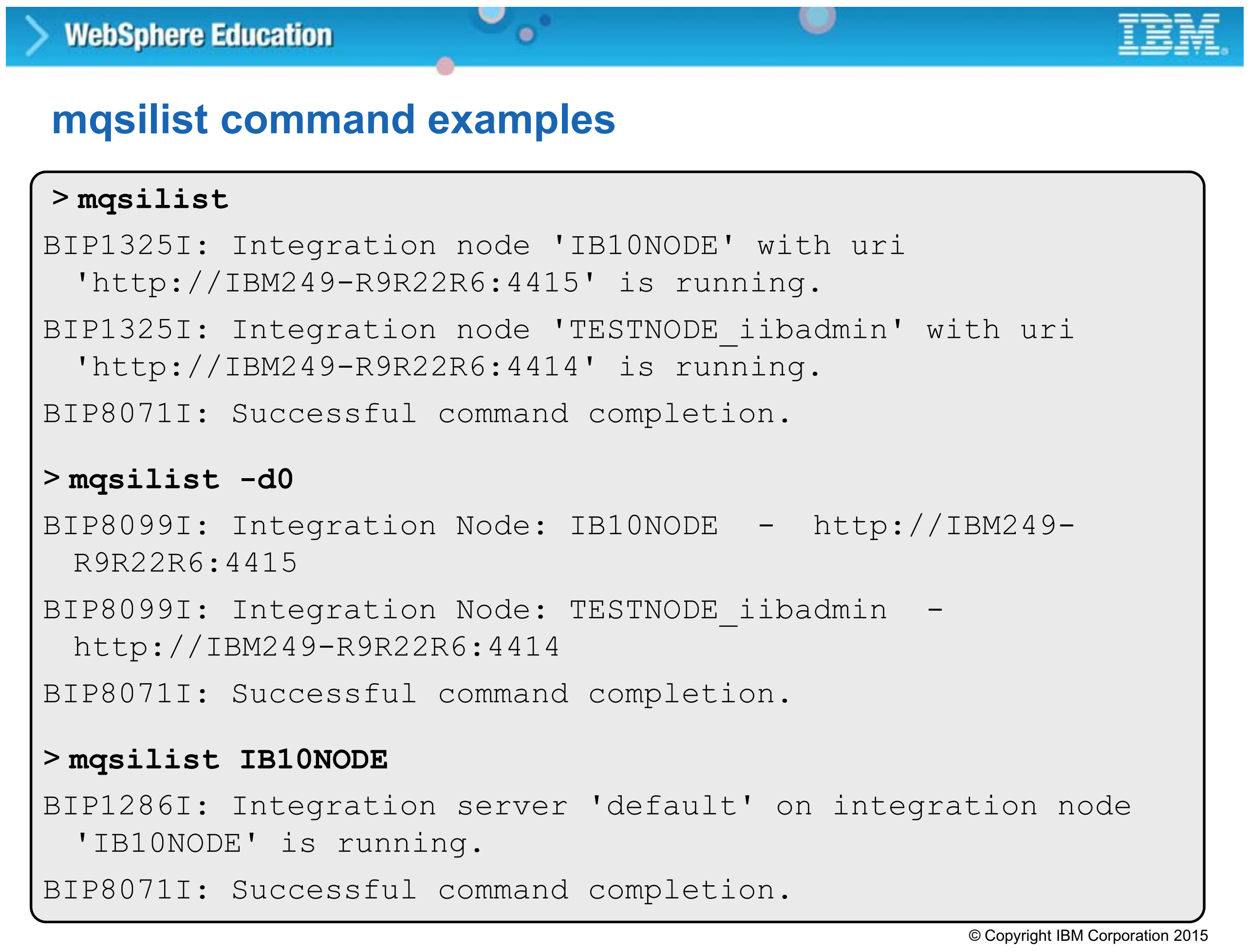
The `-a` option lists all the integration nodes that are installed on the local computer, in all installations.

The `-r` option causes the command to run recursively and display information about subcomponents.

The `-d` option specifies the type of detail to provide:

> `-d0` returns only the integration node name and the names of their associated queue managers.
>
> `-d1` returns a one line summary of each resource.
>
> `-d2` returns detailed information about each resource.

For more information about the `mqsilist` command syntax and options, see the Integration Bus product documentation.



mqsilist command examples

```
> mqsilist
BIP1325I: Integration node 'IB10NODE' with uri
  'http://IBM249-R9R22R6:4415' is running.
BIP1325I: Integration node 'TESTNODE_iibadmin' with uri
  'http://IBM249-R9R22R6:4414' is running.
BIP8071I: Successful command completion.

> mqsilist -d0
BIP8099I: Integration Node: IB10NODE  -  http://IBM249-
  R9R22R6:4415
BIP8099I: Integration Node: TESTNODE_iibadmin  -
  http://IBM249-R9R22R6:4414
BIP8071I: Successful command completion.

> mqsilist IB10NODE
BIP1286I: Integration server 'default' on integration node
  'IB10NODE' is running.
BIP8071I: Successful command completion.
```

Figure 2-27. `mqsilist command examples` WM6461.0

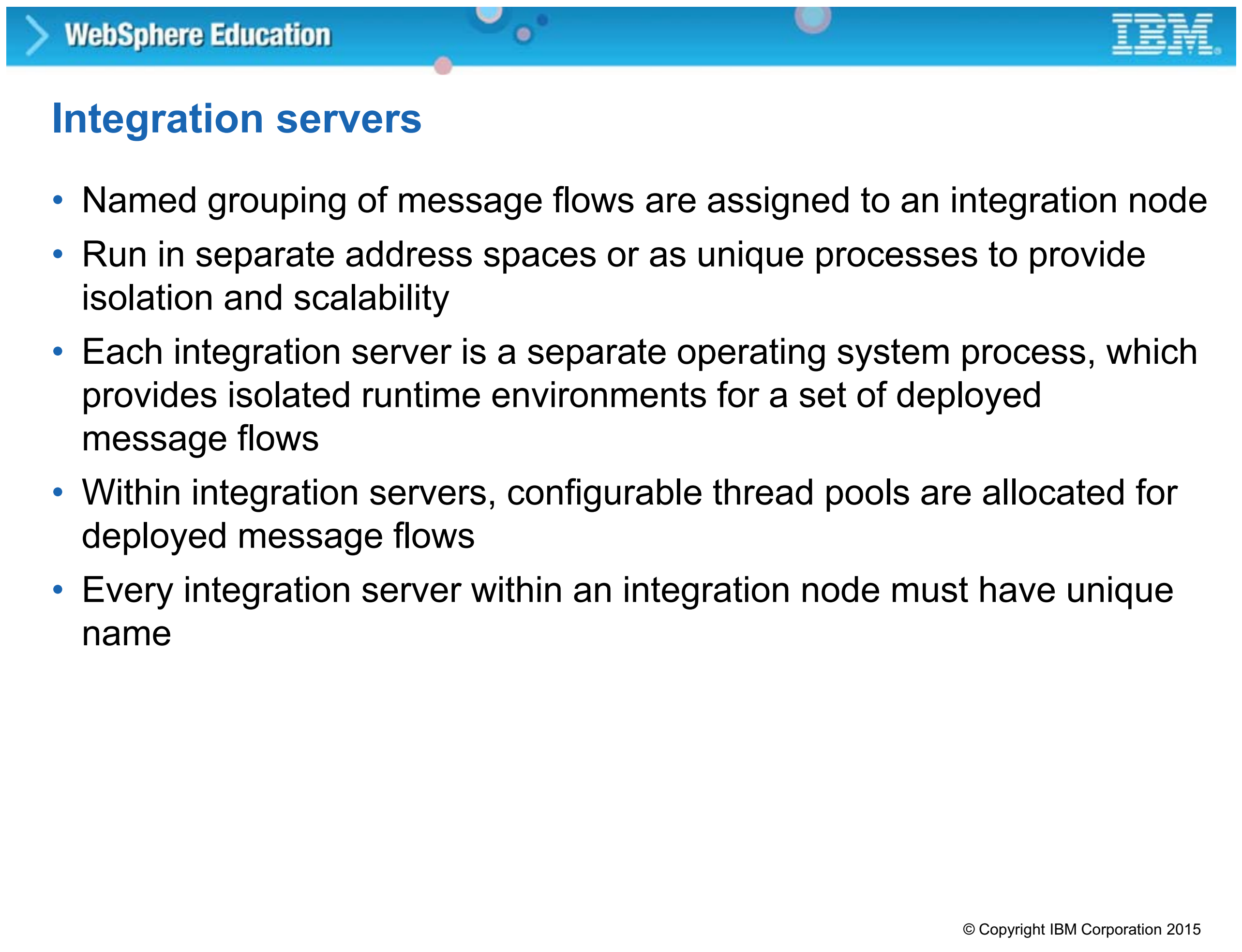
Notes:

The figure shows some examples of the `mqsilist` command and the results.

In the first example, the `mqsilist` command is run without any options. It reports the status of every integration node on the local system.

In the second example, the `mqsilist` is run with the `-d0` option, which returns the integration node and its associated queue manager (if one is configured).

In the third example, the `mqsilist` command is run with the integration node name as the only option. It returns the status of the integration server that is running on the integration node.



WebSphere Education

Integration servers

- Named grouping of message flows are assigned to an integration node
- Run in separate address spaces or as unique processes to provide isolation and scalability
- Each integration server is a separate operating system process, which provides isolated runtime environments for a set of deployed message flows
- Within integration servers, configurable thread pools are allocated for deployed message flows
- Every integration server within an integration node must have unique name

© Copyright IBM Corporation 2015

Figure 2-28. Integration servers WM6461.0

Notes:

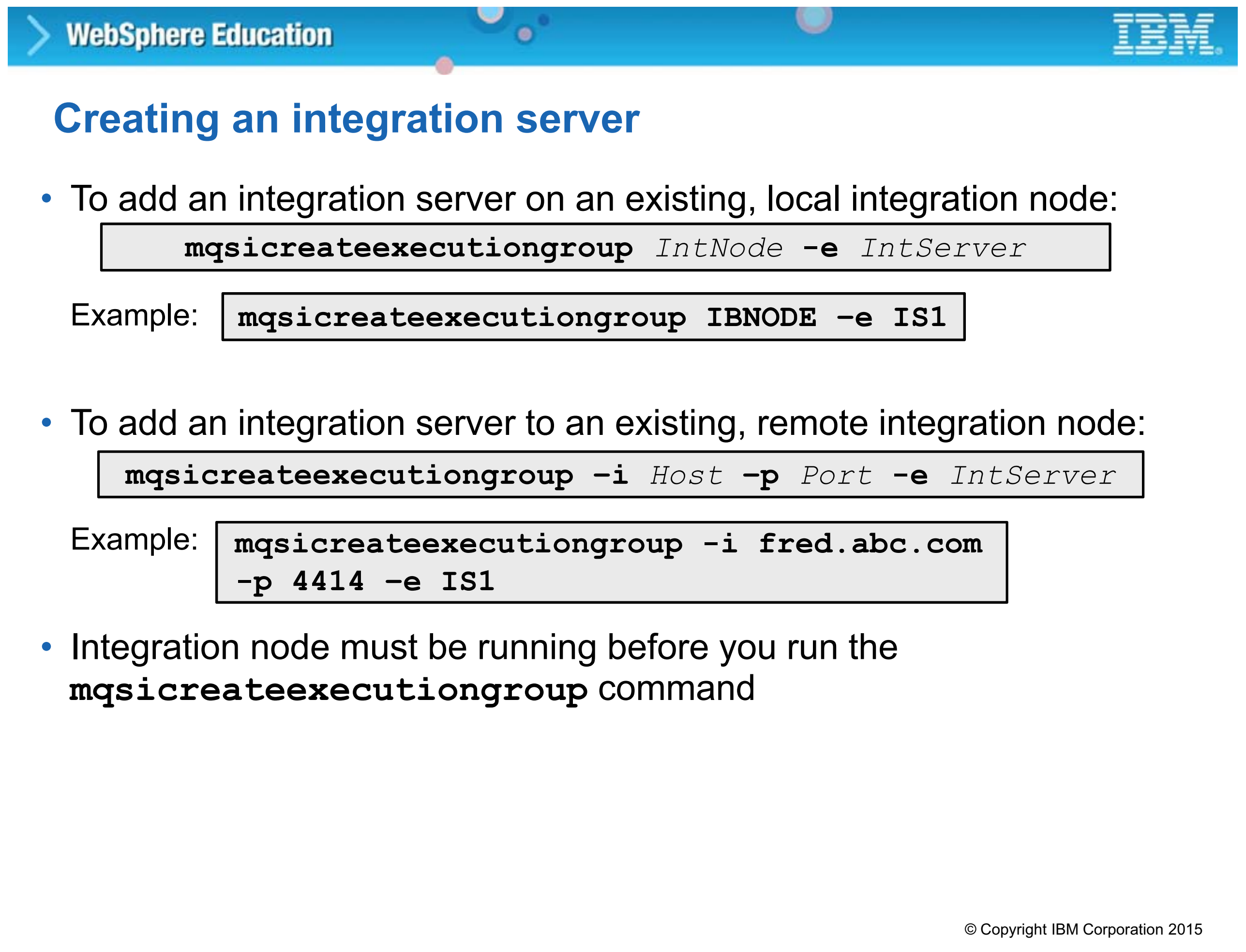
An *integration server* is a named grouping of message flows that are assigned to an integration node. Each integration server is started as a separate operating system process, providing an isolated runtime environment for a set of deployed message flows.

Within an integration server, the assigned message flows run in different thread pools. You can specify the size of the thread pool (that is, the number of threads) that are assigned for each message flow by specifying the number of instances of each message flow.

Each integration server must have a unique name on an integration node.

Some applications, such as IBM MQ, truncate the integration server name at 30 characters. If the first 30 characters of two integration servers are the same, they are considered to be the same integration server. Restrict the names of integration servers to 30 characters to avoid duplicate integration server names.

An integration server that is named `default` is created automatically on the Integration Toolkit development integration node that is named `TESTNODE_`*`username`*.



Creating an integration server

- To add an integration server on an existing, local integration node:

```
mqsicreateexecutiongroup IntNode -e IntServer
```

  Example:

```
mqsicreateexecutiongroup IBNODE -e IS1
```

- To add an integration server to an existing, remote integration node:

```
mqsicreateexecutiongroup -i Host -p Port -e IntServer
```

  Example:

```
mqsicreateexecutiongroup -i fred.abc.com
-p 4414 -e IS1
```

- Integration node must be running before you run the **`mqsicreateexecutiongroup`** command

© Copyright IBM Corporation 2015

Figure 2-29. Creating an integration server WM6461.0

Notes:

Integration servers can be created by using the `mqsicreateexecutiongroup` command, the Integration web user interface, and on Windows and Linux in the Integration Toolkit.

The first example of the `mqsicreateexecutiongroup` command creates an integration server that is named `IS1` (`-e`) on a local integration node that is named `IBNODE`.

The second example of the `mqsicreateexecutiongroup` command creates an integration server that is named `IS1` (`-e`) on the integration node that is listening on `fred.abc.com` (`-i`) with port 1414 (`-p`).

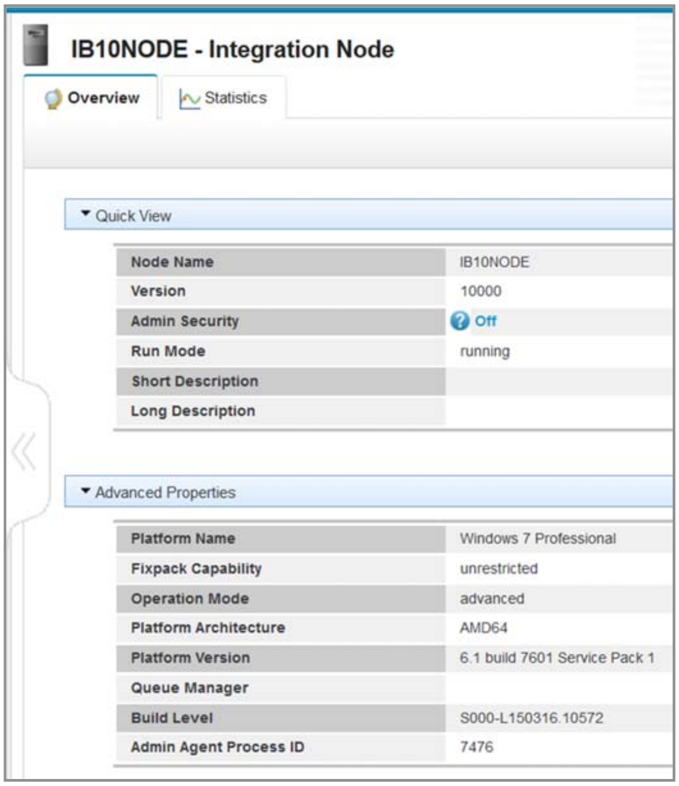
Security authorization can be implemented to restrict users who can create integration servers. Security authorization is described in a later in this course.

For more information about the `mqsicreateexecutiongroup` command and command options, see the Integration Bus product documentation.

WebSphere Education IBM

Integration node and integration server properties

- Integration Bus maintains a set of properties for each integration node and integration server
- View integration node properties by using:
  - Integration web user interface
  - Integration Toolkit
  - **`mqsireportbroker`** and **`mqsiservice`** commands
- View and modify integration server properties by using:
  - Integration web user interface
  - Integration Toolkit



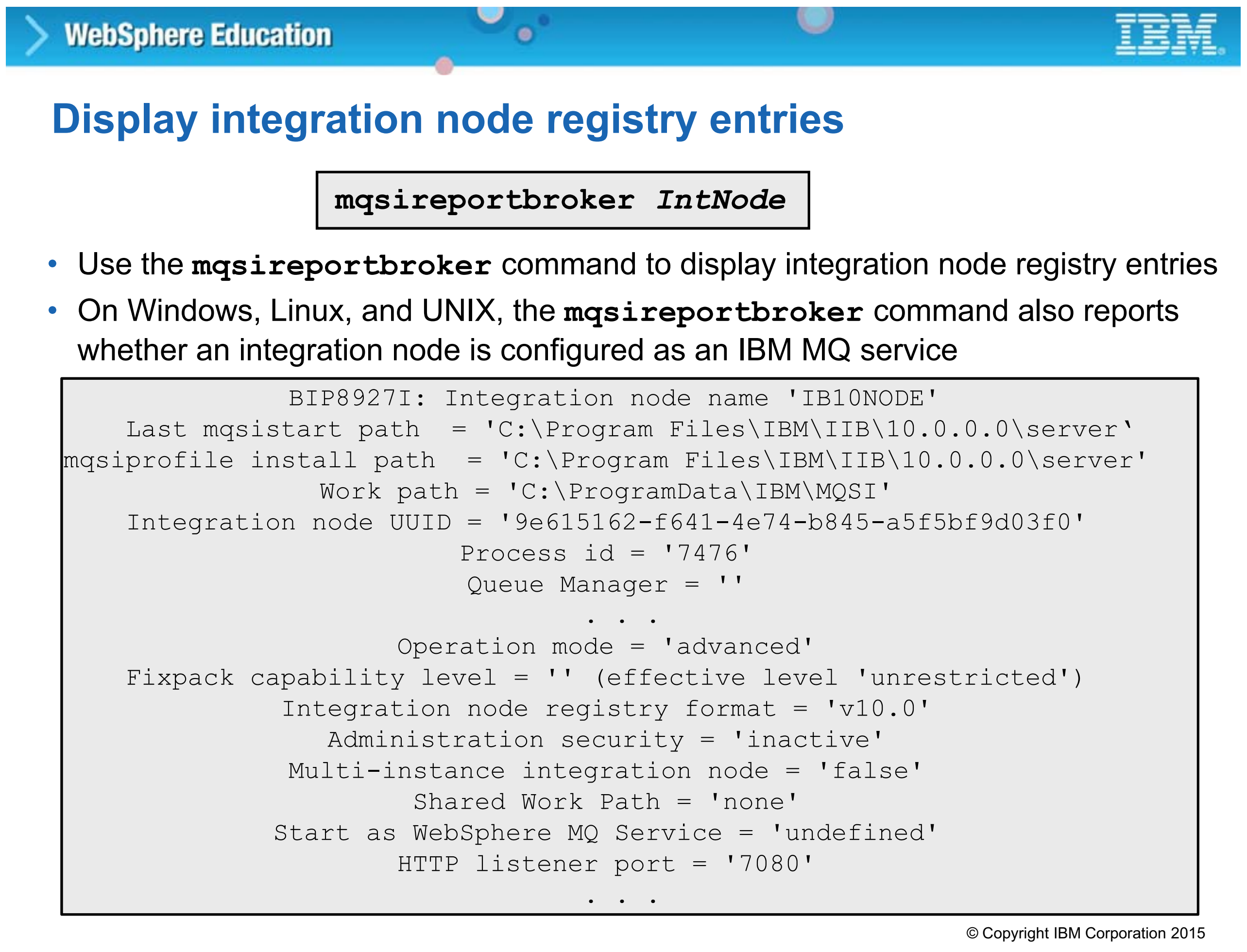
Node Name	IB10NODE
Version	10000
Admin Security	Off
Run Mode	running
Short Description	
Long Description	

Platform Name	Windows 7 Professional
Fixpack Capability	unrestricted
Operation Mode	advanced
Platform Architecture	AMD64
Platform Version	6.1 build 7601 Service Pack 1
Queue Manager	
Build Level	S000-L150316.10572
Admin Agent Process ID	7476

Figure 2-30. Integration node and integration server properties WM6461.0

Notes:

IBM Integration Bus maintains a set of properties for each integration node and integration server. You can view these properties in Integration web user interface, the Integration Toolkit, and by using the `mqsireportbroker` command.



WebSphere Education

Display integration node registry entries

```
mqsireportbroker IntNode
```

- Use the **`mqsireportbroker`** command to display integration node registry entries
- On Windows, Linux, and UNIX, the **`mqsireportbroker`** command also reports whether an integration node is configured as an IBM MQ service

```
                   BIP8927I: Integration node name 'IB10NODE'
        Last mqsistart path  = 'C:\Program Files\IBM\IIB\10.0.0.0\server'
mqsiprofile install path  = 'C:\Program Files\IBM\IIB\10.0.0.0\server'
                     Work path = 'C:\ProgramData\IBM\MQSI'
        Integration node UUID = '9e615162-f641-4e74-b845-a5f5bf9d03f0'
                              Process id = '7476'
                              Queue Manager = ''
                                     . . .
                          Operation mode = 'advanced'
        Fixpack capability level = '' (effective level 'unrestricted')
                   Integration node registry format = 'v10.0'
                      Administration security = 'inactive'
                   Multi-instance integration node = 'false'
                           Shared Work Path = 'none'
                  Start as WebSphere MQ Service = 'undefined'
                          HTTP listener port = '7080'
                                     . . .
```

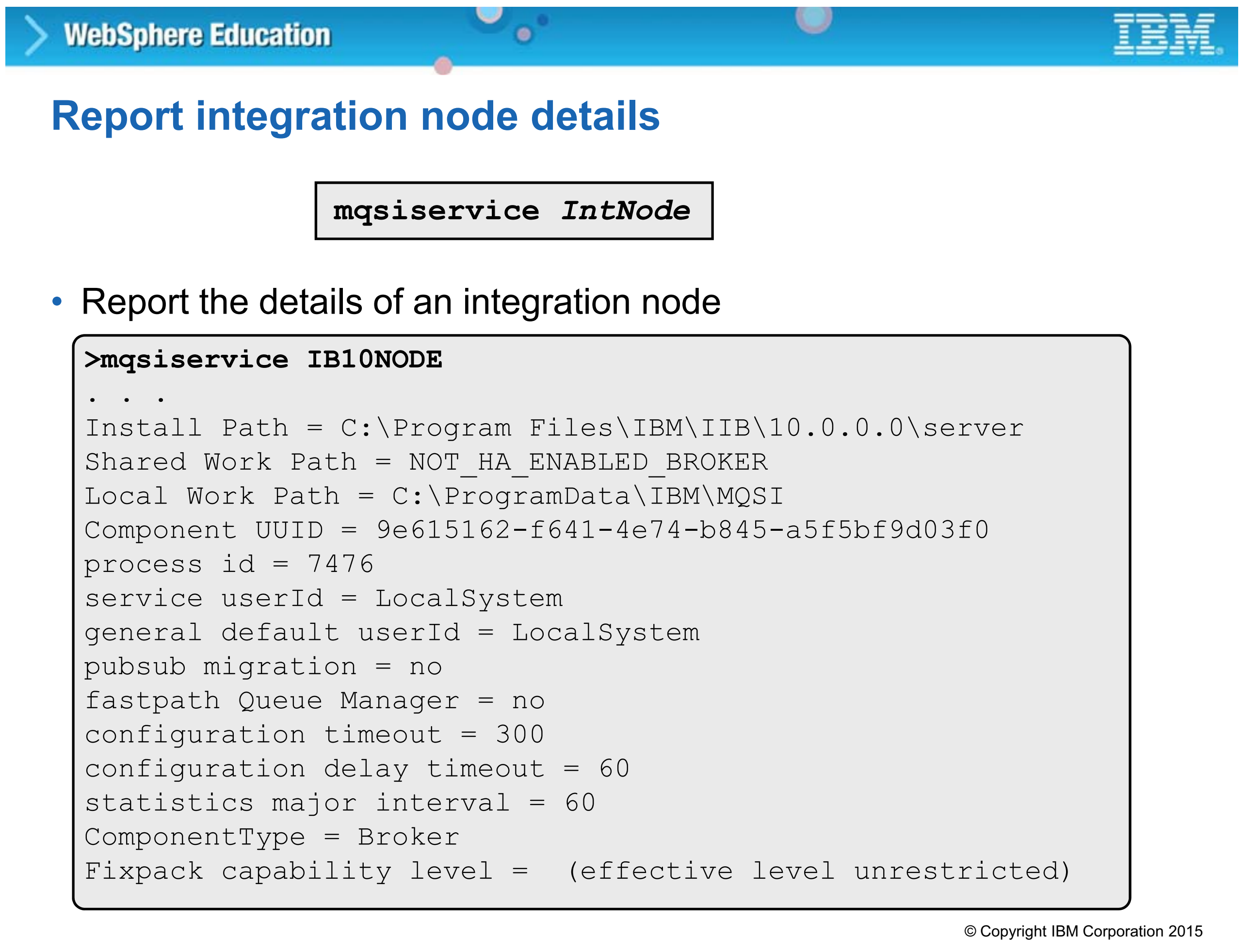
© Copyright IBM Corporation 2015

Figure 2-31. Display integration node registry entries WM6461.0

Notes:

You can use the `mqsireportbroker` command to view the property values set when you create or change the integration node.

On Windows, Linux, and UNIX, the `mqsireportbroker` command also reports whether an integration node is configured as an IBM MQ service.



Report integration node details

```
mqsiservice IntNode
```

- Report the details of an integration node

```
>mqsiservice IB10NODE
. . .
Install Path = C:\Program Files\IBM\IIB\10.0.0.0\server
Shared Work Path = NOT_HA_ENABLED_BROKER
Local Work Path = C:\ProgramData\IBM\MQSI
Component UUID = 9e615162-f641-4e74-b845-a5f5bf9d03f0
process id = 7476
service userId = LocalSystem
general default userId = LocalSystem
pubsub migration = no
fastpath Queue Manager = no
configuration timeout = 300
configuration delay timeout = 60
statistics major interval = 60
ComponentType = Broker
Fixpack capability level =  (effective level unrestricted)
```

Figure 2-32. Report integration node details WM6461.0

Notes:

The `mqsiservice` command can be run in an Integration Bus console to report on and optionally, change the properties of a component.

The figure includes an example of the `mqsiservice` command and an excerpt of the results of the `mqsiservice` command when the component is an integration node.

The `mqsiservice` command includes the following options:

`-v` is the product version information.

`componentName` is the name of the component to query for its registry data.

`-r` *`label=value`* sets the registry key 'label' to 'value'. Use with caution

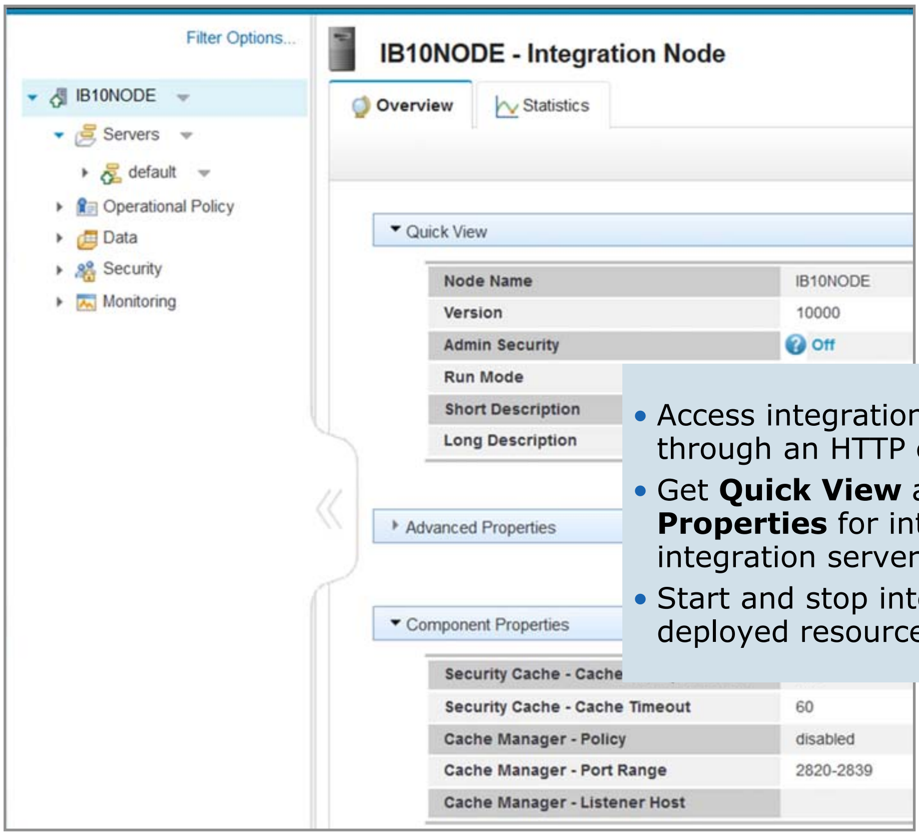
`-m` *`messageNumber`* is a message number to output.

`-c` *`messageCatalog`* is the name of the message catalog to use.

`-t` outputs information about current time and time zone.

WebSphere Education IBM

View component properties with IBM Integration web user interface



Node Name	IB10NODE
Version	10000
Admin Security	Off
Run Mode	
Short Description	
Long Description	

Security Cache - Cache	
Security Cache - Cache Timeout	60
Cache Manager - Policy	disabled
Cache Manager - Port Range	2820-2839
Cache Manager - Listener Host	

- Access integration node resources through an HTTP client
- Get **Quick View** and **Advanced Properties** for integration node, integration server, and resources
- Start and stop integration servers and deployed resources

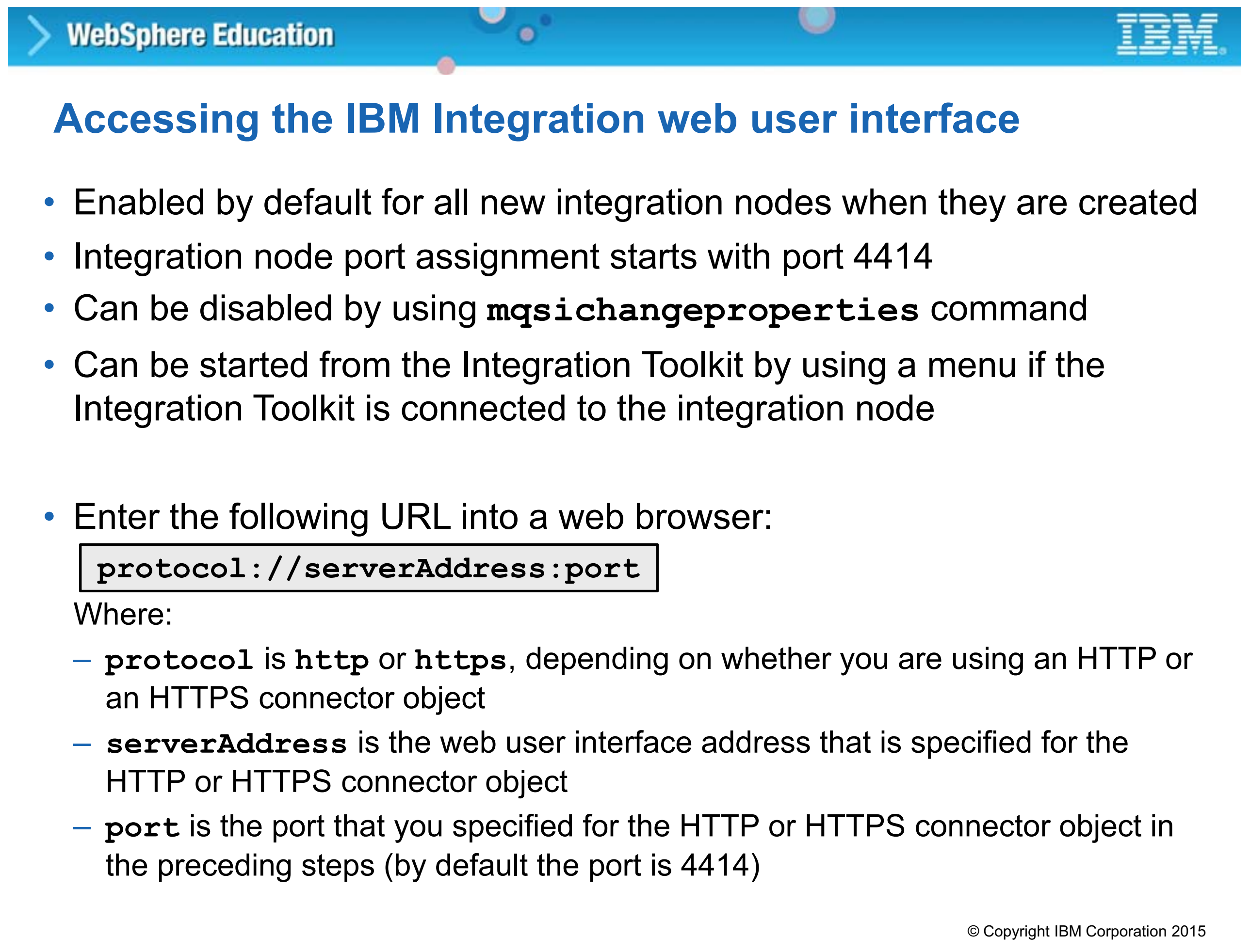
Figure 2-33. View component properties with IBM Integration web user interface WM6461.0

Notes:

With the IBM Integration web user interface, users and administrators can access integration node resources through an HTTP client. It provides administrators with a graphical user interface for managing Integration Bus components and resources.

In the Integration web user interface, you can view the properties for an integration node and the integration servers and resources on the integration node. You can also do some basic administration such as creating, starting, and stopping integration servers and starting and stopping deployed message flows. You can also complete advanced administration tasks such as creating and assigning workload management policies and defining configurable services to connect to external applications.

Support is provided for most major browsers. For more information about supported browsers, see the IBM Support site.



WebSphere Education

Accessing the IBM Integration web user interface

- Enabled by default for all new integration nodes when they are created
- Integration node port assignment starts with port 4414
- Can be disabled by using **`mqsichangeproperties`** command
- Can be started from the Integration Toolkit by using a menu if the Integration Toolkit is connected to the integration node
- Enter the following URL into a web browser:

```
protocol://serverAddress:port
```

  Where:
  - **`protocol`** is **`http`** or **`https`**, depending on whether you are using an HTTP or an HTTPS connector object
  - **`serverAddress`** is the web user interface address that is specified for the HTTP or HTTPS connector object
  - **`port`** is the port that you specified for the HTTP or HTTPS connector object in the preceding steps (by default the port is 4414)

Figure 2-34. Accessing the IBM Integration web user interface WM6461.0

Notes:

With the Integration Bus web user interface, you can access integration node resources by using a web browser. It provides integration administrators with an alternative to using commands to administer integration node resources.

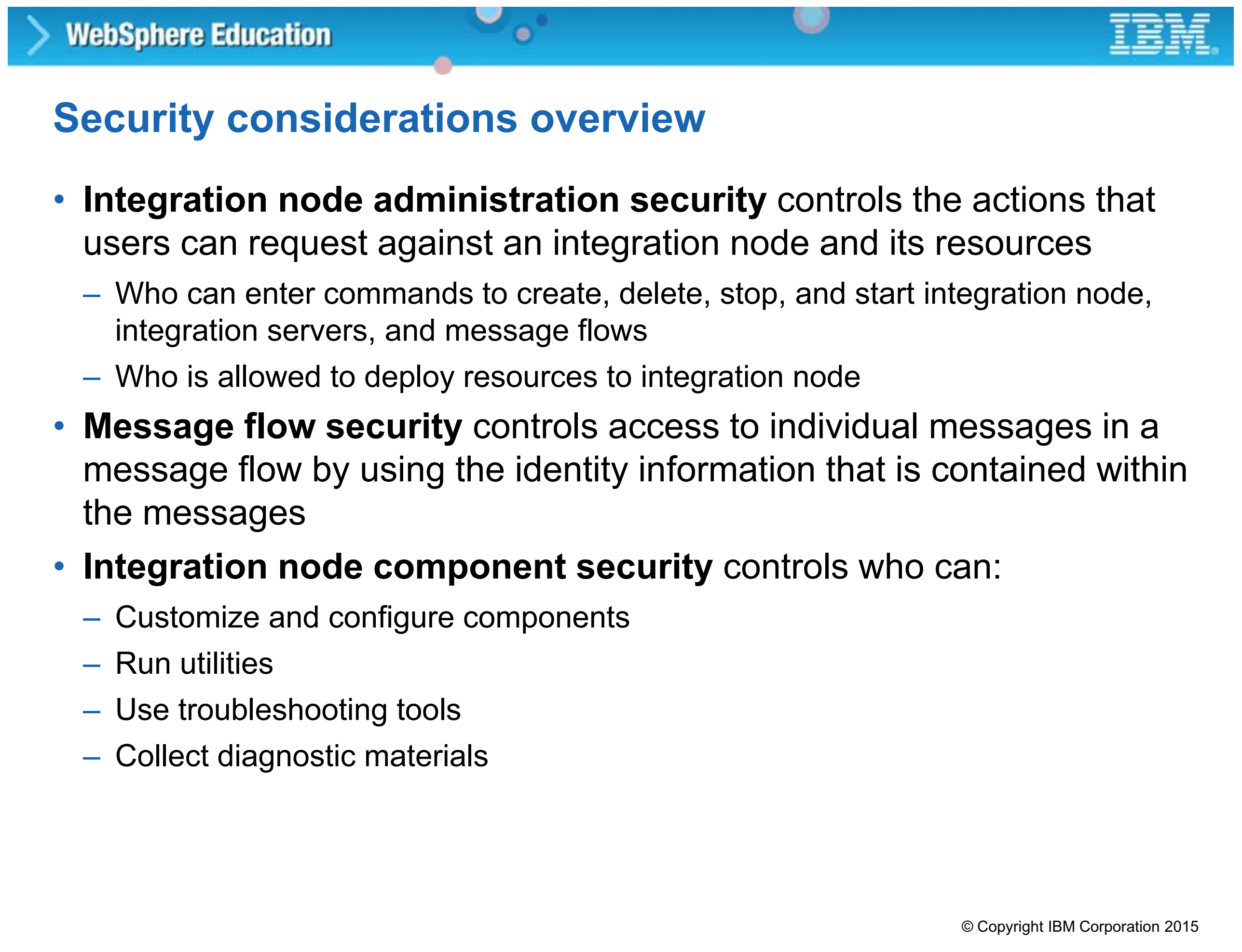
You can enable or disable the web user interface, configure the port that you use to connect to the web user interface, enable compression for the web user interface, and install user-defined web user interface content.

You can start the Integration web user interface by entering the following URL in a web browser: `protocol://serverAddress:port`. For example, to access the web user interface for a local integration node that runs on port 4414, type: `https://localhost:4414`

The Integration web user interface can also be started by clicking the integration node in the Integration Toolkit **Integration nodes** view.

2.3. Security considerations

This topic describes some IBM Integration Bus security considerations.



WebSphere Education IBM

Security considerations overview

- **Integration node administration security** controls the actions that users can request against an integration node and its resources
  - Who can enter commands to create, delete, stop, and start integration node, integration servers, and message flows
  - Who is allowed to deploy resources to integration node
- **Message flow security** controls access to individual messages in a message flow by using the identity information that is contained within the messages
- **Integration node component security** controls who can:
  - Customize and configure components
  - Run utilities
  - Use troubleshooting tools
  - Collect diagnostic materials

© Copyright IBM Corporation 2015

Figure 2-35. Security considerations overview WM6461.0

Notes:

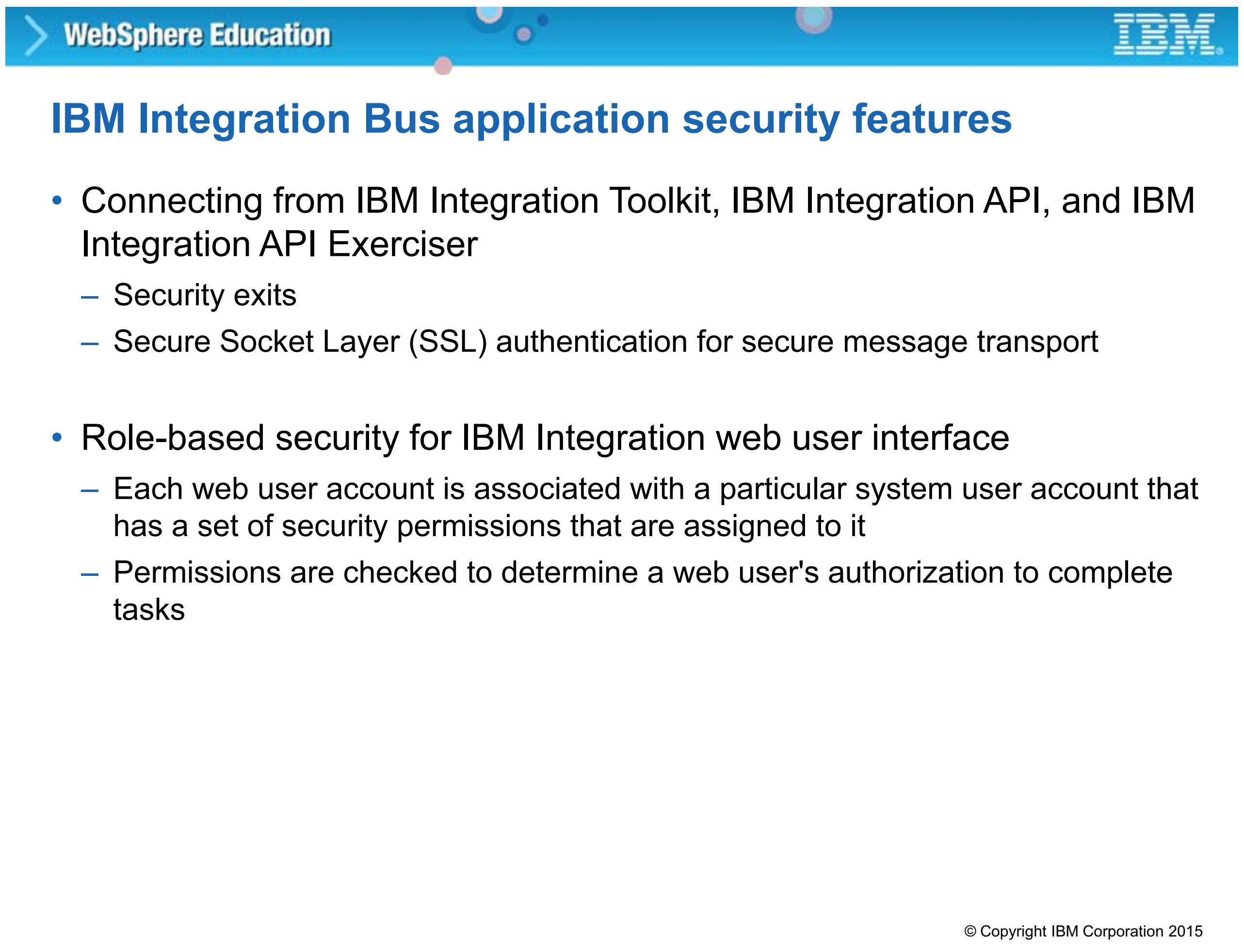
Integration Bus supports several types of security.

There is security for the integration node objects, which controls who can access the integration node and other integration node resources.

There is security as it relates to the individual message flows, which allows the ID associated with a message that starts a message flow to be authenticated, authorized, and propagated through the message flow. If you do not enable message flow security, the default Integration Bus facilities are based on transport-specific facilities. Integration Bus processes all messages that can be physically delivered to it, using the integration node service ID as a proxy for all incoming messages. In this case, even if a specific user ID is present in an incoming message, it is ignored.

Other kinds of security considerations deal with web services security, authorization for installation and configuration tasks, securing development resources, security exits, and other areas.

Security implementation and configuration are described later in this course.



WebSphere Education IBM

IBM Integration Bus application security features

- Connecting from IBM Integration Toolkit, IBM Integration API, and IBM Integration API Exerciser
  - Security exits
  - Secure Socket Layer (SSL) authentication for secure message transport
- Role-based security for IBM Integration web user interface
  - Each web user account is associated with a particular system user account that has a set of security permissions that are assigned to it
  - Permissions are checked to determine a web user's authorization to complete tasks

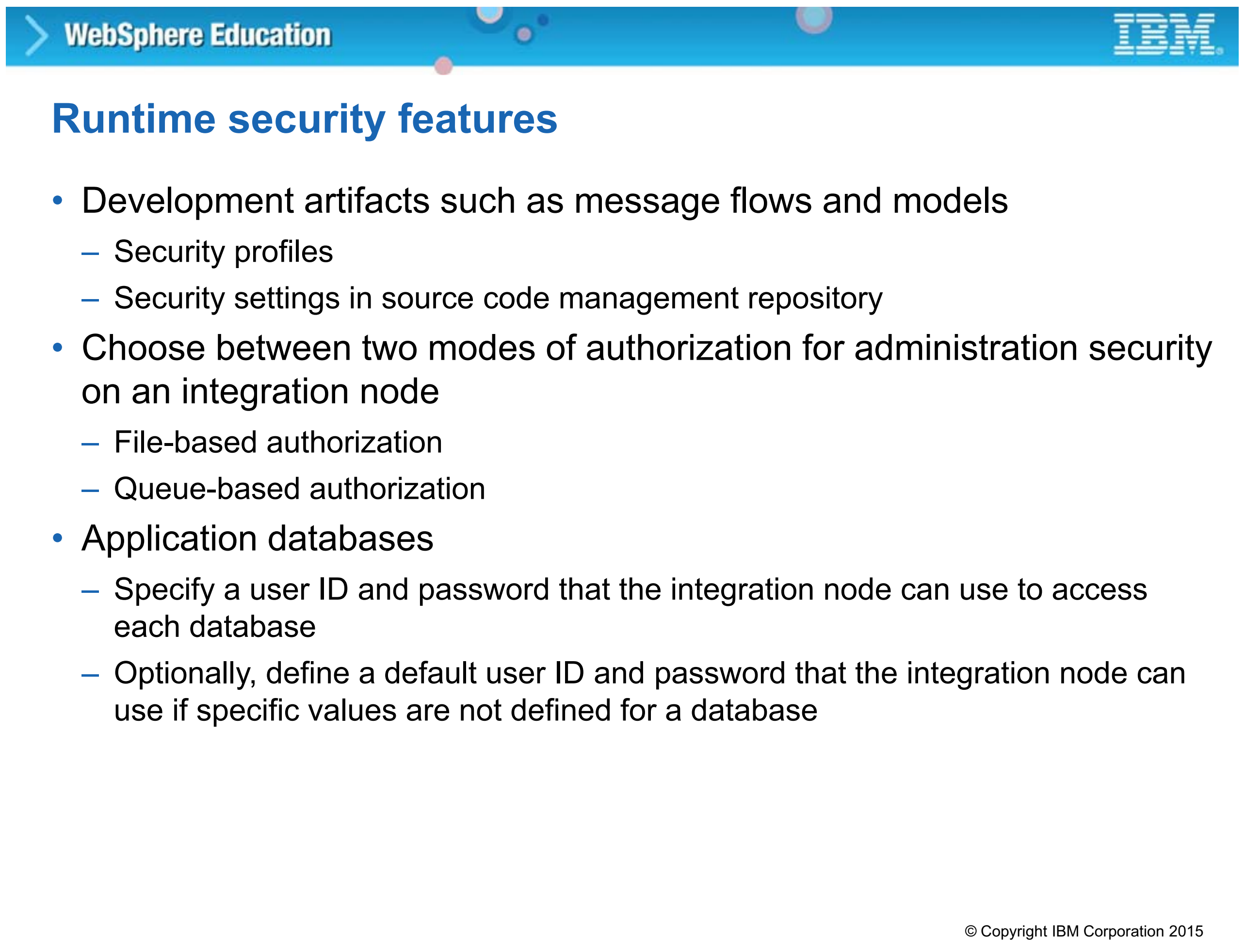
Figure 2-36. IBM Integration Bus application security features WM6461.0

Notes:

Security is an important consideration for both developers of Integration Bus applications, and for system administrators that configure Integration Bus authorities.

Integration Bus includes standard support for administration security, message flow security, and IBM Integration Bus component security. In addition, you can use security exit programs to verify that the other end of a connection is genuine.

The Integration web user interface supports role-based security. The administrator can grant permissions to the roles that are associated with web users. The Integration Bus administrators can allow those users to start and stop integration servers, applications, and message flows from the web user interface.



WebSphere Education

Runtime security features

- Development artifacts such as message flows and models
  - Security profiles
  - Security settings in source code management repository
- Choose between two modes of authorization for administration security on an integration node
  - File-based authorization
  - Queue-based authorization
- Application databases
  - Specify a user ID and password that the integration node can use to access each database
  - Optionally, define a default user ID and password that the integration node can use if specific values are not defined for a database

© Copyright IBM Corporation 2015

Figure 2-37. Runtime security features WM6461.0

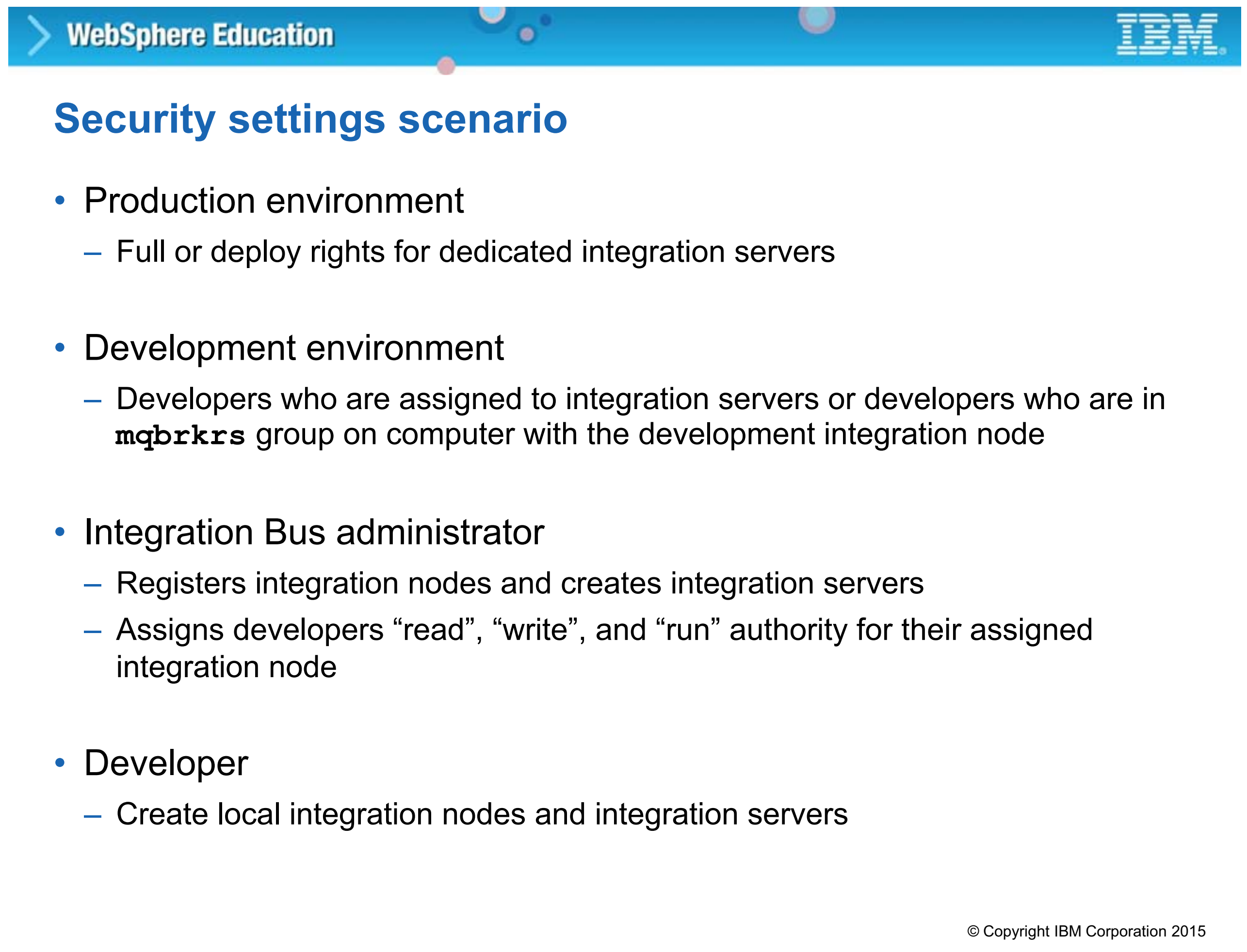
Notes:

For security of the runtime configuration, you can choose between or mix two models:

- Authorize each object separately
- Use group memberships that reflect the user’s role, such as developer or operator

In Integration Bus V10, you can choose between file-based authorization or, if the integration node is associated with an IBM MQ queue manager, queue-based authorization.

Databases can be accessed with the service ID of the component, or with a specific database ID and password. The default message flow behavior is to access all user databases and all user queues with the integration node service ID.



WebSphere Education

Security settings scenario

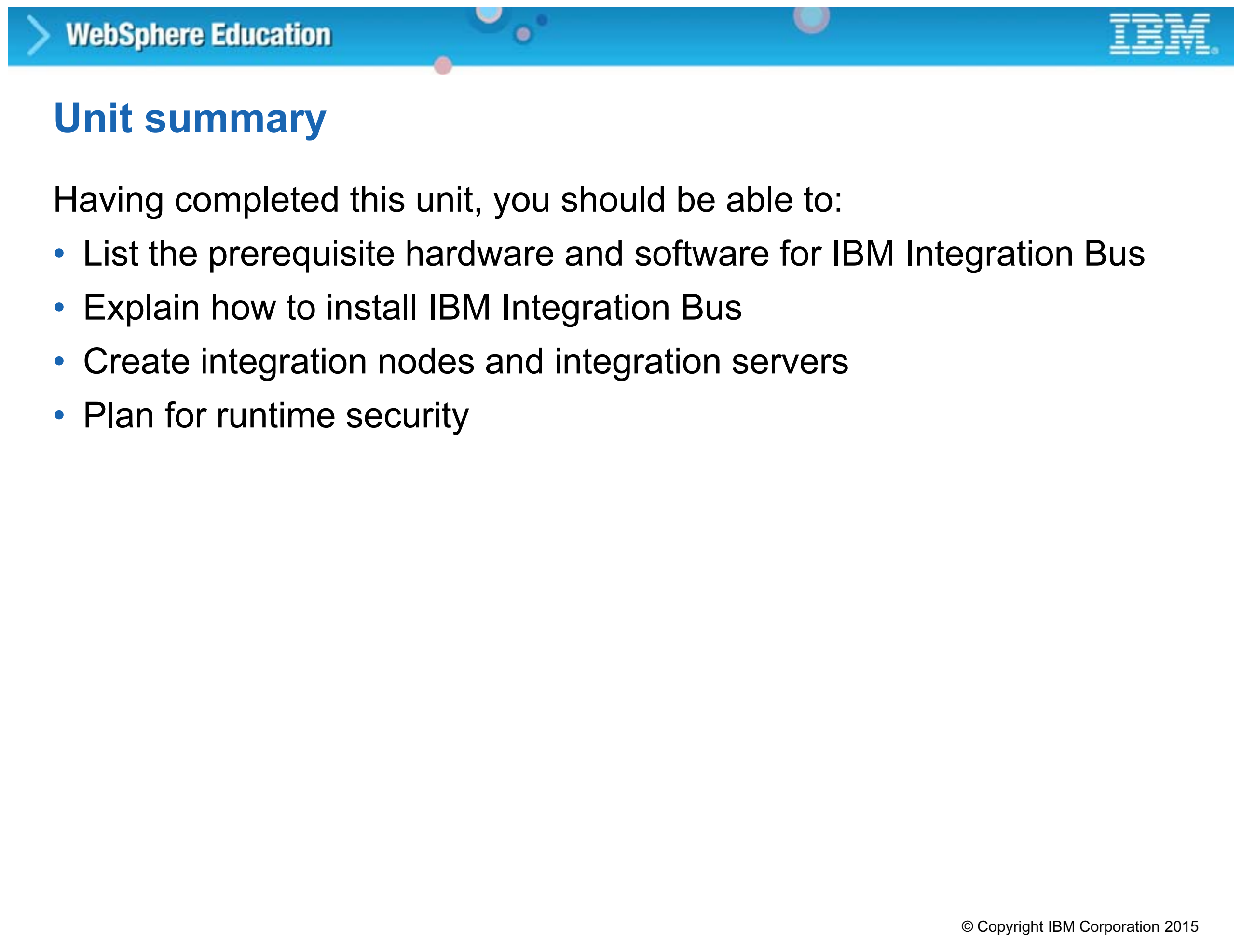
- Production environment
  - Full or deploy rights for dedicated integration servers
- Development environment
  - Developers who are assigned to integration servers or developers who are in **`mqbrkrs`** group on computer with the development integration node
- Integration Bus administrator
  - Registers integration nodes and creates integration servers
  - Assigns developers “read”, “write”, and “run” authority for their assigned integration node
- Developer
  - Create local integration nodes and integration servers

Figure 2-38. Security settings scenario WM6461.0

Notes:

The figure identifies the groups of users and some suggestions as to how best to use the various security authorizations, depending on what the accessing user needs to access.

For example, in a development environment, you should allow developers to deploy and control development integration nodes and integration servers. However, in a production environment access to integration nodes should be limited to administrators.



WebSphere Education

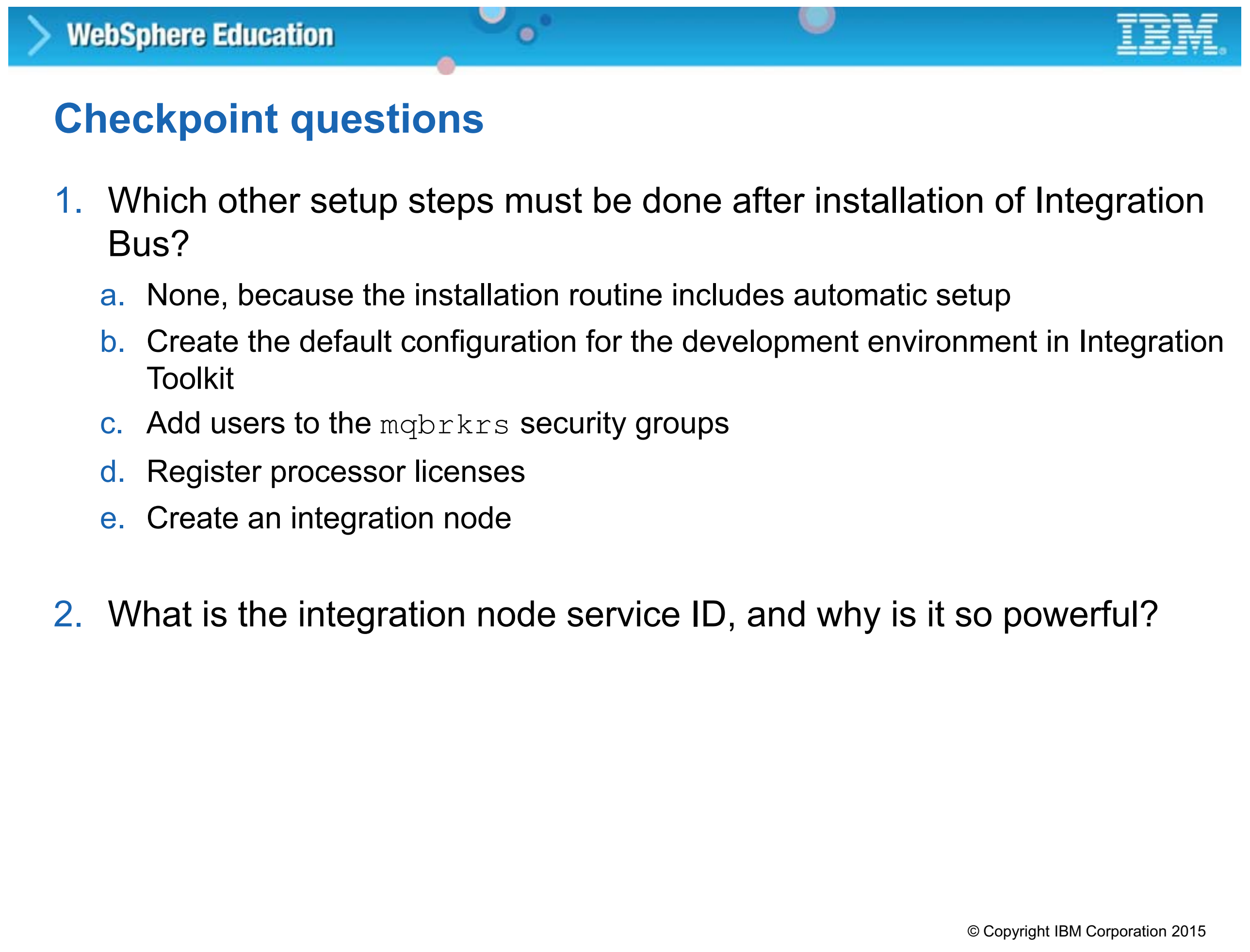
Unit summary

Having completed this unit, you should be able to:

- List the prerequisite hardware and software for IBM Integration Bus
- Explain how to install IBM Integration Bus
- Create integration nodes and integration servers
- Plan for runtime security

Figure 2-39. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. Which other setup steps must be done after installation of Integration Bus?
   a. None, because the installation routine includes automatic setup
   b. Create the default configuration for the development environment in Integration Toolkit
   c. Add users to the `mqbrkrs` security groups
   d. Register processor licenses
   e. Create an integration node

2. What is the integration node service ID, and why is it so powerful?

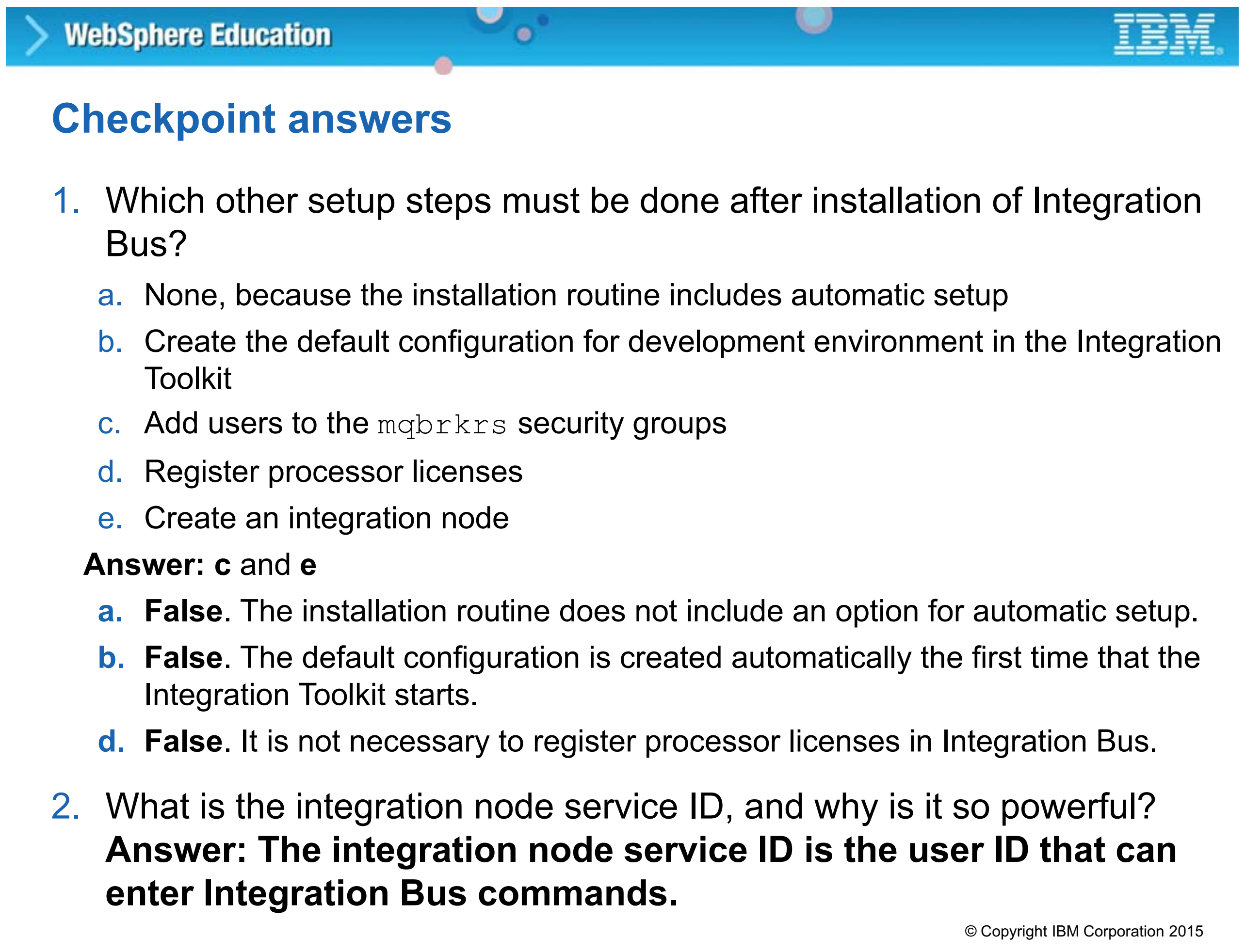
Figure 2-40. Checkpoint questions WM6461.0

Notes:

Write your answer here:

1.

2.



WebSphere Education

Checkpoint answers

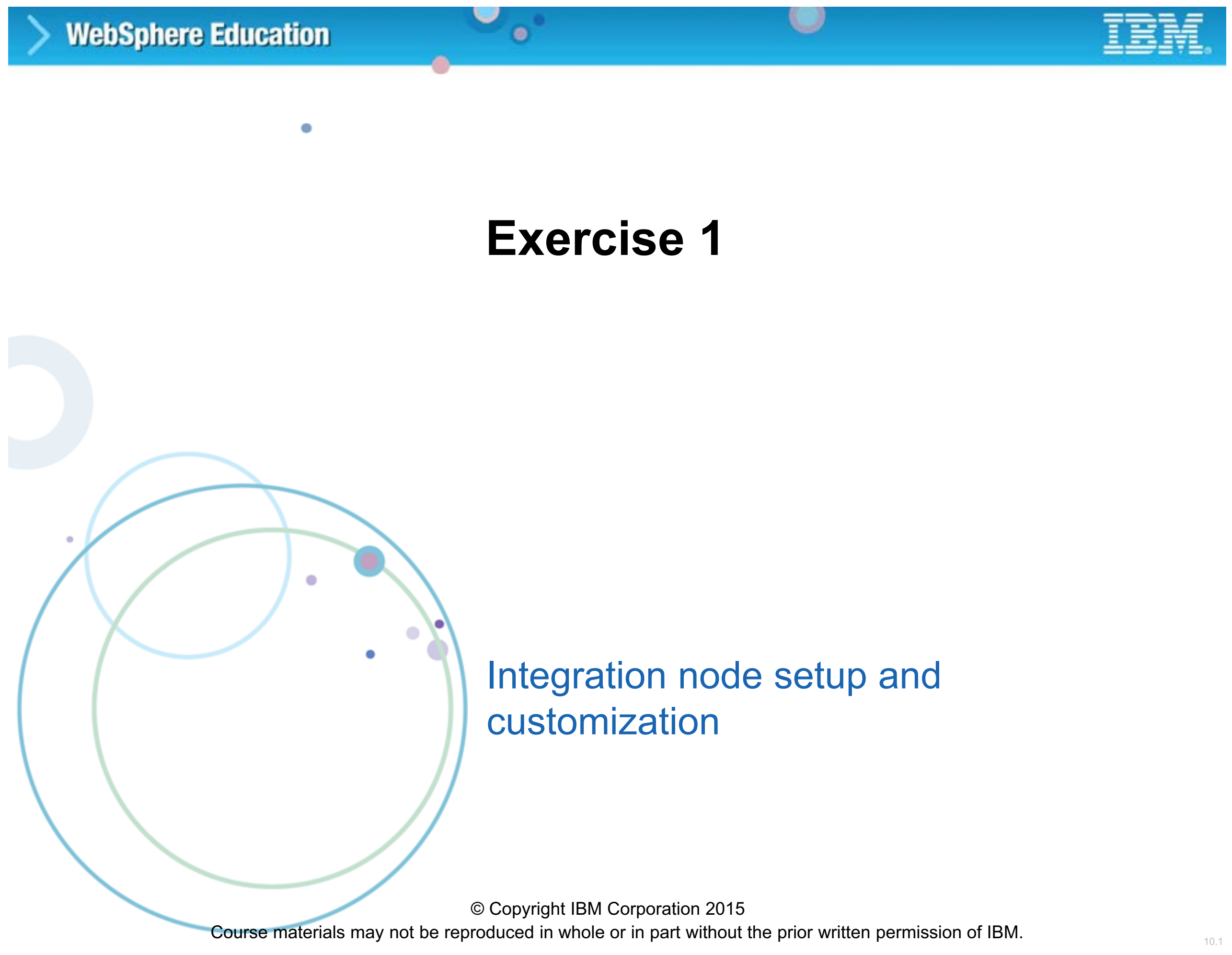
1. Which other setup steps must be done after installation of Integration Bus?
   a. None, because the installation routine includes automatic setup
   b. Create the default configuration for development environment in the Integration Toolkit
   c. Add users to the `mqbrkrs` security groups
   d. Register processor licenses
   e. Create an integration node

   Answer: c and **e**
   a. False. The installation routine does not include an option for automatic setup.
   b. False. The default configuration is created automatically the first time that the Integration Toolkit starts.
   d. False. It is not necessary to register processor licenses in Integration Bus.

2. What is the integration node service ID, and why is it so powerful?
   Answer: The integration node service ID is the user ID that can enter Integration Bus commands.

Figure 2-41. Checkpoint answers WM6461.0

Notes:



WebSphere Education

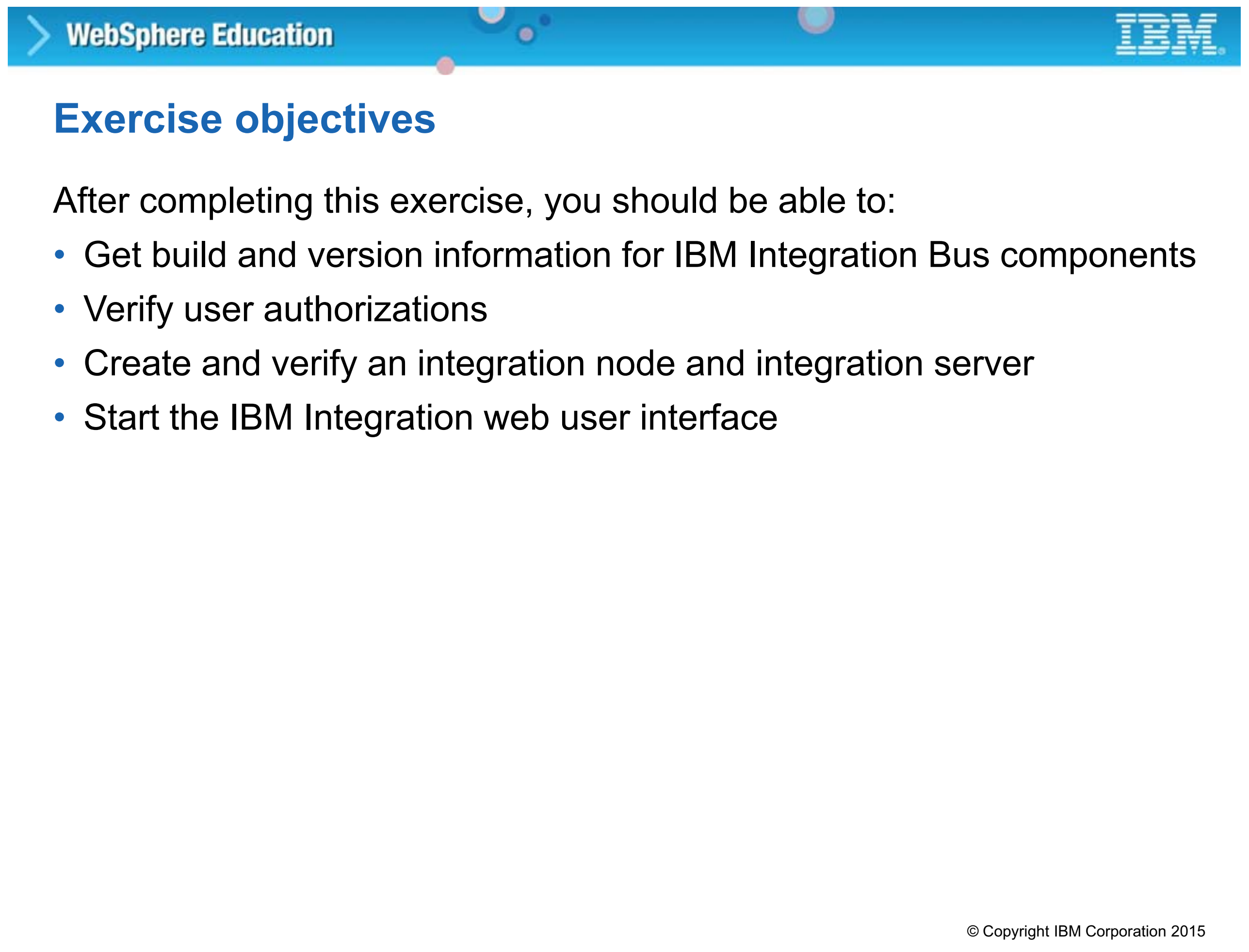
Exercise 1

Integration node setup and customization

Figure 2-42. Exercise 1 WM6461.0

Notes:

In this exercise, you complete the configuration tasks that prepare the IBM Integration Bus environment for use. You also start the IBM Integration web user interface to view the integration node and integration server properties.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Get build and version information for IBM Integration Bus components
- Verify user authorizations
- Create and verify an integration node and integration server
- Start the IBM Integration web user interface

© Copyright IBM Corporation 2015

Figure 2-43. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 3. Connecting to IBM MQ

What this unit is about

In this unit, you learn about the connectivity and topology options for enabling IBM Integration Bus integration nodes to get messages from or put messages to queues on local or remote IBM MQ queue managers.

What you should be able to do

After completing this unit, you should be able to:

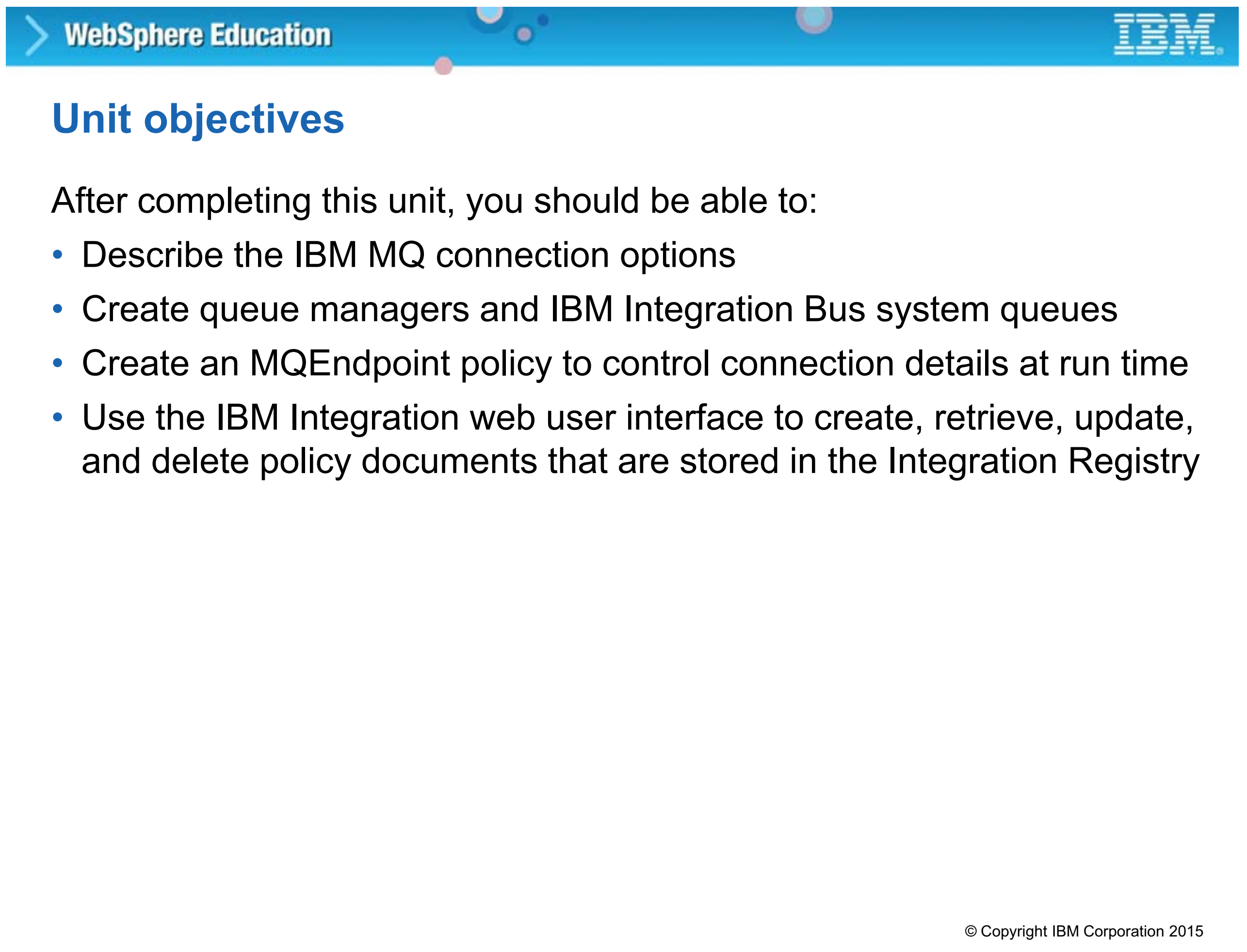
- Describe the IBM MQ connection options
- Create queue managers and IBM Integration Bus system queues
- Create an MQEndpoint policy to control connection details at run time
- Use the IBM Integration web user interface to create, retrieve, update, and delete policy documents that are stored in the Integration Registry

How you will check your progress

- Checkpoint questions
- Lab exercise

References

IBM Knowledge Center



Unit objectives

After completing this unit, you should be able to:

- Describe the IBM MQ connection options
- Create queue managers and IBM Integration Bus system queues
- Create an MQEndpoint policy to control connection details at run time
- Use the IBM Integration web user interface to create, retrieve, update, and delete policy documents that are stored in the Integration Registry

Figure 3-1. Unit objectives WM6461.0

Notes:

3.1. IBM MQ overview

WebSphere Education IBM

IBM MQ functional overview

- Data is transferred between applications in messages
- Messages waiting to be processed or routed are stored on queues

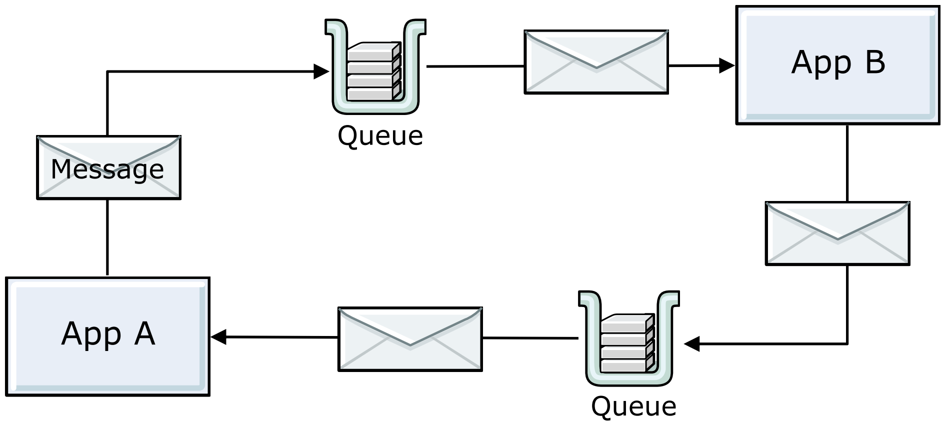


Figure 3-2. IBM MQ functional overview WM6461.0

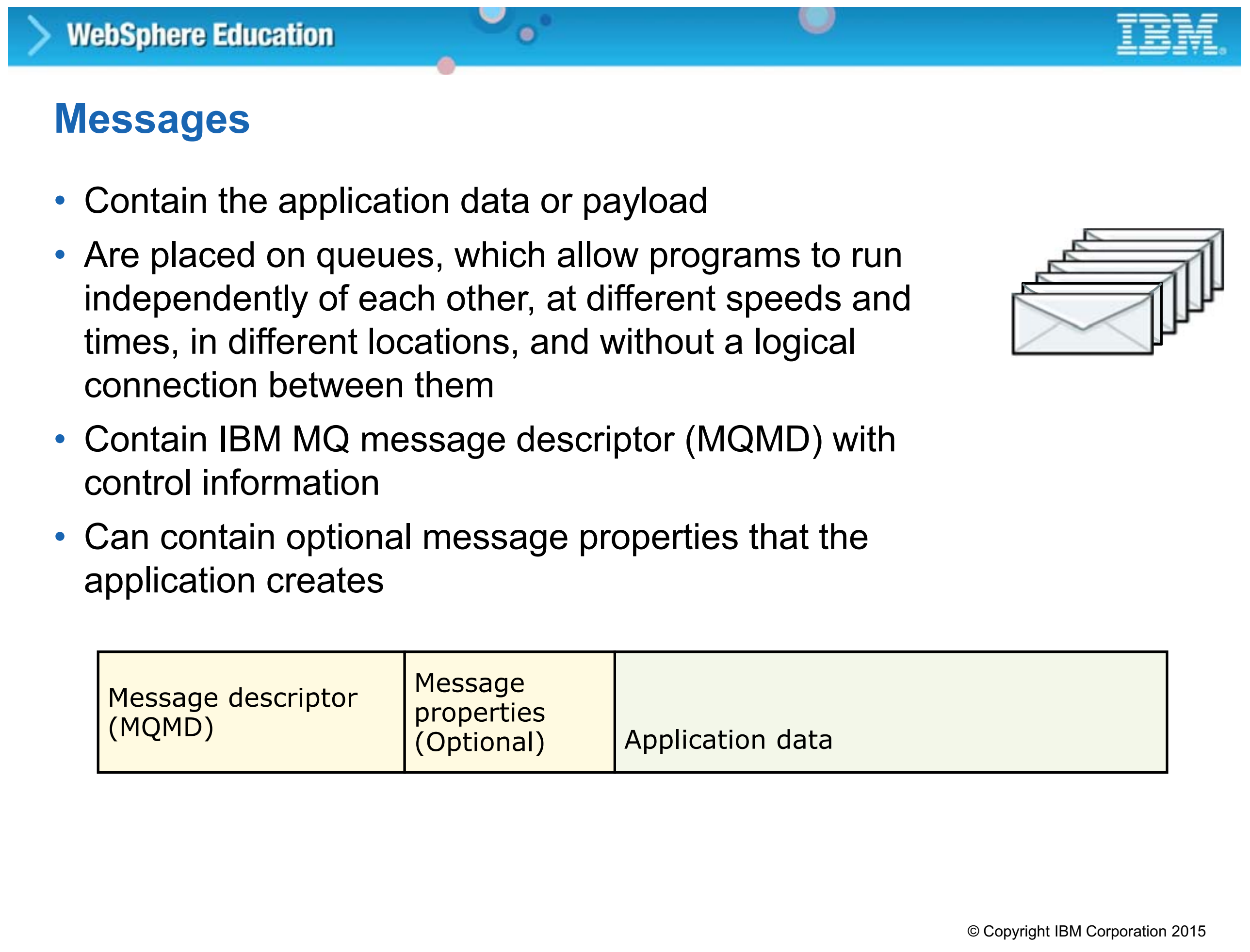
Notes:

IBM MQ uses messages and queues to provide program-to-program communication. The communicating applications can be on the same system, or distributed across a network of IBM and non-IBM systems.

A *queue* is a place to store messages until they can be processed.

The major benefits of IBM MQ are:

- There is a common application programming interface that is consistent across all the supported operating systems.
- IBM MQ can transfer data with assured delivery; messages do not get lost, even when a system failure occurs. There is no duplicate delivery.
- Communicating applications do not have to be active at the same time.
- An application is divided into discrete functional modules that communicate with each other with messages. In this way, the modules can run on different systems, be scheduled at different times, or they can act in parallel.



WebSphere Education

Messages

- Contain the application data or payload
- Are placed on queues, which allow programs to run independently of each other, at different speeds and times, in different locations, and without a logical connection between them
- Contain IBM MQ message descriptor (MQMD) with control information
- Can contain optional message properties that the application creates

Message descriptor (MQMD)	Message properties (Optional)	Application data

© Copyright IBM Corporation 2015

Figure 3-3. Messages WM6461.0

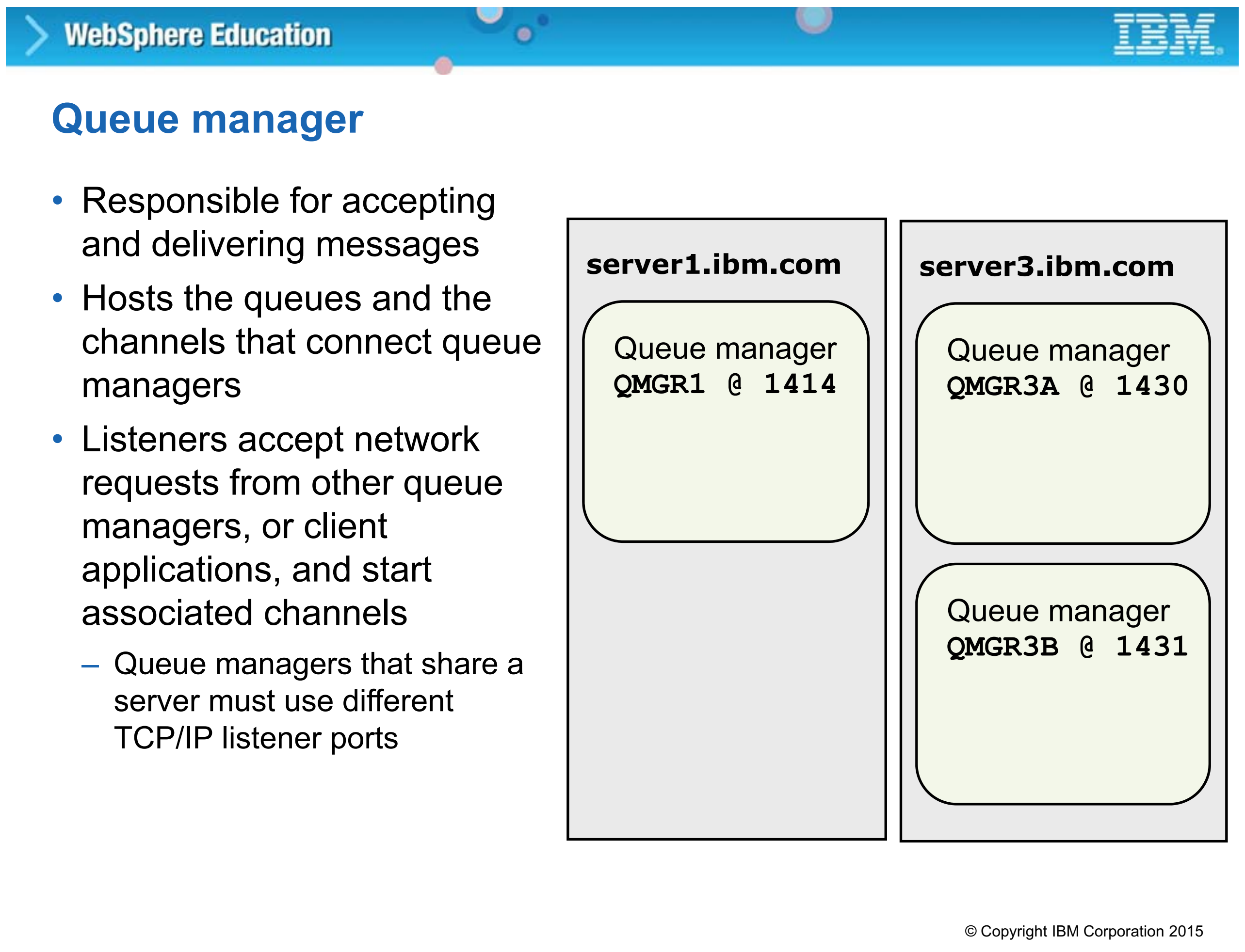
Notes:

A *message* is any information that one application needs to communicate to another application. A message can convey a request for a service, or it can be a reply to a request. It might also report on the progress of another message; to confirm its arrival or report on an error, for example. A message can also carry information for which no reply is expected.

A message consists of two parts:

- Message descriptor
- Application data

The message descriptor contains information about the message. The sending application supplies both the message descriptor and the application data when it puts a message on a queue. Both the message descriptor and the application data are returned to the application that gets the message from the queue. The application that puts the messages on the queue sets some of the fields in the message descriptor; the queue manager sets others on behalf of the application.



WebSphere Education

Queue manager

- Responsible for accepting and delivering messages
- Hosts the queues and the channels that connect queue managers
- Listeners accept network requests from other queue managers, or client applications, and start associated channels
  - Queue managers that share a server must use different TCP/IP listener ports

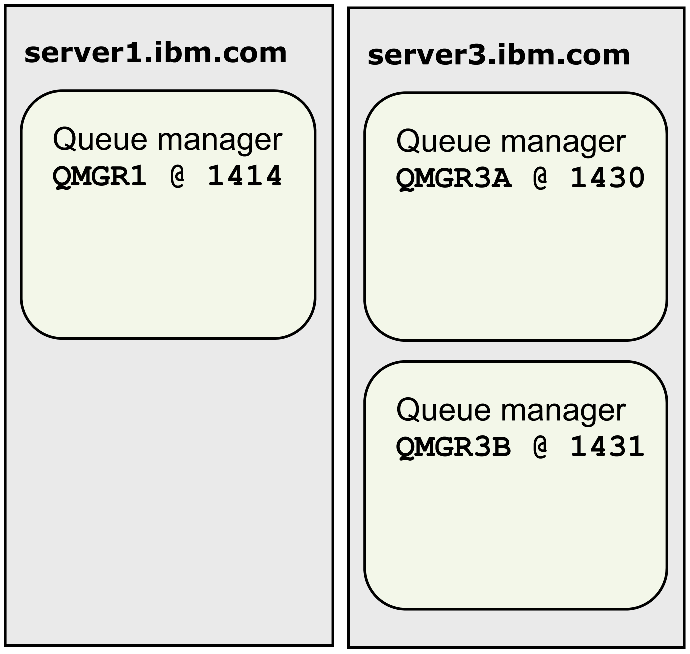

server1.ibm.com
Queue manager QMGR1 @ 1414
server3.ibm.com
Queue manager QMGR3A @ 1430
Queue manager QMGR3B @ 1431

Figure 3-4. Queue manager WM6461.0

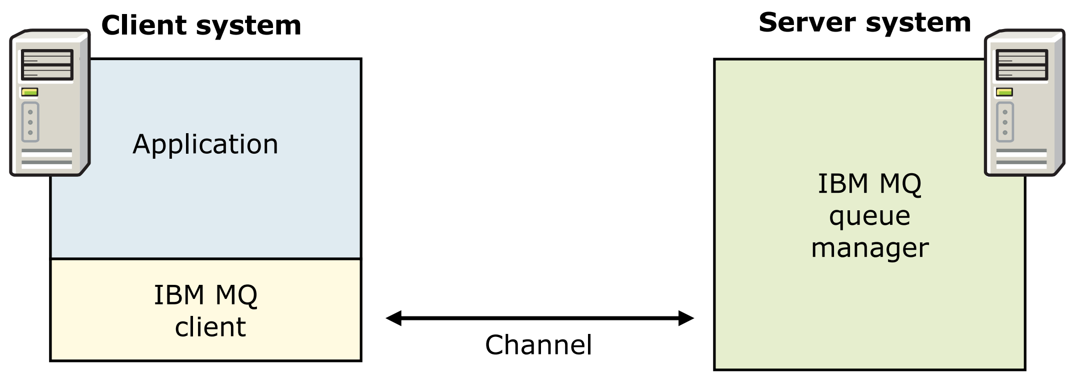
Notes:

Queue managers are the main components in an IBM MQ messaging network. The queue managers host the other objects in the network, such as the queues and the channels that connect the queue managers together.

In a distributed environment, each queue manager is assigned a unique TCP/IP port.

WebSphere Education

IBM MQ client overview



- IBM MQ component on a system without a queue manager
  - Allows an application that runs on the client system to connect to a queue manager that is running on another system
  - Communicates with the queue manager by using a *channel*
  - Client channel definition table (CCDT) determines the channel definitions and authentication information that is used by client applications to connect to the queue manager

Figure 3-5. IBM MQ client overview WM6461.0

Notes:

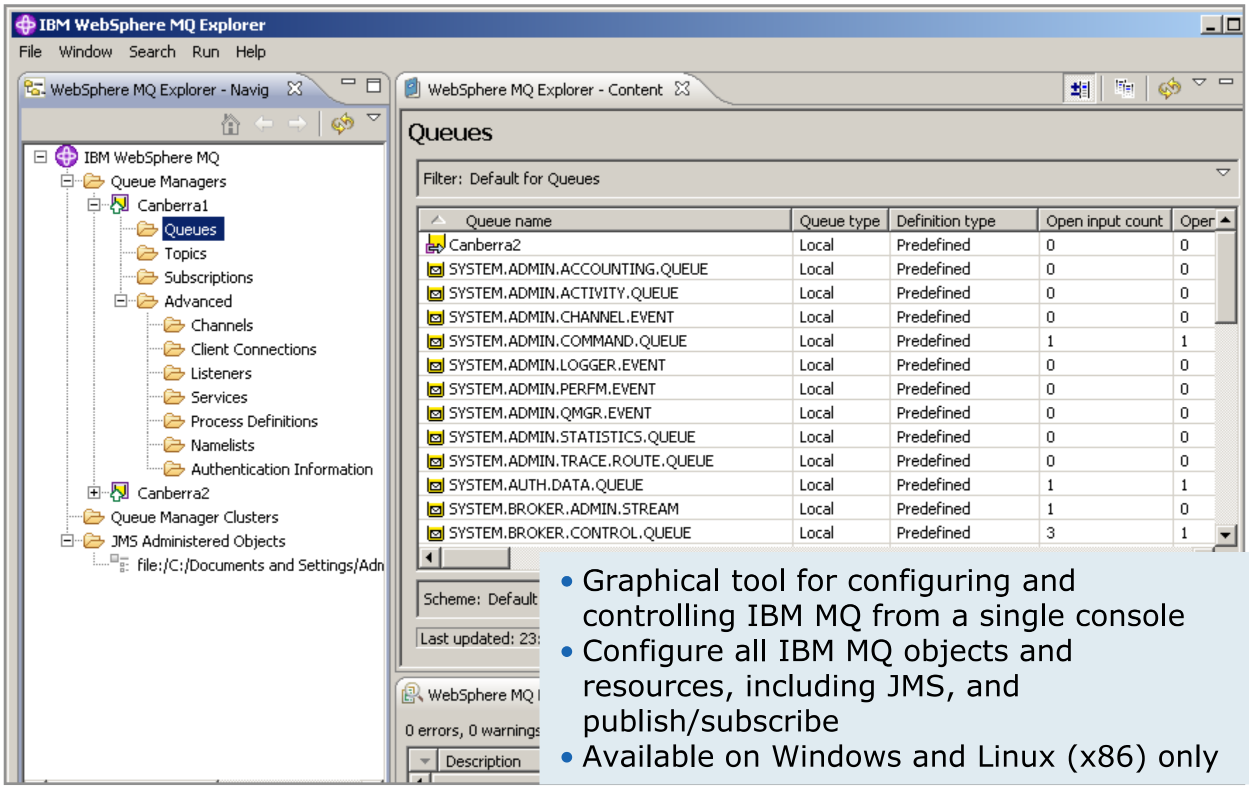
An IBM MQ client is a component of IBM MQ that allows an application that is running on one system to send MQI calls to a queue manager that is running on another system.

The client connection receives the input parameters of an MQI call from the application and sends them over a communications connection to the server connection on the same system as the queue manager. The server connection then sends the MQI call to the queue manager on behalf of the application. After the queue manager completes the MQI call, the server connection sends the output parameters of the callback to the client connection, which then passes them onto the application.

The combination of a client connection, a server connection, and a communications connection between them (for example, an SNA LU6.2 conversation or a TCP connection) is called an MQI channel.

WebSphere Education IBM

IBM MQ Explorer



Queue name	Queue type	Definition type	Open input count	Oper
Canberra2	Local	Predefined	0	0
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Local	Predefined	0	0
SYSTEM.ADMIN.ACTIVITY.QUEUE	Local	Predefined	0	0
SYSTEM.ADMIN.CHANNEL.EVENT	Local	Predefined	0	0
SYSTEM.ADMIN.COMMAND.QUEUE	Local	Predefined	1	1
SYSTEM.ADMIN.LOGGER.EVENT	Local	Predefined	0	0
SYSTEM.ADMIN.PERFM.EVENT	Local	Predefined	0	0
SYSTEM.ADMIN.QMGR.EVENT	Local	Predefined	0	0
SYSTEM.ADMIN.STATISTICS.QUEUE	Local	Predefined	0	0
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Local	Predefined	0	0
SYSTEM.AUTH.DATA.QUEUE	Local	Predefined	1	1
SYSTEM.BROKER.ADMIN.STREAM	Local	Predefined	1	0
SYSTEM.BROKER.CONTROL.QUEUE	Local	Predefined	3	1

- Graphical tool for configuring and controlling IBM MQ from a single console
- Configure all IBM MQ objects and resources, including JMS, and publish/subscribe
- Available on Windows and Linux (x86) only

Figure 3-6. IBM MQ Explorer WM6461.0

Notes:

IBM MQ Explorer is the graphical user interface for administering and monitoring IBM MQ objects, whether your local computer or a remote system hosts them.

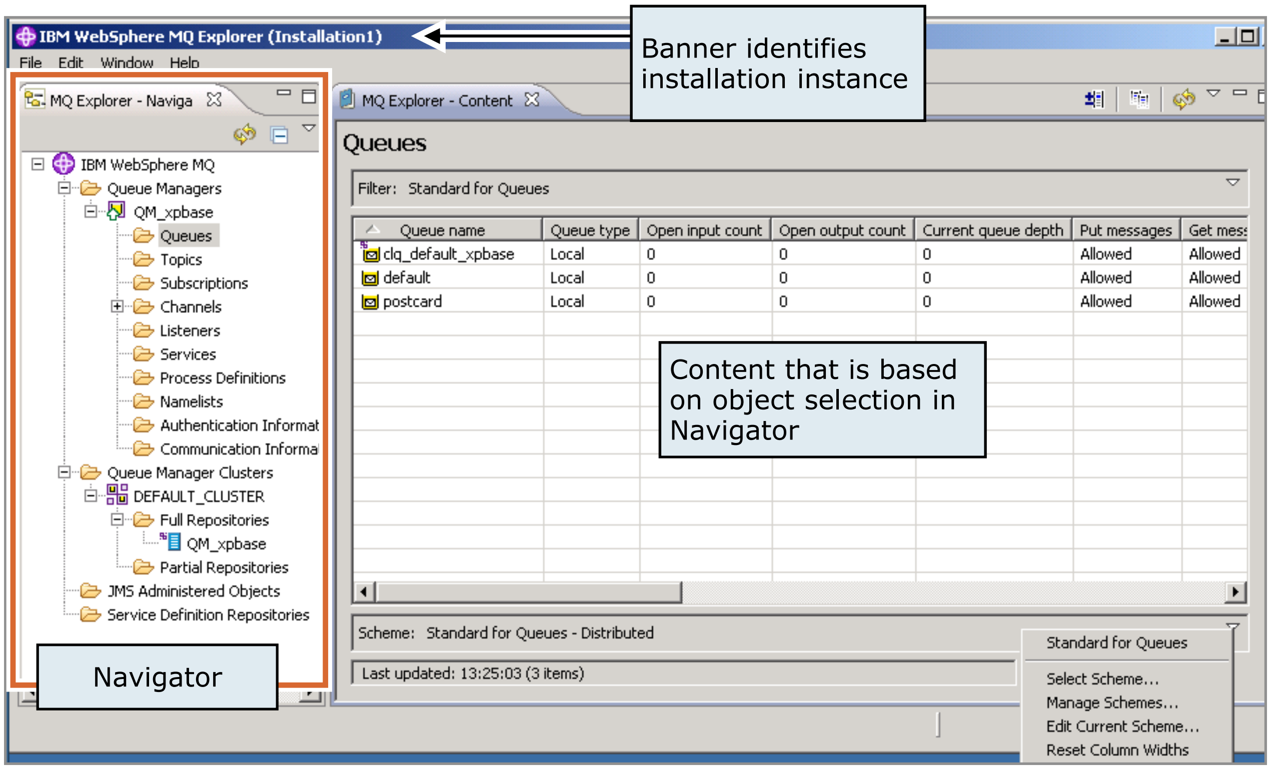
IBM MQ Explorer also supports:

- Advanced filtering
- Management of application connections
- Grouping of queue managers to simplify management
- Publish/subscribe administration

IBM MQ Explorer runs on Windows and Linux operating systems but can be used to remotely manage IBM MQ objects on other operating systems.

WebSphere Education

IBM MQ Explorer default views



Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output count	Current queue depth	Put messages	Get mess
clq_default_xpbase	Local	0	0	0	Allowed	Allowed
default	Local	0	0	0	Allowed	Allowed
postcard	Local	0	0	0	Allowed	Allowed

Scheme: Standard for Queues - Distributed

Last updated: 13:25:03 (3 items)

© Copyright IBM Corporation 2015

Figure 3-7. IBM MQ Explorer default views WM6461.0

Notes:

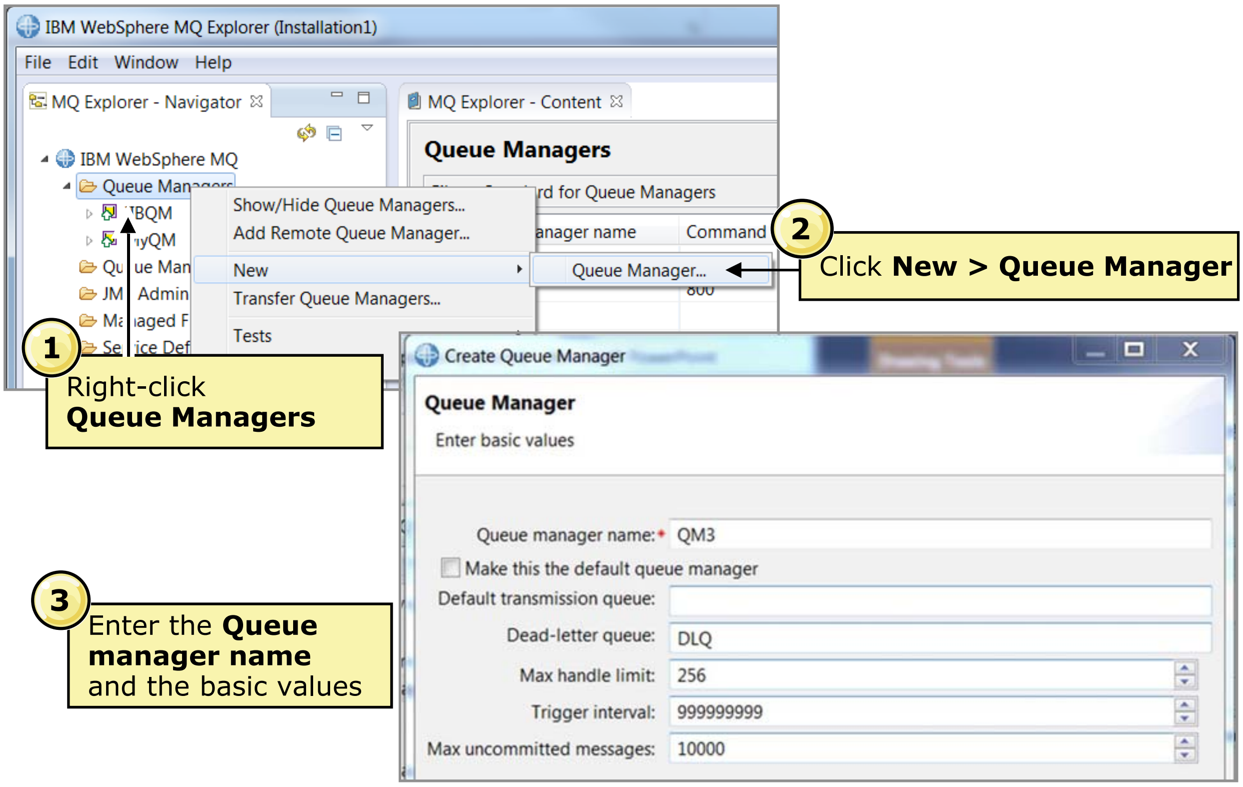
The figure shows the general layout of the IBM MQ Explorer user interface.

The **Navigator** view on the left contains a list of all the objects that IBM MQ Explorer manages. Status icons next to some of the objects, such as queue managers, indicate the status of the object.

The **Content** views contain more information about the object that is selected in the Navigator. For example, if the **Queues** folder under a queue manager is selected in the Navigator view, the content view contains information about the queues on that queue manager.

By using the **Scheme** icon (bottom portion of the **Queues** content view), you can edit the current scheme and select the information to display in the queue property columns. You can also change the order of the columns.

Creating a local queue manager with IBM MQ Explorer (1 of 3)


IBM WebSphere MQ Explorer (Installation1)
File Edit Window Help
MQ Explorer - Navigator
MQ Explorer - Content
IBM WebSphere MQ
Queue Managers
Show/Hide Queue Managers...
Add Remote Queue Manager...
New
Transfer Queue Managers...
Tests
Queue Manager...
Command
1 Right-click Queue Managers
2 Click New > Queue Manager
3 Enter the Queue manager name and the basic values
Create Queue Manager

Queue Manager

Enter basic values

Queue manager name:* QM3

Make this the default queue manager

Default transmission queue:

Dead-letter queue: DLQ

Max handle limit: 256

Trigger interval: 999999999

Max uncommitted messages: 10000

Figure 3-8. Creating a local queue manager with IBM MQ Explorer (1 of 3) WM6461.0

Notes:

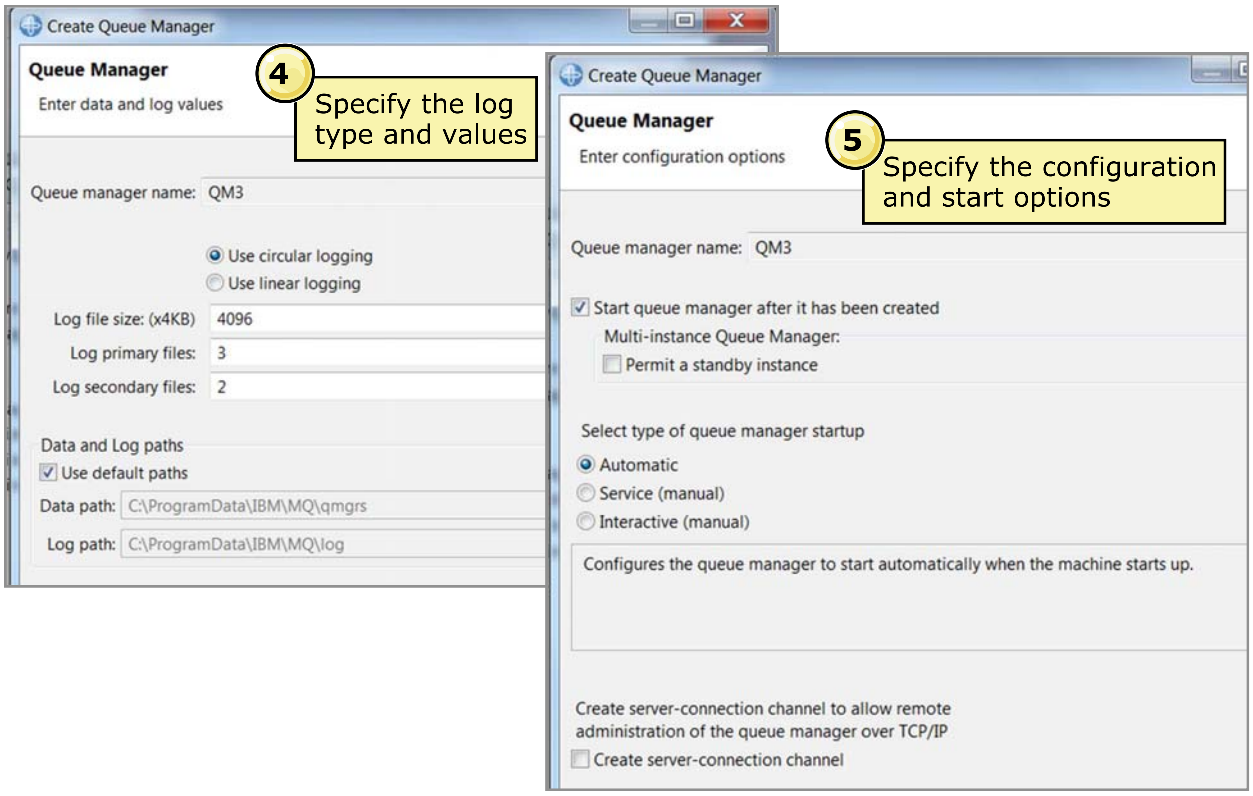
A queue manager can be created from a command or by using IBM MQ Explorer. IBM MQ Explorer automatically shows all local queue managers regardless of the method that is used to create them.

The figure shows the first three steps for adding a queue manager to the server by using the IBM MQ Explorer.

1. Right-click **Queue Managers** in the **MQ Explorer - Navigator** view.
2. Click **New > Queue Manager**.
3. Specify a queue manager name and other basic configuration. Basic configuration includes specifying a default transmission queue, dead-letter queue, and trigger interval.

   You should always specific a dead-letter queue for queue manager to ensure that errors are handled correctly.

Creating a local queue manager with IBM MQ Explorer (2 of 3)


WebSphere Education
IBM
Create Queue Manager
Queue Manager
Enter data and log values
4
Specify the log type and values
Create Queue Manager
Queue Manager
Enter configuration options
5
Specify the configuration and start options

Queue manager name: QM3

Use circular logging [✓]
Use linear logging

Log file size: (x4KB) 4096
Log primary files: 3
Log secondary files: 2

Data and Log paths
[✓] Use default paths
Data path: C:\ProgramData\IBM\MQ\qmgrs
Log path: C:\ProgramData\IBM\MQ\log

Queue manager name: QM3

[✓] Start queue manager after it has been created
Multi-instance Queue Manager:
[ ] Permit a standby instance

Select type of queue manager startup
[✓ Automatic]
Service (manual)
Interactive (manual)

Configures the queue manager to start automatically when the machine starts up.

Create server-connection channel to allow remote administration of the queue manager over TCP/IP
[ ] Create server-connection channel

© Copyright IBM Corporation 2015

Figure 3-9. Creating a local queue manager with IBM MQ Explorer (2 of 3) WM6461.0

Notes:

4. The next step for creating a queue manager is to specify the queue manager log values.

   The log data is held in a series of files called log files. The log file size is specified in units of 4 KB pages.

   For more information about the IBM MQ log files, see the IBM Knowledge Center for IBM MQ.

5. The next step for creating a queue manager is to enter the queue manager configuration options.

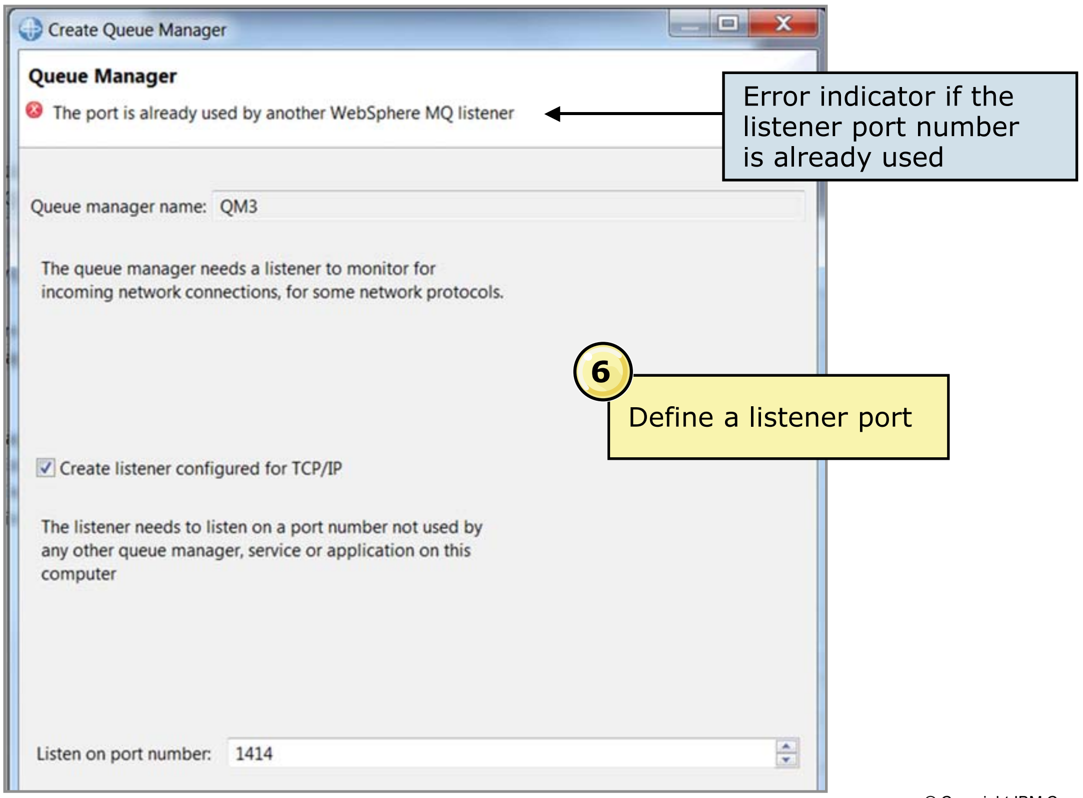
   Select **Start queue manager after it has been created** if you want the queue manager to start immediately after it is created.

   Select **Automatic** if you want the queue manager to automatically start after a system restart.

   Select **Create server-connection channel** to allow another administrator on another computer to manage this queue manager on this computer.

WebSphere Education

Creating a local queue manager with IBM MQ Explorer (3 of 3)



The port is already used by another WebSphere MQ listener

Error indicator if the listener port number is already used

Queue manager name: QM3

The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

☑ Create listener configured for TCP/IP

The listener needs to listen on a port number not used by any other queue manager, service or application on this computer

Listen on port number: 1414

Figure 3-10. Creating a local queue manager with IBM MQ Explorer (3 of 3) WM6461.0

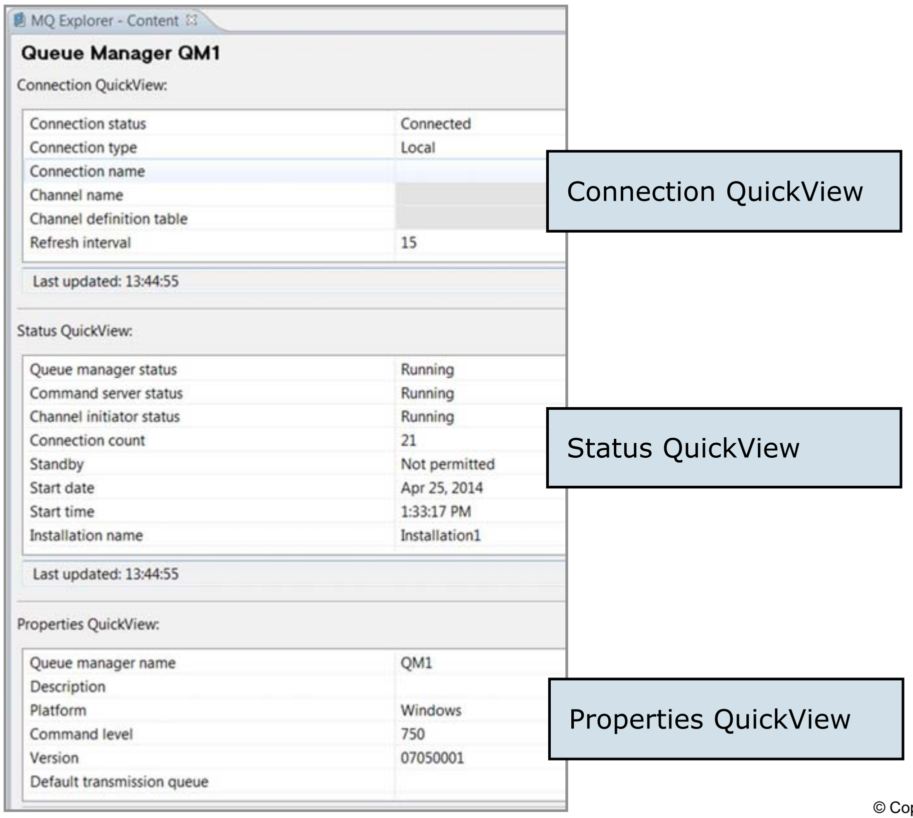
Notes:

6. Create a listener for TCP/IP and identify the listener port value.

   A unique port number is required for each listener on the same host. If IBM MQ detects that another listener is using TCP/IP port number, an error message is displayed at the top of the page, as shown in the figure.

WebSphere Education IBM

Queue manager content view



Queue Manager QM1

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15

Last updated: 13:44:55

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	21
Standby	Not permitted
Start date	Apr 25, 2014
Start time	1:33:17 PM
Installation name	Installation1

Last updated: 13:44:55

Properties QuickView:

Queue manager name	QM1
Description	
Platform	Windows
Command level	750
Version	07050001
Default transmission queue	

Figure 3-11. Queue manager content view WM6461.0

Notes:

The Queue Manager content view is divided into three QuickView panes.

- The **Connection QuickView** shows the queue manager connection status information, such as connection status, connection type, and connection name.
- The **Status QuickView** shows the status of the queue manager. It includes the command server status, channel initiator status, and the installation name.
- The **Properties QuickView** shows some of the configurable properties of the queue manager such as the operating system, command level (version of IBM MQ), and the default transmission queue.

WebSphere Education

Creating a local queue with IBM MQ Explorer (1 of 2)

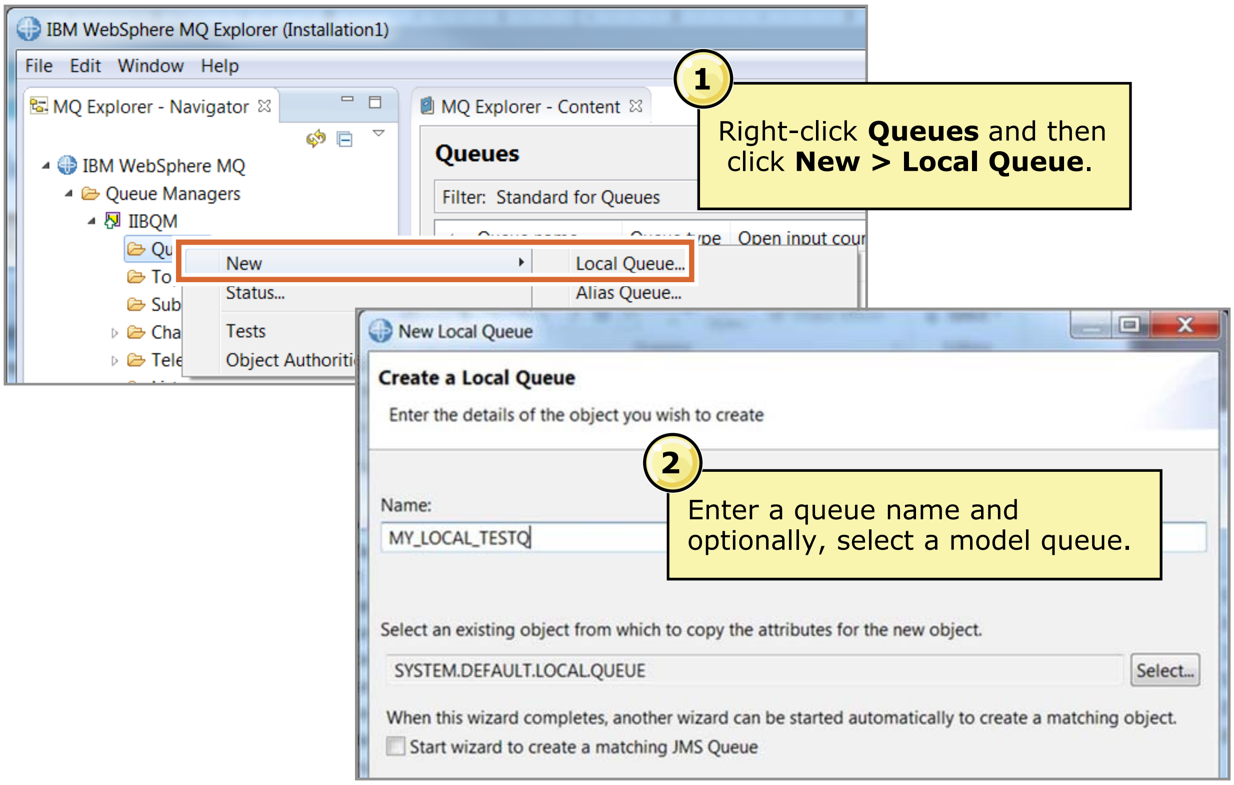

IBM WebSphere MQ Explorer (Installation1)
File Edit Window Help
MQ Explorer - Navigator
MQ Explorer - Content
IBM WebSphere MQ
Queue Managers
IBQM
Queues
Filter: Standard for Queues
New
Local Queue...
Status...
Alias Queue...
Tests
Object Authoriti
New Local Queue
Create a Local Queue
Enter the details of the object you wish to create
Name:
MY_LOCAL_TESTQ
Select an existing object from which to copy the attributes for the new object.
SYSTEM.DEFAULT.LOCAL.QUEUE
Select...
When this wizard completes, another wizard can be started automatically to create a matching object.
Start wizard to create a matching JMS Queue
1
Right-click **Queues** and then click **New > Local Queue**.
2
Enter a queue name and optionally, select a model queue.

Figure 3-12. Creating a local queue with IBM MQ Explorer (1 of 2) WM6461.0

Notes:

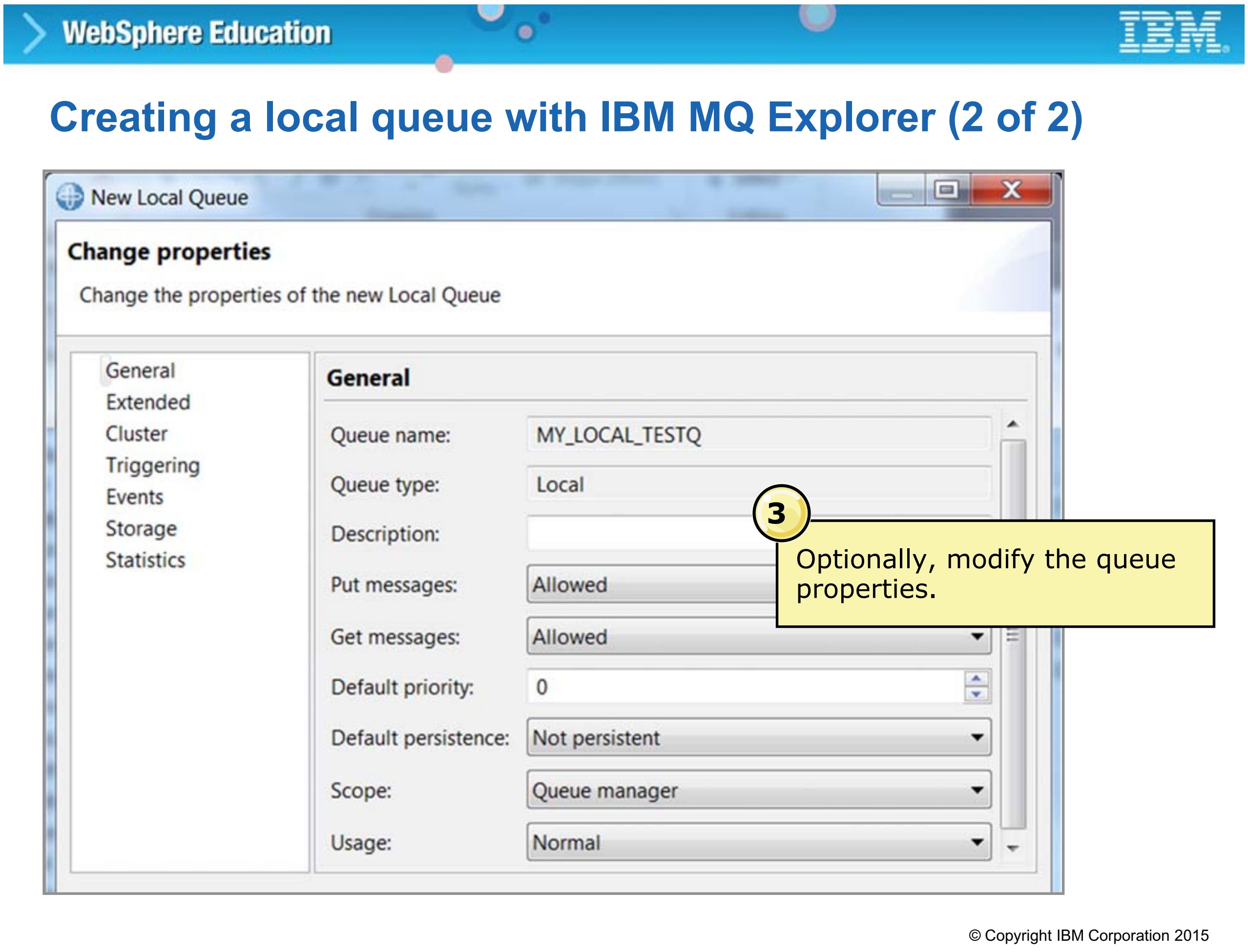
It is possible to define queues for any local queue manager that is connected to IBM MQ Explorer.

To create a new local queue:

1. Right-click the **Queues** folder under the queue manager in the **MQ Explorer - Navigator** view and then click **New > Local Queue**.
2. Enter a queue name.

   By default, IBM MQ uses a default SYSTEM queue as the model for the new queue. You can specify a different queue to use as the model for this queue if you want.

   There is also an option to automatically start another wizard to create a matching JMS queue.



Creating a local queue with IBM MQ Explorer (2 of 2)

New Local Queue

Change properties

Change the properties of the new Local Queue

General

Queue name: MY_LOCAL_TESTQ

Queue type: Local

Description:

Put messages: Allowed

Get messages: Allowed

Default priority: 0

Default persistence: Not persistent

Scope: Queue manager

Usage: Normal

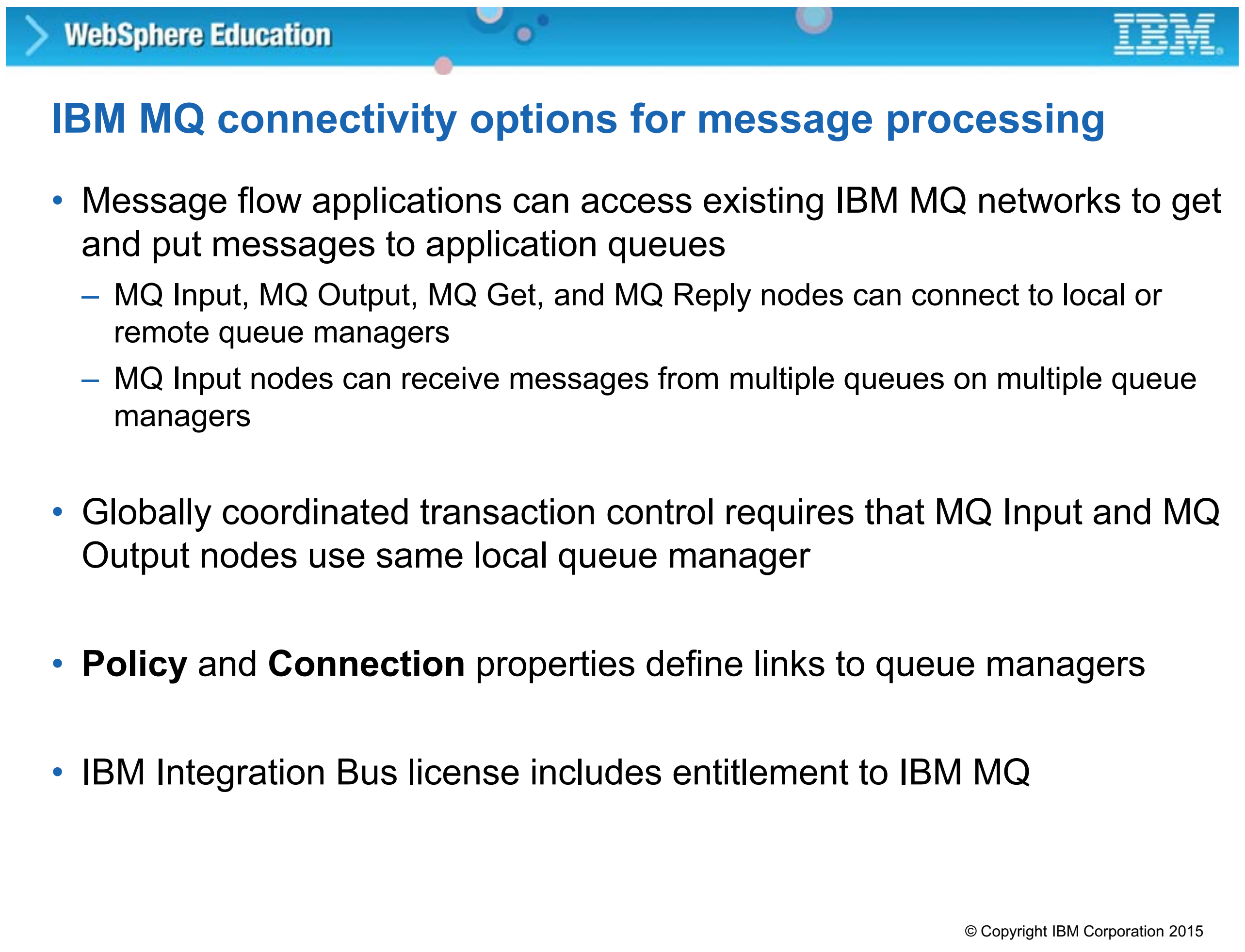
3 Optionally, modify the queue properties.

Figure 3-13. Creating a local queue with IBM MQ Explorer (2 of 2) WM6461.0

Notes:

3. The next step is to modify the queue properties, if necessary. Functions are used to categorize the queue properties:
   - General
   - Extended
   - Cluster
   - Triggering
   - Events
   - Storage
   - Statistics

3.2. Connectivity to IBM Integration Bus



IBM MQ connectivity options for message processing

- Message flow applications can access existing IBM MQ networks to get and put messages to application queues
  - MQ Input, MQ Output, MQ Get, and MQ Reply nodes can connect to local or remote queue managers
  - MQ Input nodes can receive messages from multiple queues on multiple queue managers
- Globally coordinated transaction control requires that MQ Input and MQ Output nodes use same local queue manager
- **Policy** and **Connection** properties define links to queue managers
- IBM Integration Bus license includes entitlement to IBM MQ

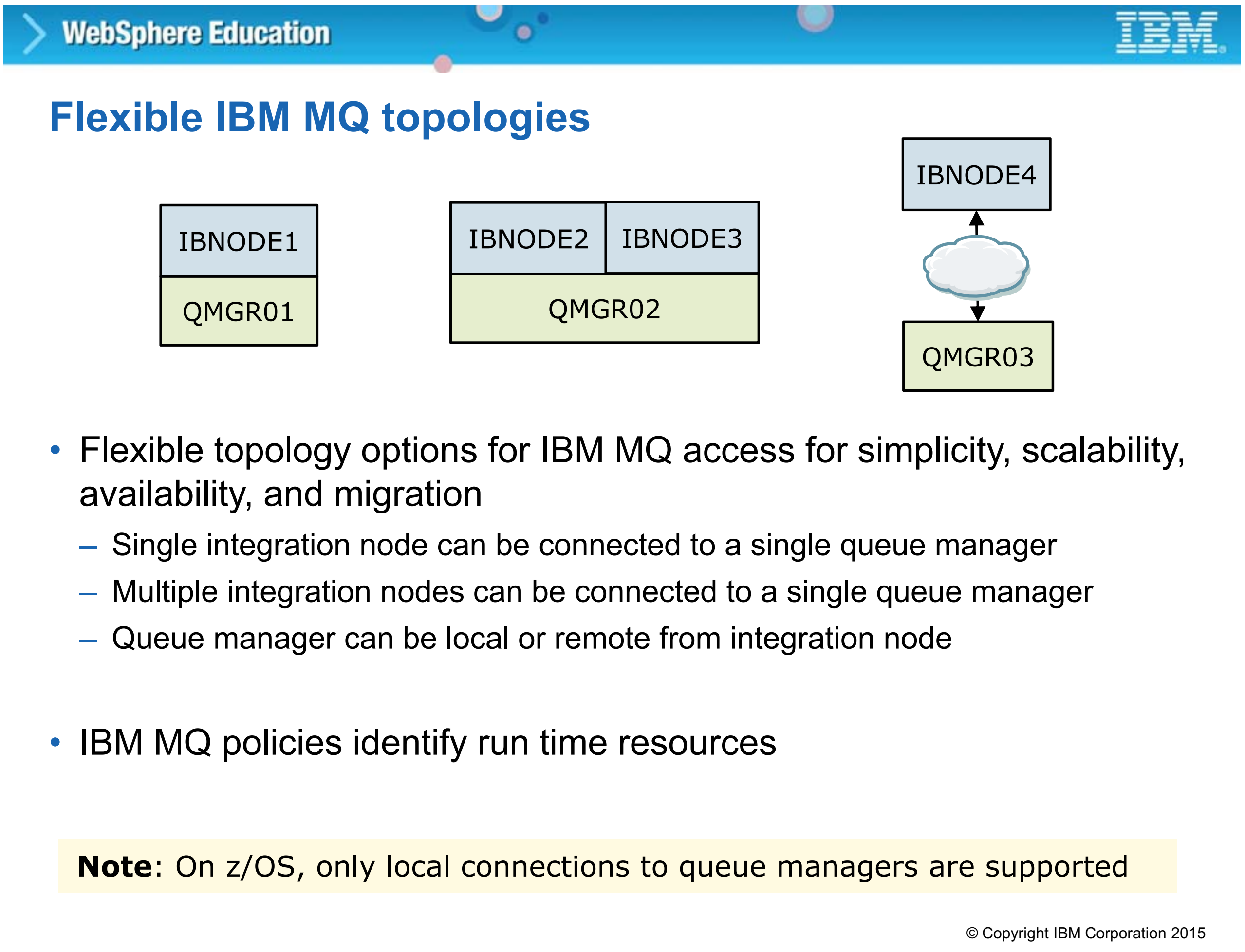
Figure 3-14. IBM MQ connectivity options for message processing WM6461.0

Notes:

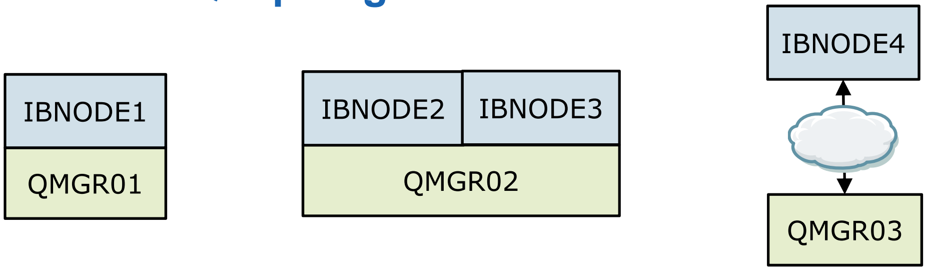
IBM Integration Bus contains processing nodes that can access existing IBM MQ networks to get and put messages.

You can set a property in an IBM MQ input node and output node to globally coordinate message flow transactions with a transaction manager to ensure the data integrity of transactions. For globally coordinated transactions on distributed systems, you must specify a local queue manager on the integration node. This queue manager is the global transaction manager, and no other IBM MQ resources can be used in the message flow. On z/OS, all queue managers are globally coordinated.

You can specify a connection from an IBM MQ node to a specific local or remote queue manager by using connection properties on the node, including the destination queue manager name, host name, port, and channel. Alternatively, you can use an MQEndpoint policy to control the values of IBM MQ node connection properties at run time, or to specify an IBM MQ broker for event publication.



Flexible IBM MQ topologies



- Flexible topology options for IBM MQ access for simplicity, scalability, availability, and migration
  - Single integration node can be connected to a single queue manager
  - Multiple integration nodes can be connected to a single queue manager
  - Queue manager can be local or remote from integration node
- IBM MQ policies identify run time resources

Note: On z/OS, only local connections to queue managers are supported

Figure 3-15. Flexible IBM MQ topologies WM6461.0

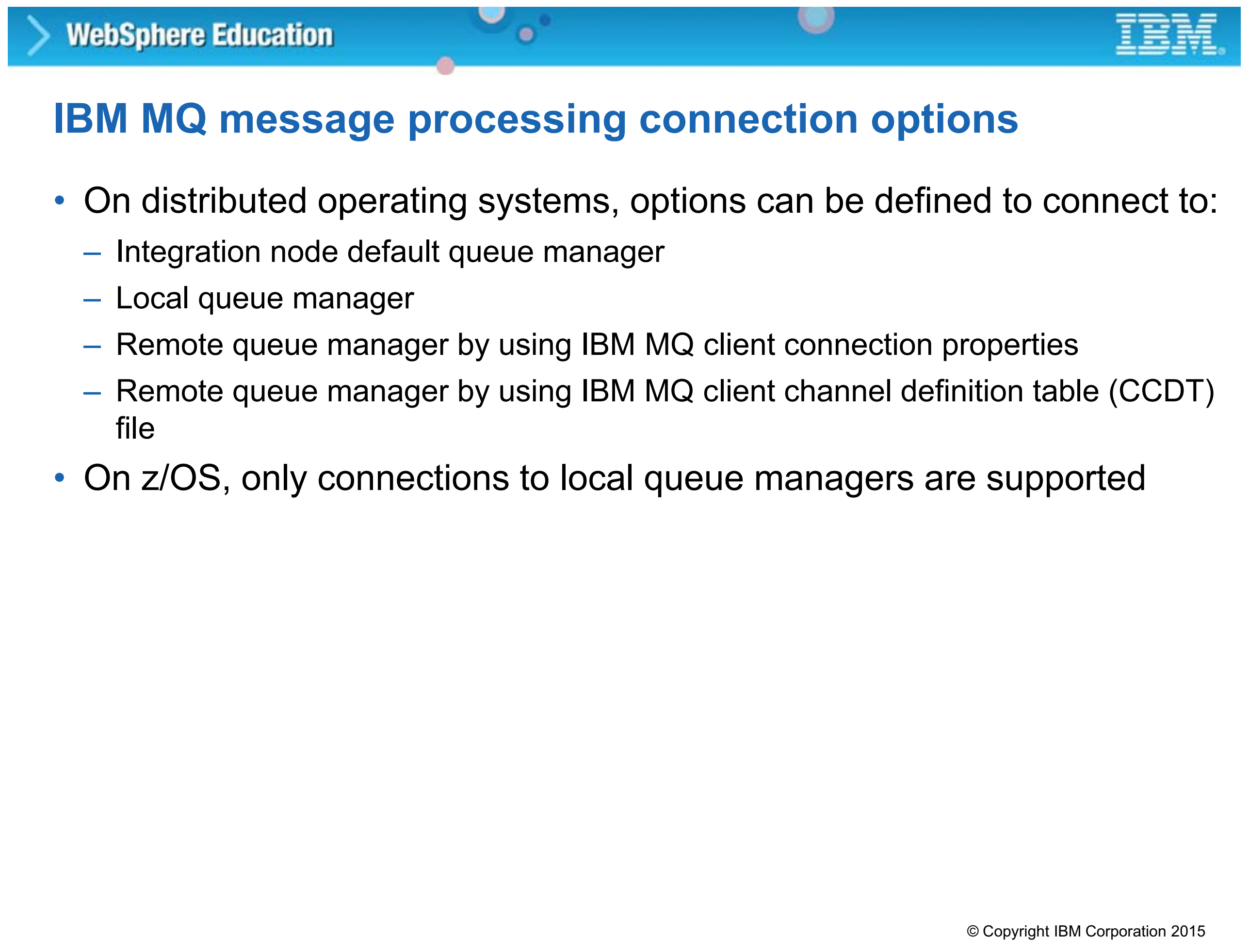
Notes:

On distributed systems, IBM Integration Bus supports local and remote queue managers.

On distributed systems, support for IBM MQ is extended. You can configure local or client connections to IBM MQ so that the integration nodes can get messages from or put messages to queues on any local or remote queue manager. On z/OS, you can have IBM MQ message flow nodes connect to different local queue managers, not just the queue manager that is specified on the integration node.

If the IBM MQ queue manager is running on the same computer as the integration node, you can specify a local connection to the queue manager. Alternatively, if the queue manager is hosted on a separate computer from IBM Integration Bus, you can configure a client connection from the integration node so that it can access the messages on the remote queue manager.

You can also create message flows that contain multiple MQInput and MQOutput nodes, each of which can access different queue managers as specified in the node. Using this feature, you can adapt the message flows to your existing IBM MQ topologies.



IBM MQ message processing connection options

- On distributed operating systems, options can be defined to connect to:
  - Integration node default queue manager
  - Local queue manager
  - Remote queue manager by using IBM MQ client connection properties
  - Remote queue manager by using IBM MQ client channel definition table (CCDT) file
- On z/OS, only connections to local queue managers are supported

© Copyright IBM Corporation 2015

Figure 3-16. IBM MQ message processing connection options WM6461.0

Notes:

Integration Bus provides message processing nodes for handling messages that are received from or sent to IBM MQ applications.

You can configure a local or client connection to IBM MQ to enable the Integration Bus message flows to access messages on IBM MQ queues.

On distributed operating systems, if you want to connect to a remote queue manager to process IBM MQ messages, you must install either an IBM MQ Client or an IBM MQ Server on the same computer as IBM Integration Bus. You also must install an IBM MQ Server on the computer that is running your queue manager.

On z/OS, IBM MQ is a prerequisite for IBM Integration Bus, and only local connections to queue managers are supported.

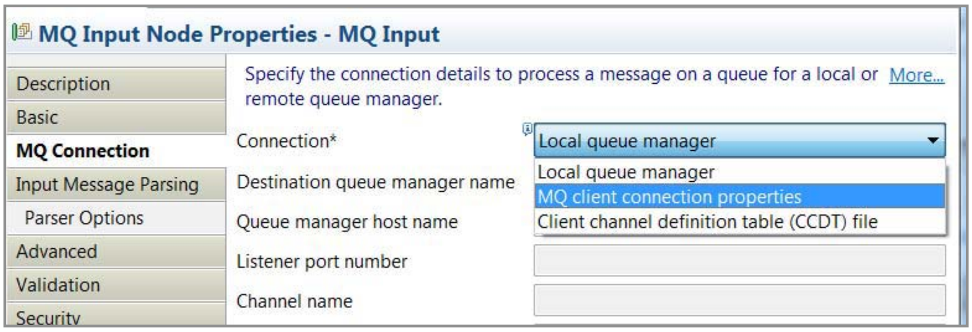
The developer specifies the connection information on the **MQ Connection** properties tab of the message processing nodes in the Integration Toolkit. The system administrator can use an MQEndpoint policy to override the IBM MQ connection properties at run time.

WebSphere Education IBM

Client connections to IBM MQ

- To connect to a remote queue manager, developers can define a client connection on the IBM MQ nodes by using one of the following options:
  - Select **MQ client connection properties** and specify the connection details of the queue manager explicitly
  - Select **Client channel definition table** (CCDT) and specify the location of the CCDT file by running the **`mqsichangeproperties`** command with the integration node registry object property **`mqCCDT`**

  Example:
  ```
  mqsichangeproperties IBNODE -o BrokerRegistry -n mqCCDT
  -v "C:\Program Files\IBM\MQ\Qmgrs\QM1\@ipcc\AMQCLCHL.TAB"
  ```


MQ Input Node Properties - MQ Input

Specify the connection details to process a message on a queue for a local or remote queue manager. More...

Connection*: Local queue manager

Destination queue manager name

Queue manager host name

Listener port number

Channel name

Figure 3-17. Client connections to IBM MQ WM6461.0

Notes:

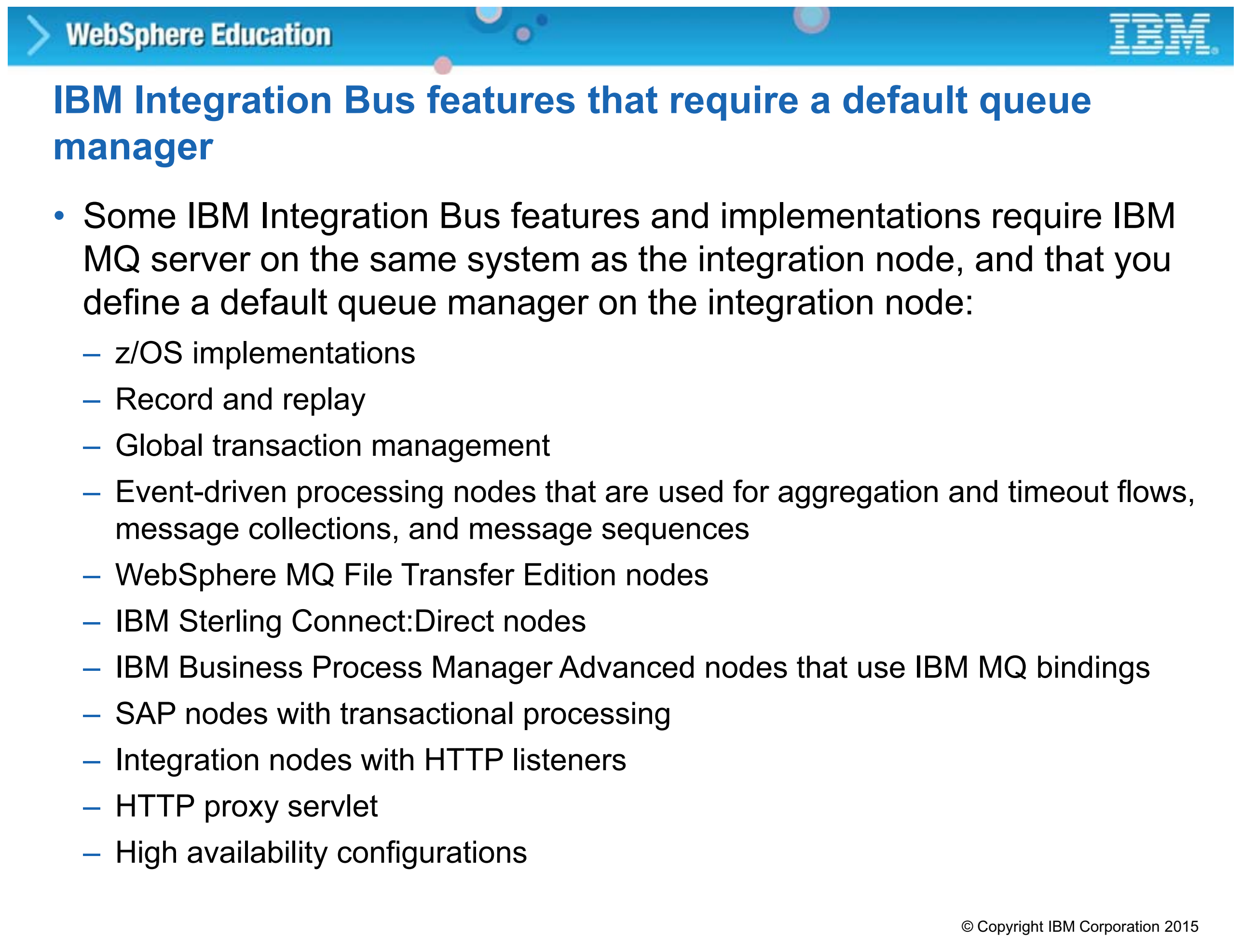
You can choose to configure either a local or client connection between your integration node and your queue manager, depending on the configuration of your existing architecture.

If your queue manager is running on the same computer as your integration node, you can specify a local connection. You can specify a local connection by choosing a specific queue manager on an IBM MQ node or by using the default queue manager that is specified on the integration node.

Alternatively, if IBM MQ is installed on separate computer from Integration Bus, you can define a client connection by configuring the connection details on the IBM MQ node or policy, or by specifying a client channel definition table (CCDT) to control the client connection information.

Important

If you want to connect to a remote queue manager from Integration Bus to process IBM MQ messages, you must install either an IBM MQ client or an IBM MQ Server on the same computer as Integration Bus.



WebSphere Education

IBM Integration Bus features that require a default queue manager

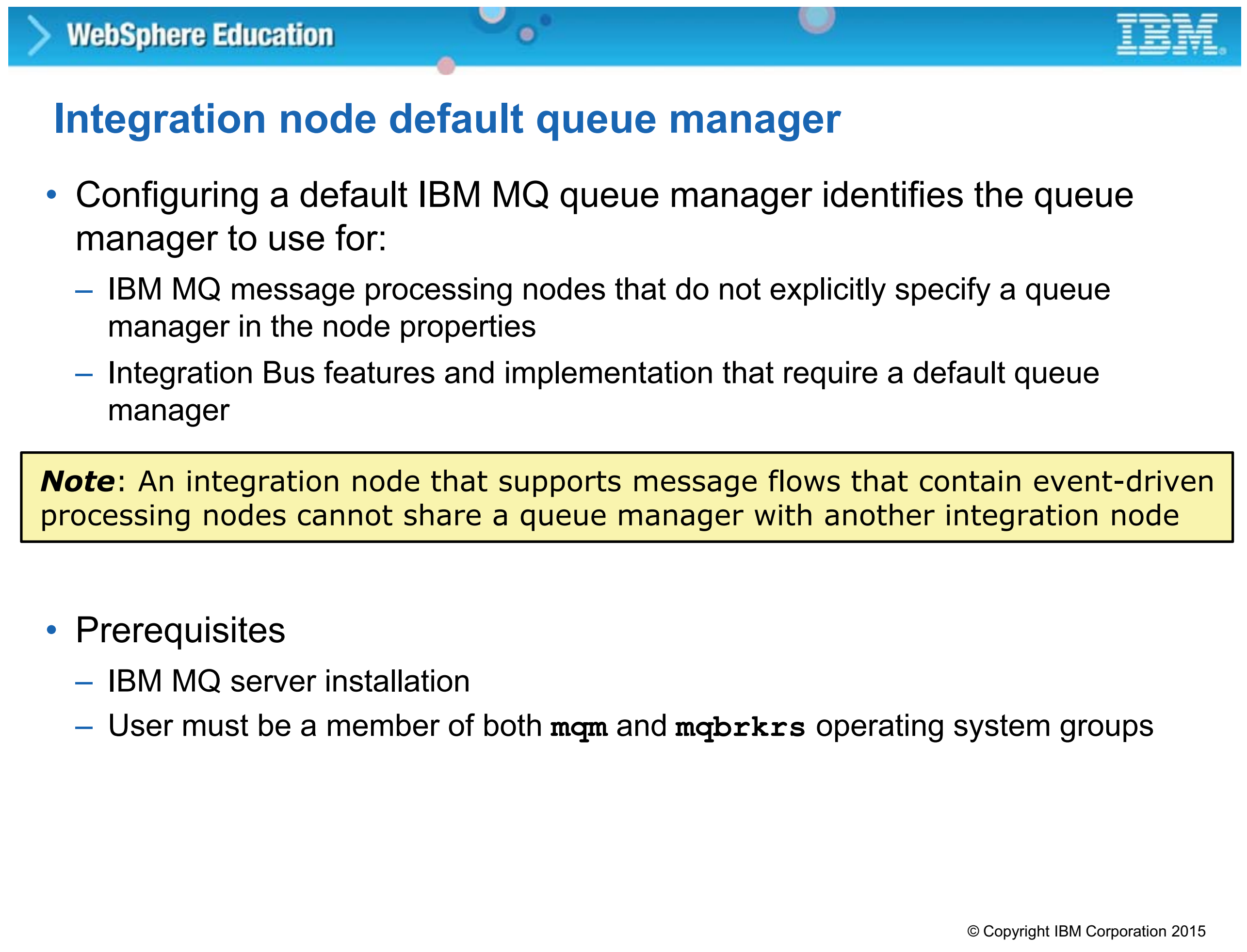
- Some IBM Integration Bus features and implementations require IBM MQ server on the same system as the integration node, and that you define a default queue manager on the integration node:
  - z/OS implementations
  - Record and replay
  - Global transaction management
  - Event-driven processing nodes that are used for aggregation and timeout flows, message collections, and message sequences
  - WebSphere MQ File Transfer Edition nodes
  - IBM Sterling Connect:Direct nodes
  - IBM Business Process Manager Advanced nodes that use IBM MQ bindings
  - SAP nodes with transactional processing
  - Integration nodes with HTTP listeners
  - HTTP proxy servlet
  - High availability configurations

Figure 3-18. IBM Integration Bus features that require a default queue manager WM6461.0

Notes:

On z/OS, IBM MQ is required for installation, and only local connections to queue managers are supported. You must have a queue manager that is specified for the integration node, but you can also connect to other local queue managers on IBM MQ message flow nodes by using server bindings for the connection.

The integration node listener requires access to IBM MQ Server, so you must install it if you want to use an integration node listener to manage HTTP messages in your HTTP or SOAP flows. However, if you use HTTP nodes or SOAP nodes with the integration server embedded listener, they do not require access to IBM MQ.



WebSphere Education IBM

Integration node default queue manager

- Configuring a default IBM MQ queue manager identifies the queue manager to use for:
  - IBM MQ message processing nodes that do not explicitly specify a queue manager in the node properties
  - Integration Bus features and implementation that require a default queue manager

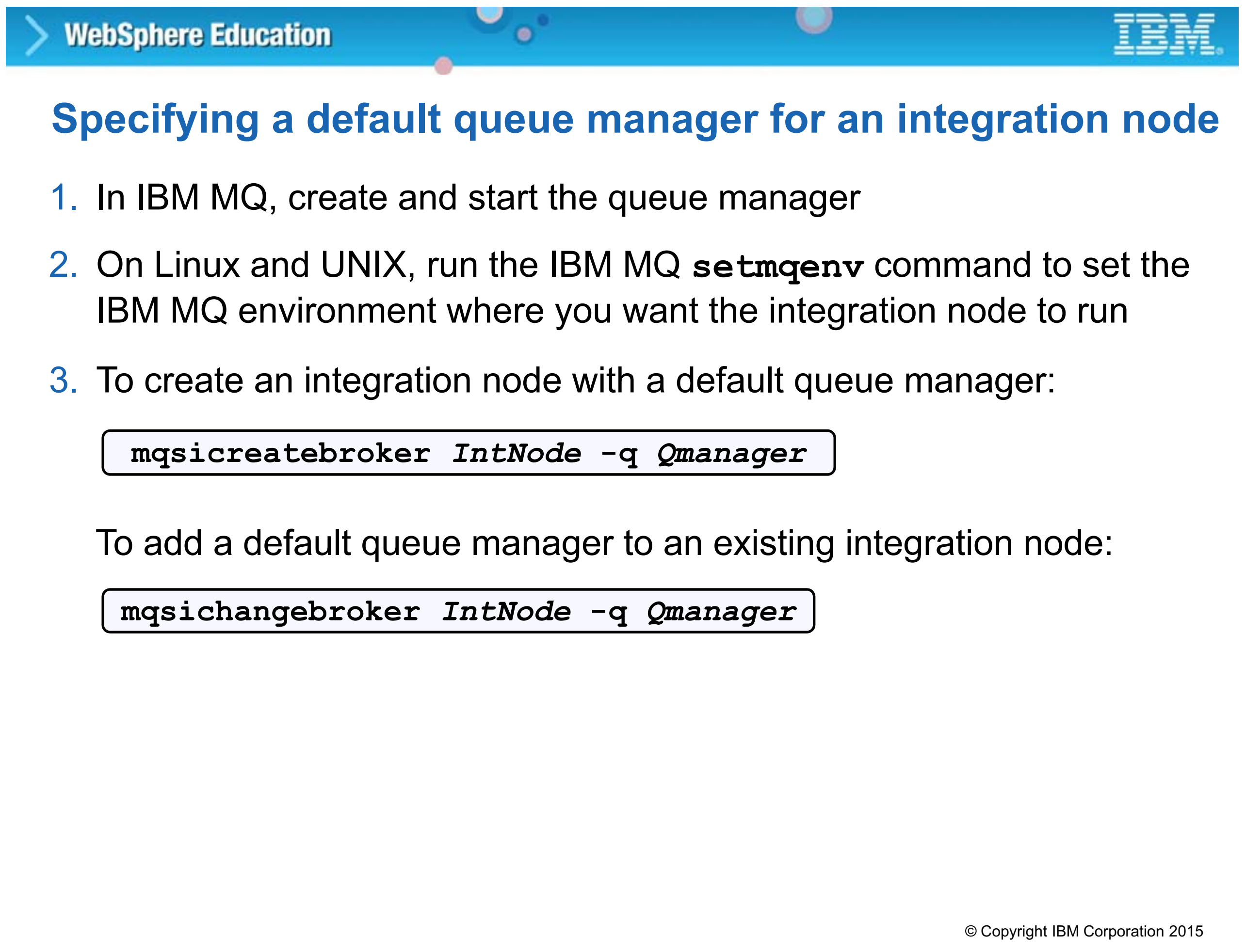
Note: An integration node that supports message flows that contain event-driven processing nodes cannot share a queue manager with another integration node

- Prerequisites
  - IBM MQ server installation
  - User must be a member of both **`mqm`** and **`mqbrkrs`** operating system groups

Figure 3-19. Integration node default queue manager WM6461.0

Notes:

Some Integration Bus capabilities require access to IBM MQ components. A subset of these capabilities require a set of default system queues to be created on the queue manager that is associated with the integration node.



WebSphere Education

Specifying a default queue manager for an integration node

1. In IBM MQ, create and start the queue manager
2. On Linux and UNIX, run the IBM MQ **setmqenv** command to set the IBM MQ environment where you want the integration node to run
3. To create an integration node with a default queue manager:

```
mqsicreatebroker IntNode -q Qmanager
```

To add a default queue manager to an existing integration node:

```
mqsichangebroker IntNode -q Qmanager
```

Figure 3-20. Specifying a default queue manager for an integration node WM6461.0

Notes:

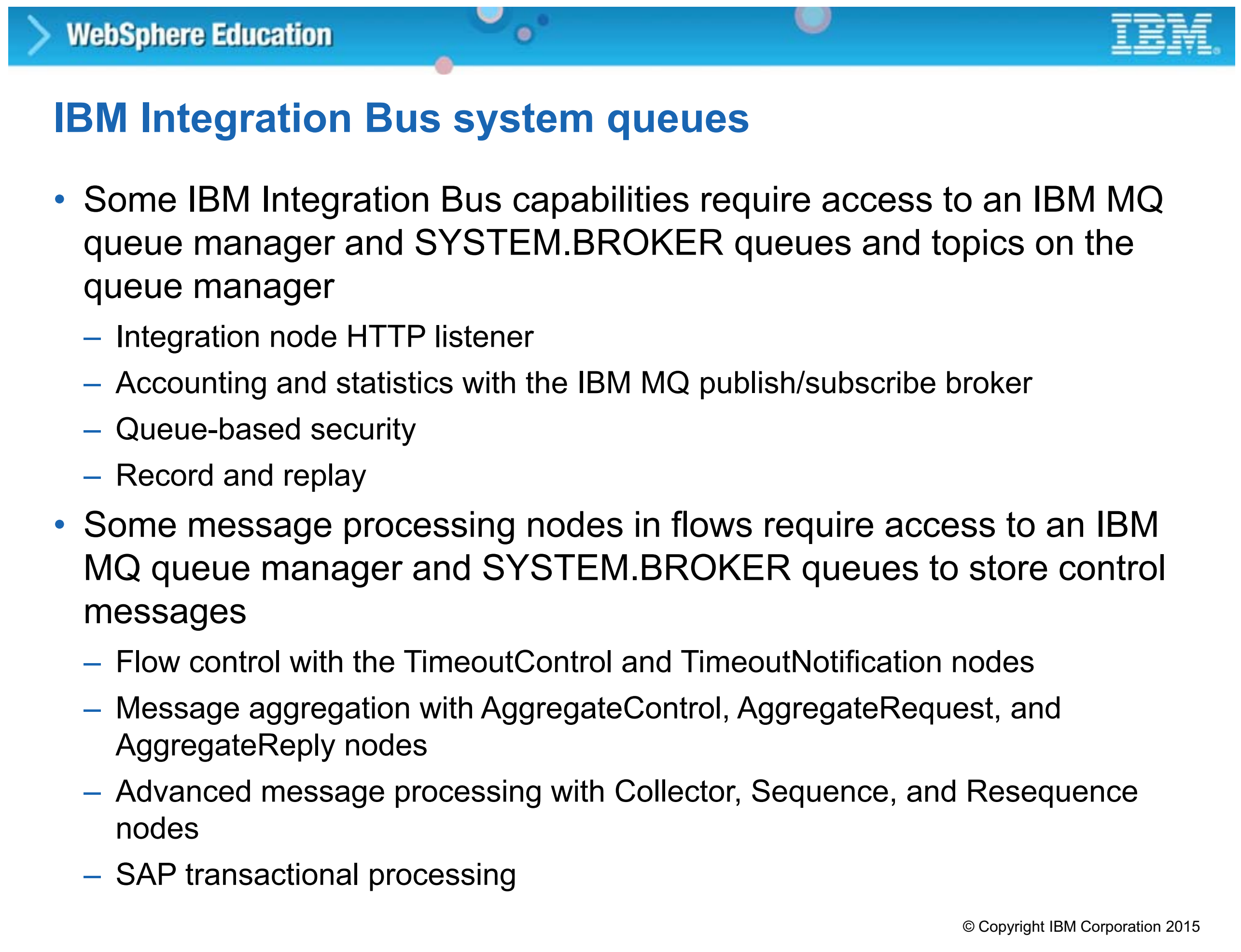
You can create a default queue manager that is used by default for IBM MQ processing in the message flow if no queue manager is specified on the IBM MQ node.

You create the integration node by using the **mqsicreatebroker** command. If you specify the optional **-q** parameter, the named local queue manager is associated with the integration node.

On z/OS, the specified queue manager must exist locally, and the required queues are created automatically. If the named queue manager does not exist, the **mqsicreatebroker** command fails, and an error is generated.

On Windows only, the **mqsicreatebroker** command installs a service under which the integration node runs.

If the integration node exists, you can modify the integration node to use a default queue manager by using the **mqsichangebroker** command.



WebSphere Education

IBM Integration Bus system queues

- Some IBM Integration Bus capabilities require access to an IBM MQ queue manager and SYSTEM.BROKER queues and topics on the queue manager
  - Integration node HTTP listener
  - Accounting and statistics with the IBM MQ publish/subscribe broker
  - Queue-based security
  - Record and replay
- Some message processing nodes in flows require access to an IBM MQ queue manager and SYSTEM.BROKER queues to store control messages
  - Flow control with the TimeoutControl and TimeoutNotification nodes
  - Message aggregation with AggregateControl, AggregateRequest, and AggregateReply nodes
  - Advanced message processing with Collector, Sequence, and Resequence nodes
  - SAP transactional processing

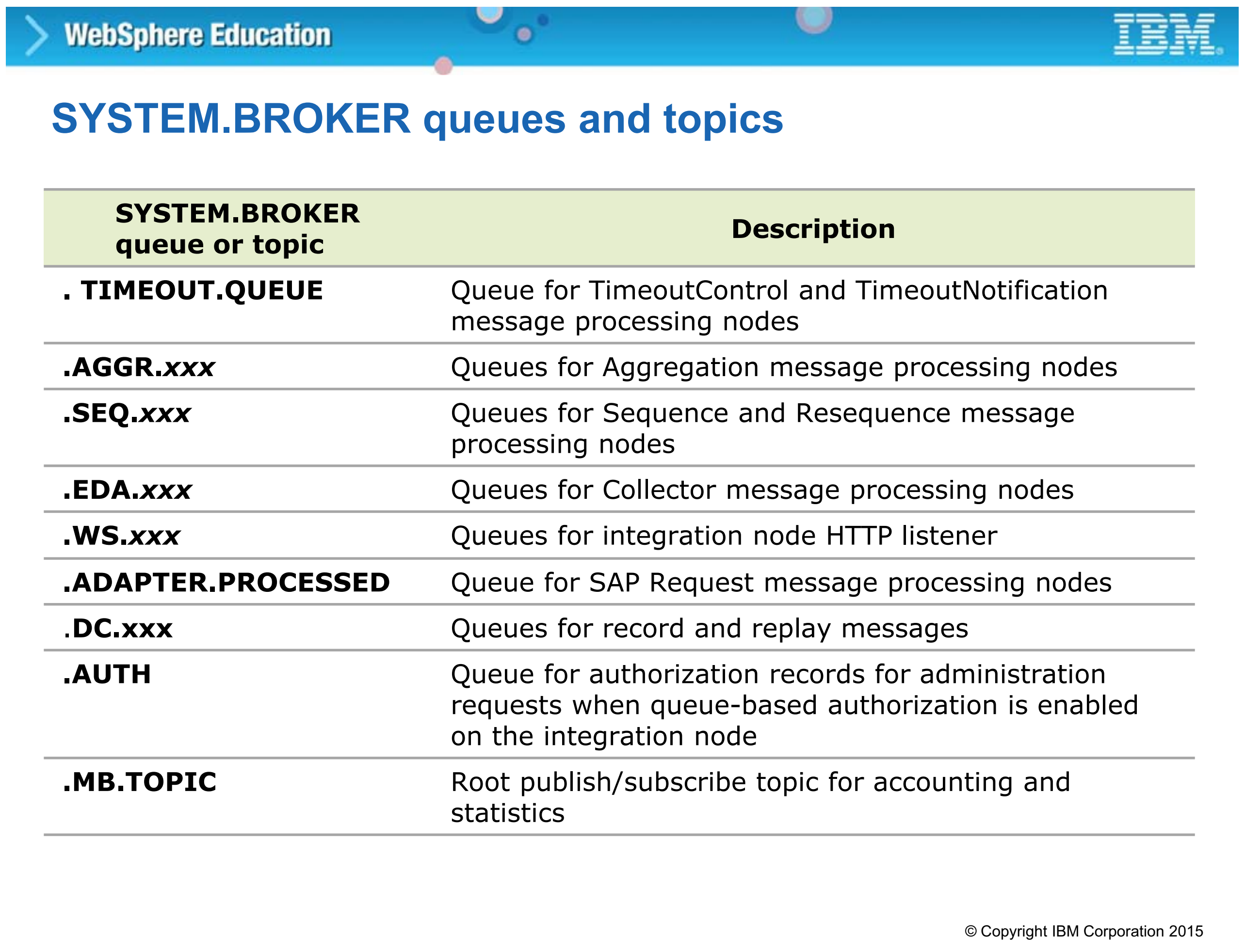
© Copyright IBM Corporation 2015

Figure 3-21. IBM Integration Bus system queues WM6461.0

Notes:

IBM MQ is no longer a prerequisite for using Integration Bus on distributed systems. However, some Integration Bus capabilities require access to IBM MQ components, and a subset of these capabilities require a set of default system queues on the queue manager that is associated with the integration node.

The SYSTEM.BROKER queues are used to store information about in-flight messages.



WebSphere Education

SYSTEM.BROKER queues and topics

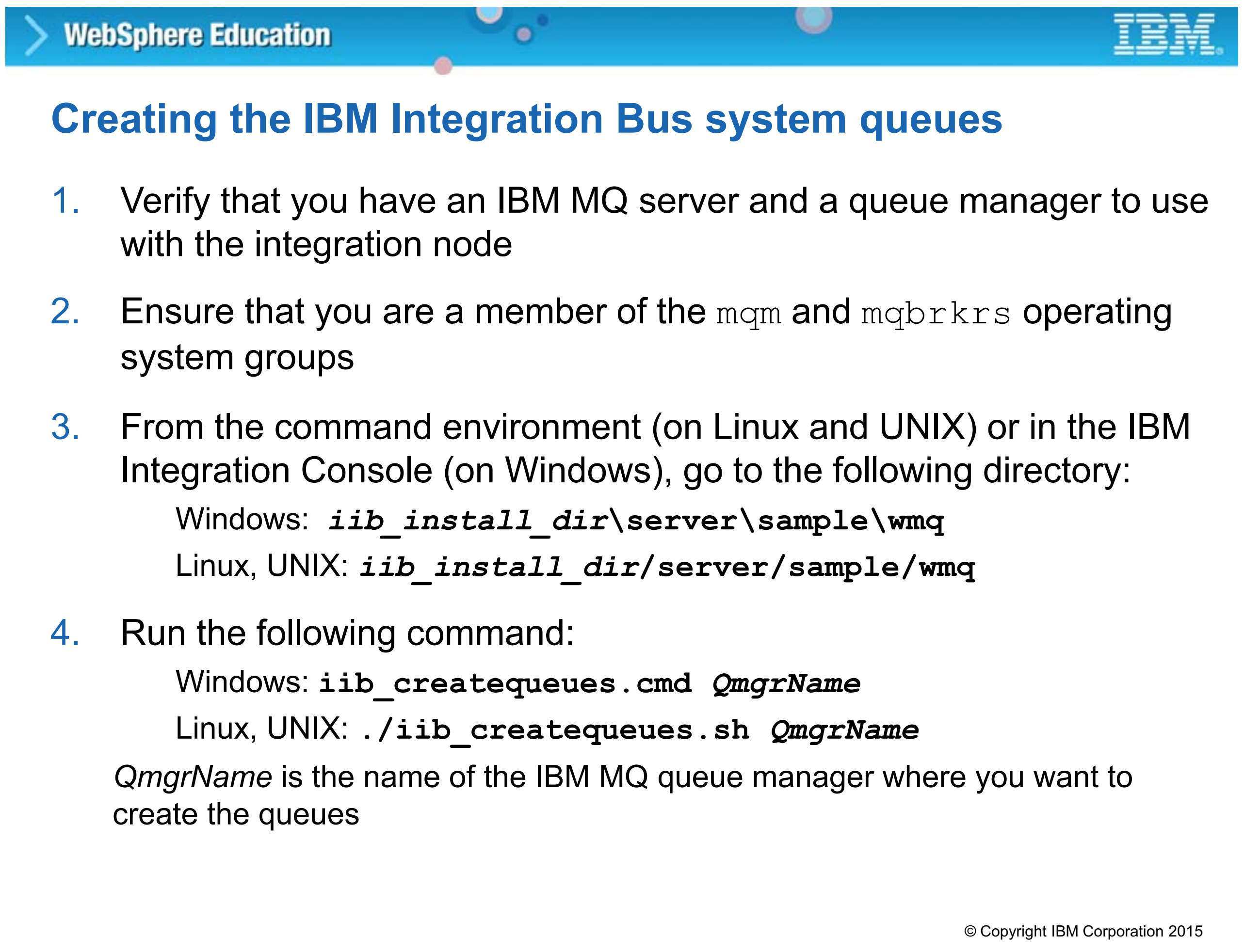
SYSTEM.BROKER queue or topic	Description
. TIMEOUT.QUEUE	Queue for TimeoutControl and TimeoutNotification message processing nodes
.AGGR.*xxx*	Queues for Aggregation message processing nodes
.SEQ.*xxx*	Queues for Sequence and Resequence message processing nodes
.EDA.*xxx*	Queues for Collector message processing nodes
.WS.*xxx*	Queues for integration node HTTP listener
.ADAPTER.PROCESSED	Queue for SAP Request message processing nodes
.DC.xxx	Queues for record and replay messages
.AUTH	Queue for authorization records for administration requests when queue-based authorization is enabled on the integration node
.MB.TOPIC	Root publish/subscribe topic for accounting and statistics

Figure 3-22. SYSTEM.BROKER queues and topics WM6461.0

Notes:

The names of Integration Bus queues begin with the reserved characters SYSTEM.BROKER. These queues are in addition to the default IBM MQ objects that are created when you install that product.

The figure summarizes the SYSTEM.BROKER queues. You work with some of these queues such as the SYSTEM.BROKER.AUTH queue, later in this course.



WebSphere Education

Creating the IBM Integration Bus system queues

1. Verify that you have an IBM MQ server and a queue manager to use with the integration node
2. Ensure that you are a member of the `mqm` and `mqbrkrs` operating system groups
3. From the command environment (on Linux and UNIX) or in the IBM Integration Console (on Windows), go to the following directory:

   Windows: ***iib_install_dir*\server\sample\wmq**

   Linux, UNIX: ***iib_install_dir*/server/sample/wmq**
4. Run the following command:

   Windows: **iib_createqueues.cmd *QmgrName***

   Linux, UNIX: **./iib_createqueues.sh *QmgrName***

   QmgrName is the name of the IBM MQ queue manager where you want to create the queues

Figure 3-23. Creating the IBM Integration Bus system queues WM6461.0

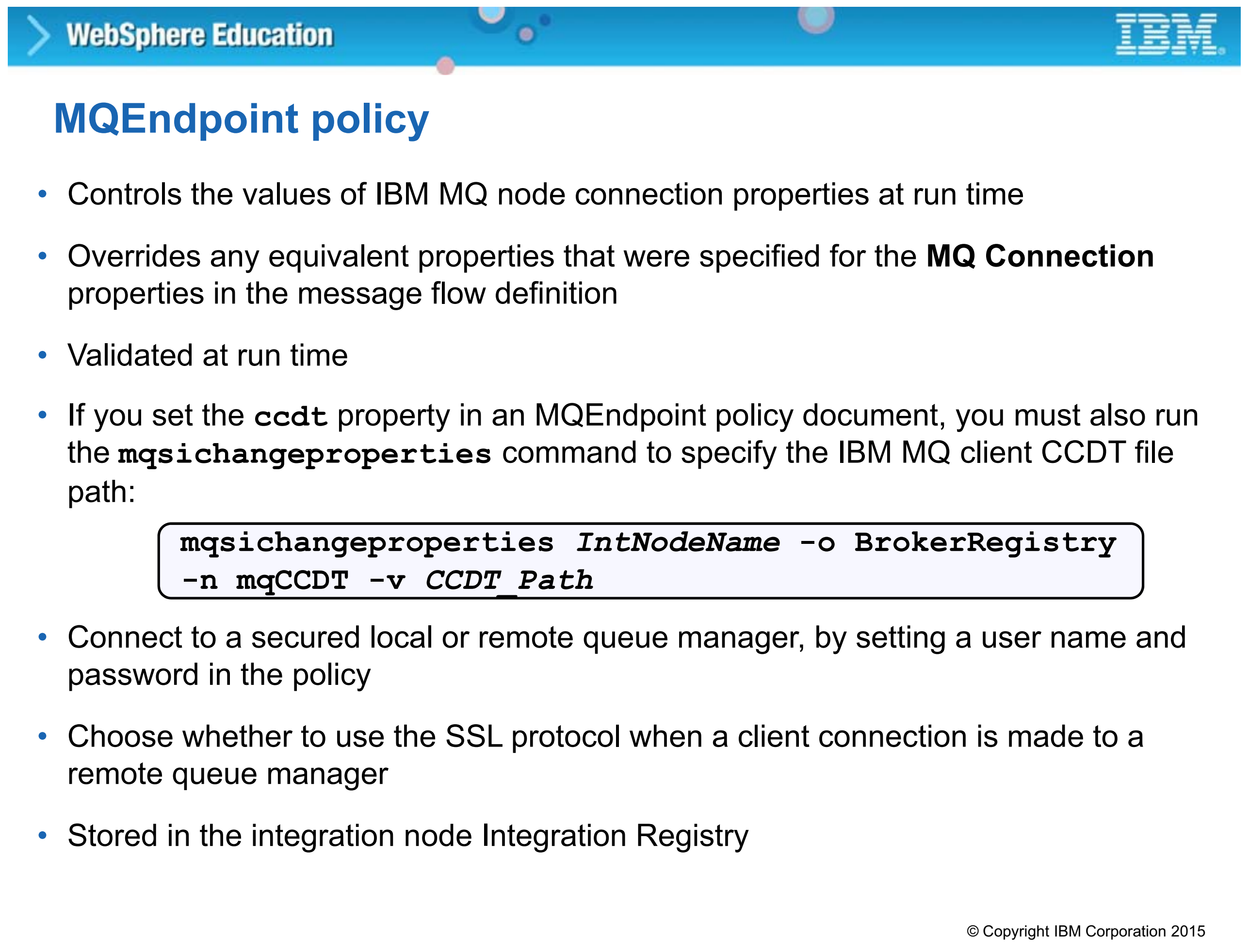
Notes:

You can create the default IBM Integration Bus queues on the queue manager that is associated with your integration node by running a script.

This figure lists the steps for creating the system queues that IBM Integration Bus needs for some functions.

1. You must have an IBM MQ server and a queue manager to use with your IBM Integration Bus deployment.
2. You must be a member of the `mqm` and `mqbrkrs` operating system groups. On Linux and UNIX, ensure that the command environment is initialized by running the **`mqsiprofile`** command.
3. Run the command scripts.

3.3. MQEndpoint policies



WebSphere Education IBM

MQEndpoint policy

- Controls the values of IBM MQ node connection properties at run time
- Overrides any equivalent properties that were specified for the **MQ Connection** properties in the message flow definition
- Validated at run time
- If you set the **`ccdt`** property in an MQEndpoint policy document, you must also run the **`mqsichangeproperties`** command to specify the IBM MQ client CCDT file path:

```
mqsichangeproperties IntNodeName -o BrokerRegistry
-n mqCCDT -v CCDT_Path
```

- Connect to a secured local or remote queue manager, by setting a user name and password in the policy
- Choose whether to use the SSL protocol when a client connection is made to a remote queue manager
- Stored in the integration node Integration Registry

Figure 3-24. MQEndpoint policy WM6461.0

Notes:

You can specify either a local or client connection to IBM MQ from an IBM message processing node by using one of the following methods:

- Set explicit connection properties for a specific IBM MQ node on the **MQ Connection** tab.
- Specify an MQEndpoint policy to control connection properties for one or more IBM MQ nodes on the **Policy** tab.

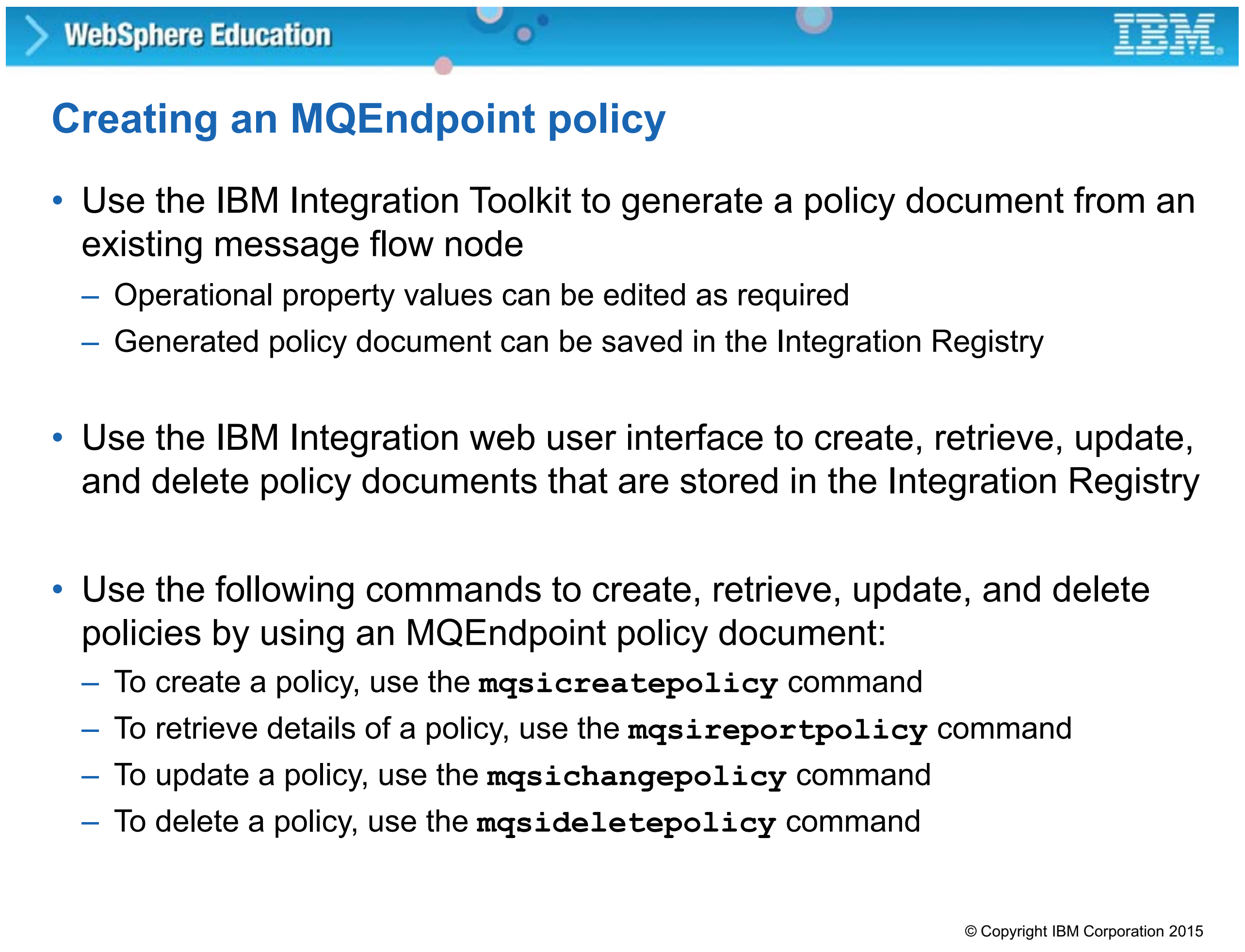
An MQEndpoint policy is an operational policy, which dynamically controls the connection properties that are applied at run time. It overrides any equivalent properties that were specified in the message flow node properties.

You can use a single MQEndpoint policy for multiple IBM MQ nodes. When you update the connection properties in the MQEndpoint policy, the updated properties are automatically applied to all IBM MQ nodes that have an attached policy.

MQEndpoint policies are stored in the integration node Integration Registry.

You can create and configure MQEndpoint policies for IBM MQ nodes by using any of the following methods:

- Create the MQEndpoint policy when configuring the IBM MQ node in Integration Toolkit
- Use the `mqsichangeproperties` command and a policy file
- Create an MQEndpoint policy in the Integration web user interface



WebSphere Education

Creating an MQEndpoint policy

- Use the IBM Integration Toolkit to generate a policy document from an existing message flow node
  - Operational property values can be edited as required
  - Generated policy document can be saved in the Integration Registry
- Use the IBM Integration web user interface to create, retrieve, update, and delete policy documents that are stored in the Integration Registry
- Use the following commands to create, retrieve, update, and delete policies by using an MQEndpoint policy document:
  - To create a policy, use the **`mqsicreatepolicy`** command
  - To retrieve details of a policy, use the **`mqsireportpolicy`** command
  - To update a policy, use the **`mqsichangepolicy`** command
  - To delete a policy, use the **`mqsideletepolicy`** command

© Copyright IBM Corporation 2015

Figure 3-25. Creating an MQEndpoint policy WM6461.0

Notes:

This figure lists the options for creating an MQEndpoint policy.

A developer can create an MQEndpoint policy by using the Integration Toolkit.

An administrator can create an MQEndpoint policy by using the Integration web user interface.

You can also create an MQEndpoint policy by using the **`mqsicreatepolicy`** and specifying an MQEndpoint policy document. For example, to create an MQEndoint policy that is named `mqep_policy` from the file `my_policy.xml` for integration node IBNODE, type:

```
mqsicreatepolicy IBNODE -t MQEndpoint -l mqep_policy -f my_policy.xml
```

WebSphere Education IBM

Creating an MQEndpoint policy in the IBM Integration web user interface

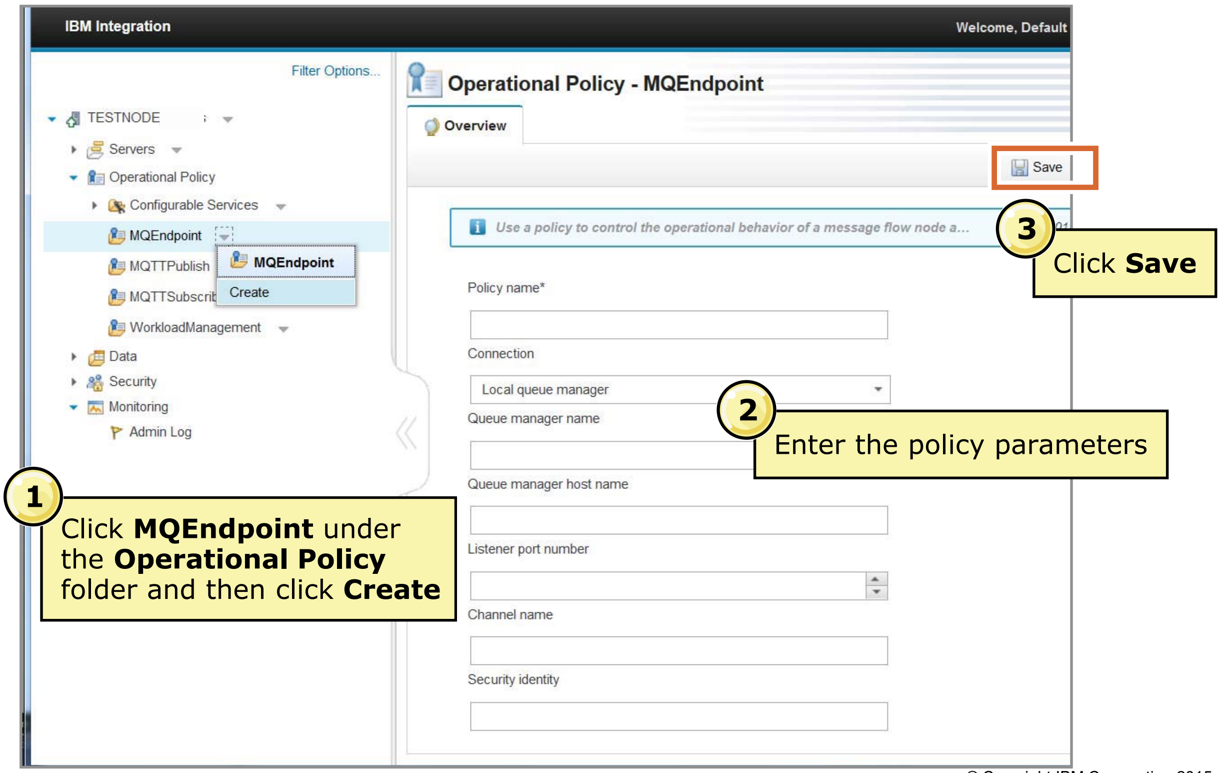


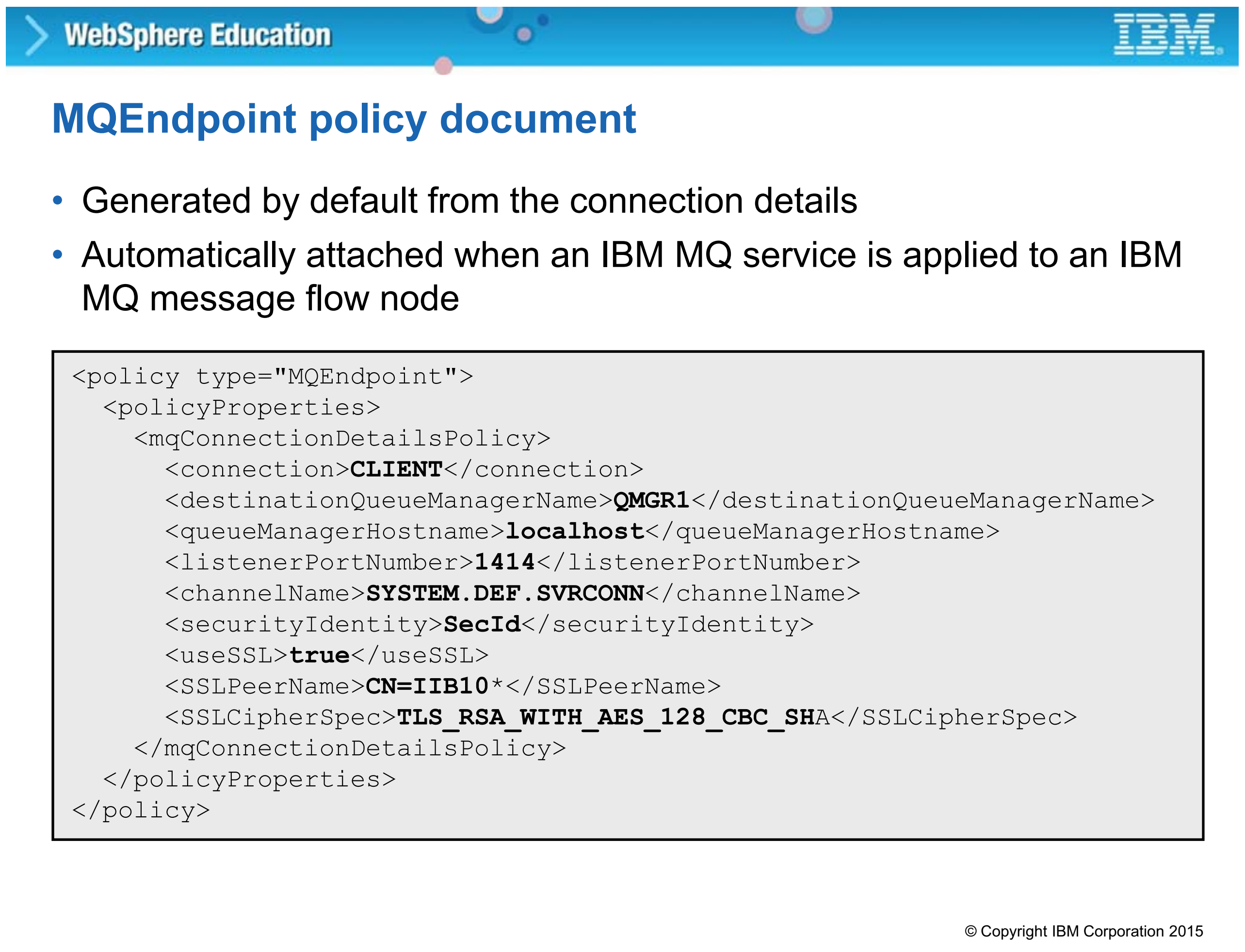
Figure 3-26. Creating an MQEndpoint policy in the IBM Integration web user interface WM6461.0

Notes:

You can use the Integration web interface to create, retrieve, update, and delete policy documents that are stored in the Integration Registry.

You can also use the message flow view to retrieve and update a policy that is attached to a message flow node.

1. In the navigation tree, expand **Servers > *server_name* > *resource***, where *server_name* is the name of the integration server, and *resource* is where you stored your message flow.
2. Expand **Message Flows**, and select the name of the message flow, or subflow, you want to view.
3. Click the **Operational Policy** tab from the top of the message flow pane, and the message flow, or subflow, is displayed in the **Node Policies** section.
4. If the message flow, or subflow, includes a node that has an operational policy that is attached, the policy icon is displayed on the upper-right corner of the node. Click the policy icon to retrieve and update the policy document.



WebSphere Education

MQEndpoint policy document

- Generated by default from the connection details
- Automatically attached when an IBM MQ service is applied to an IBM MQ message flow node

```
<policy type="MQEndpoint">
  <policyProperties>
    <mqConnectionDetailsPolicy>
      <connection>CLIENT</connection>
      <destinationQueueManagerName>QMGR1</destinationQueueManagerName>
      <queueManagerHostname>localhost</queueManagerHostname>
      <listenerPortNumber>1414</listenerPortNumber>
      <channelName>SYSTEM.DEF.SVRCONN</channelName>
      <securityIdentity>SecId</securityIdentity>
      <useSSL>true</useSSL>
      <SSLPeerName>CN=IIB10*</SSLPeerName>
      <SSLCipherSpec>TLS_RSA_WITH_AES_128_CBC_SHA</SSLCipherSpec>
    </mqConnectionDetailsPolicy>
  </policyProperties>
</policy>
```

Figure 3-27. MQEndpoint policy document WM6461.0

Notes:

In the Integration Toolkit, a developer can create an IBM MQ service to discover artifacts from IBM MQ queue managers. You can use an MQ Service connection detail to generate an MQEndpoint policy document.

WebSphere Education IBM

Integration Registry

- Stores service definitions and operational policies in an integration node
  - Facilitates collaboration and reuse
- Populated by:
  - Publishing MQEndpoint, MQTTPublish, MQTTSubscribe, and workload management policies from the Integration Toolkit or the Integration web user interface
  - Publishing discovered IBM MQ services from the Integration Toolkit
- Can be accessed in the Integration Toolkit and the Integration web user interface

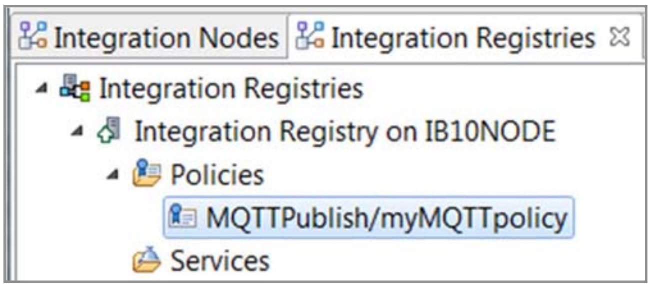


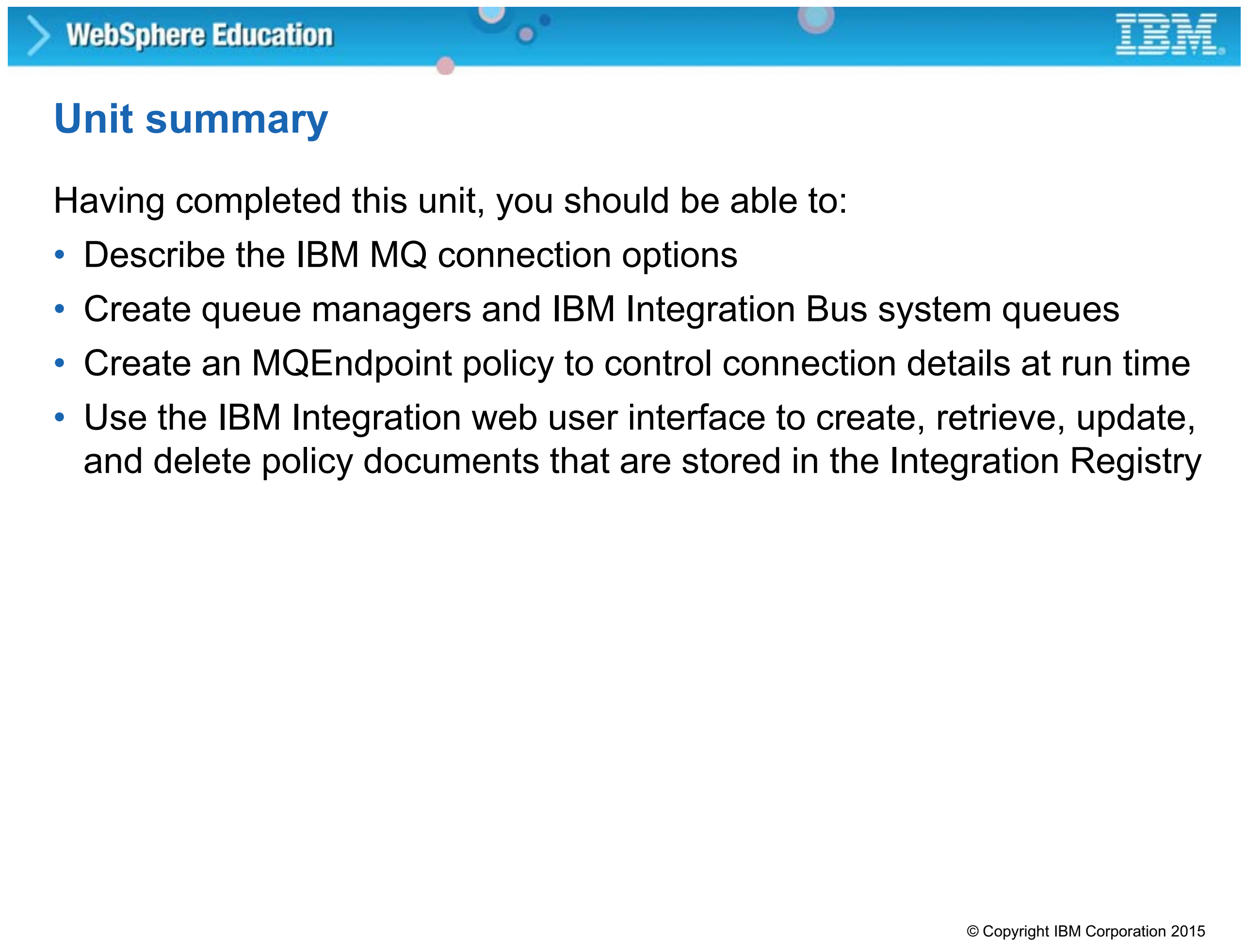
Figure 3-28. Integration Registry WM6461.0

Notes:

The Integration Registry is an integration node file repository for policies and services.

The Integration Registry is hosted inside a runtime integration node. Every runtime integration node can host one (and only one) Integration Registry.

All integration nodes have an Integration Registry available by default.



WebSphere Education
IBM

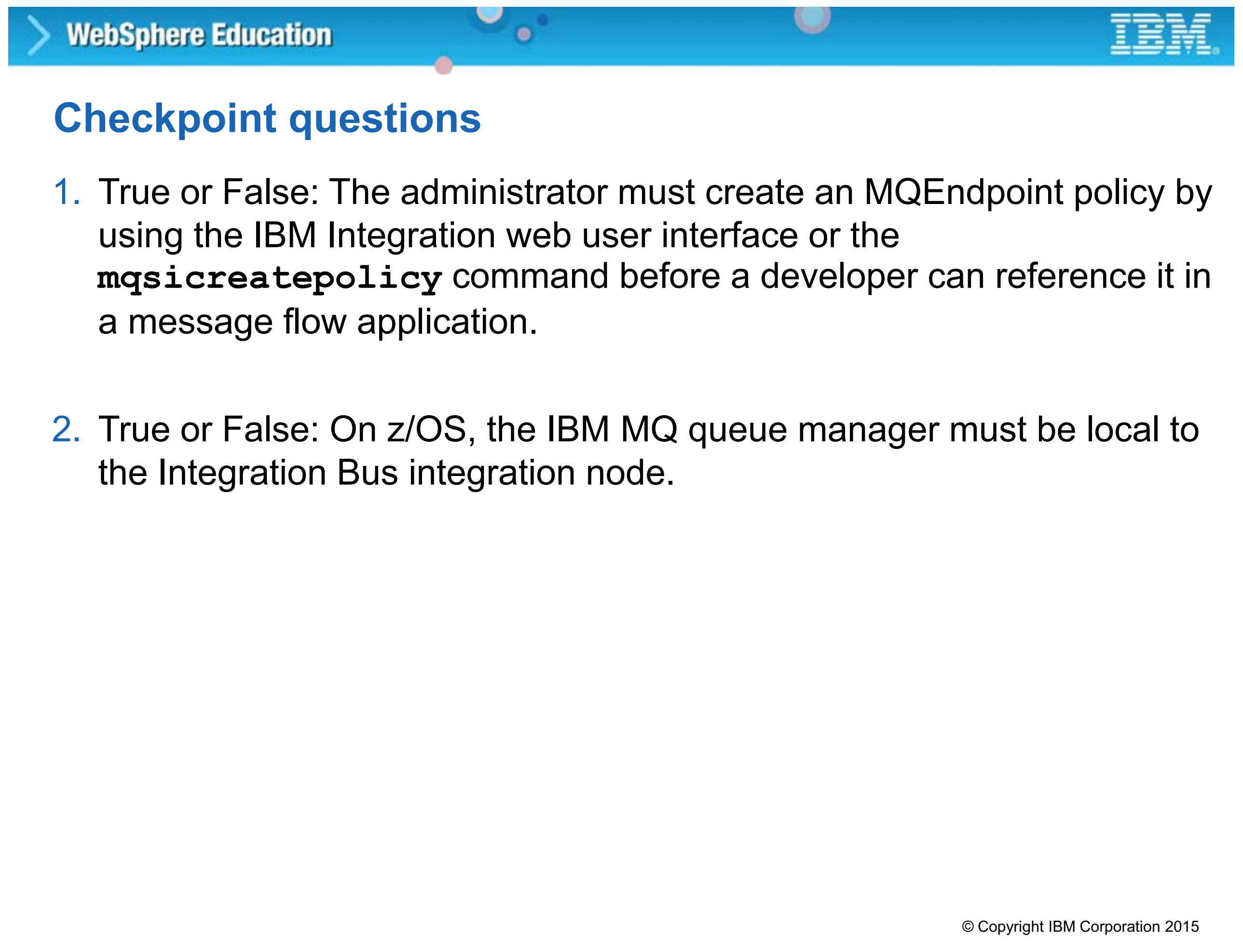
Unit summary

Having completed this unit, you should be able to:

- Describe the IBM MQ connection options
- Create queue managers and IBM Integration Bus system queues
- Create an MQEndpoint policy to control connection details at run time
- Use the IBM Integration web user interface to create, retrieve, update, and delete policy documents that are stored in the Integration Registry

Figure 3-29. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. True or False: The administrator must create an MQEndpoint policy by using the IBM Integration web user interface or the **`mqsicreatepolicy`** command before a developer can reference it in a message flow application.

2. True or False: On z/OS, the IBM MQ queue manager must be local to the Integration Bus integration node.

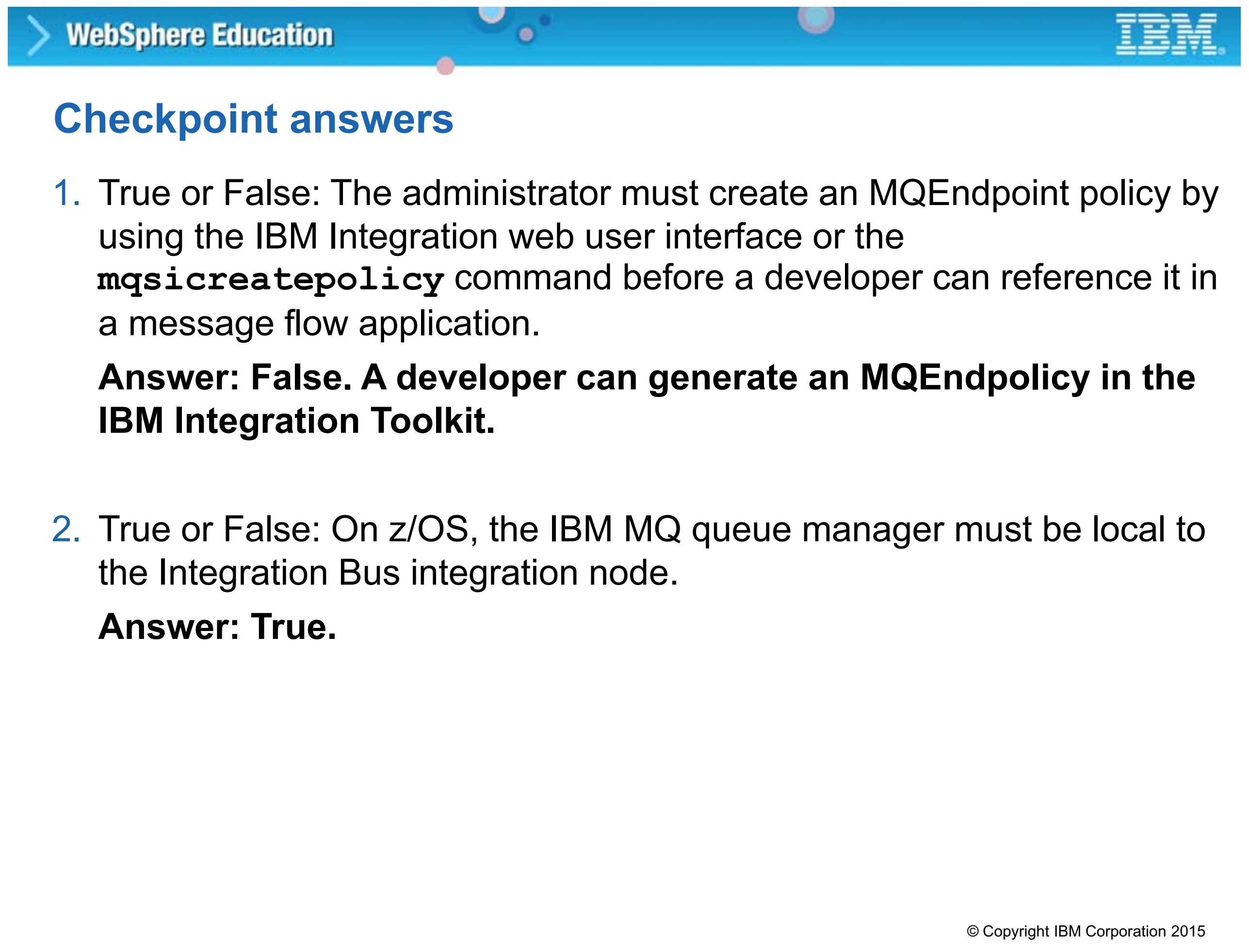
Figure 3-30. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education IBM

Checkpoint answers

1. True or False: The administrator must create an MQEndpoint policy by using the IBM Integration web user interface or the **`mqsicreatepolicy`** command before a developer can reference it in a message flow application.

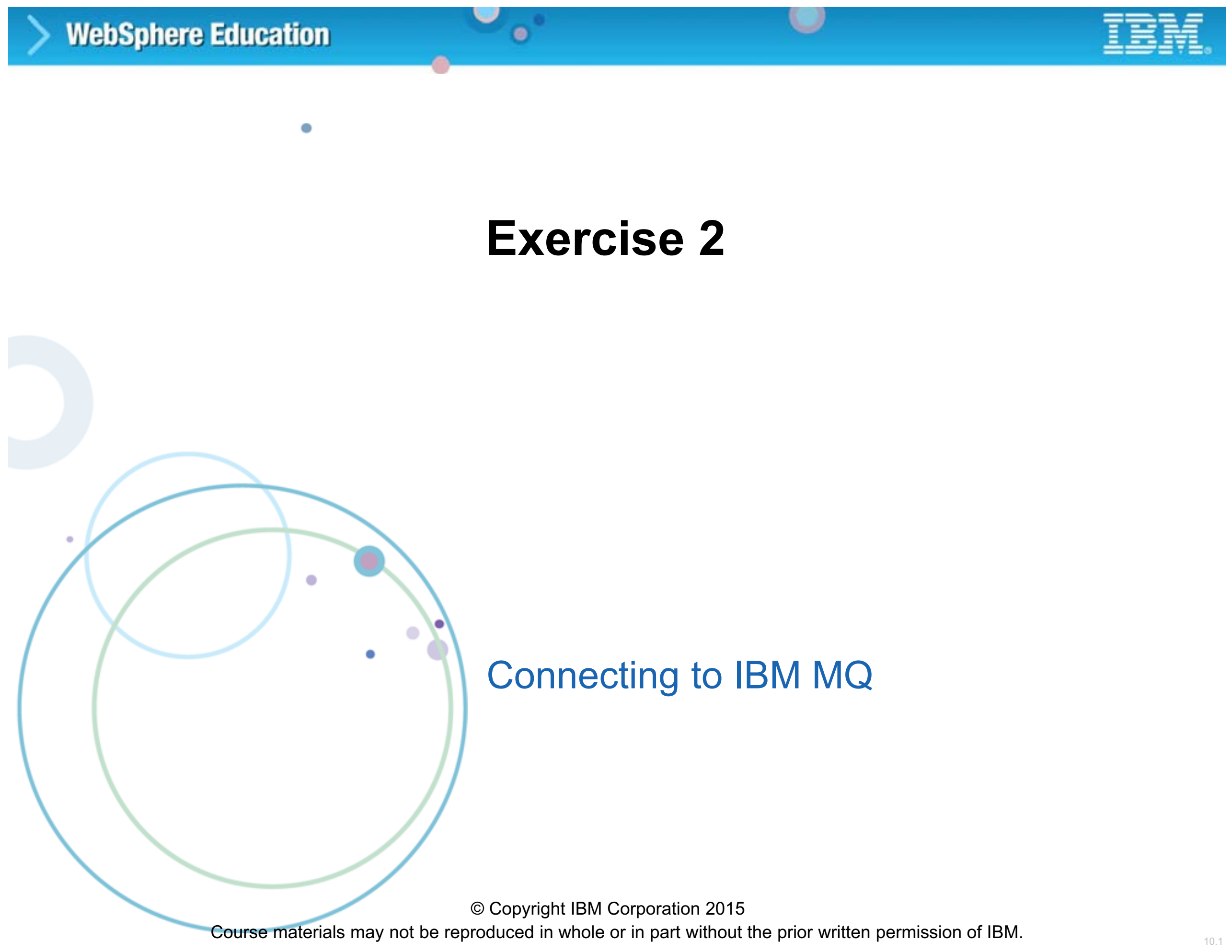
   Answer: False. A developer can generate an MQEndpolicy in the IBM Integration Toolkit.

2. True or False: On z/OS, the IBM MQ queue manager must be local to the Integration Bus integration node.

   Answer: True.

Figure 3-31. Checkpoint answers WM6461.0

Notes:



WebSphere Education

Exercise 2

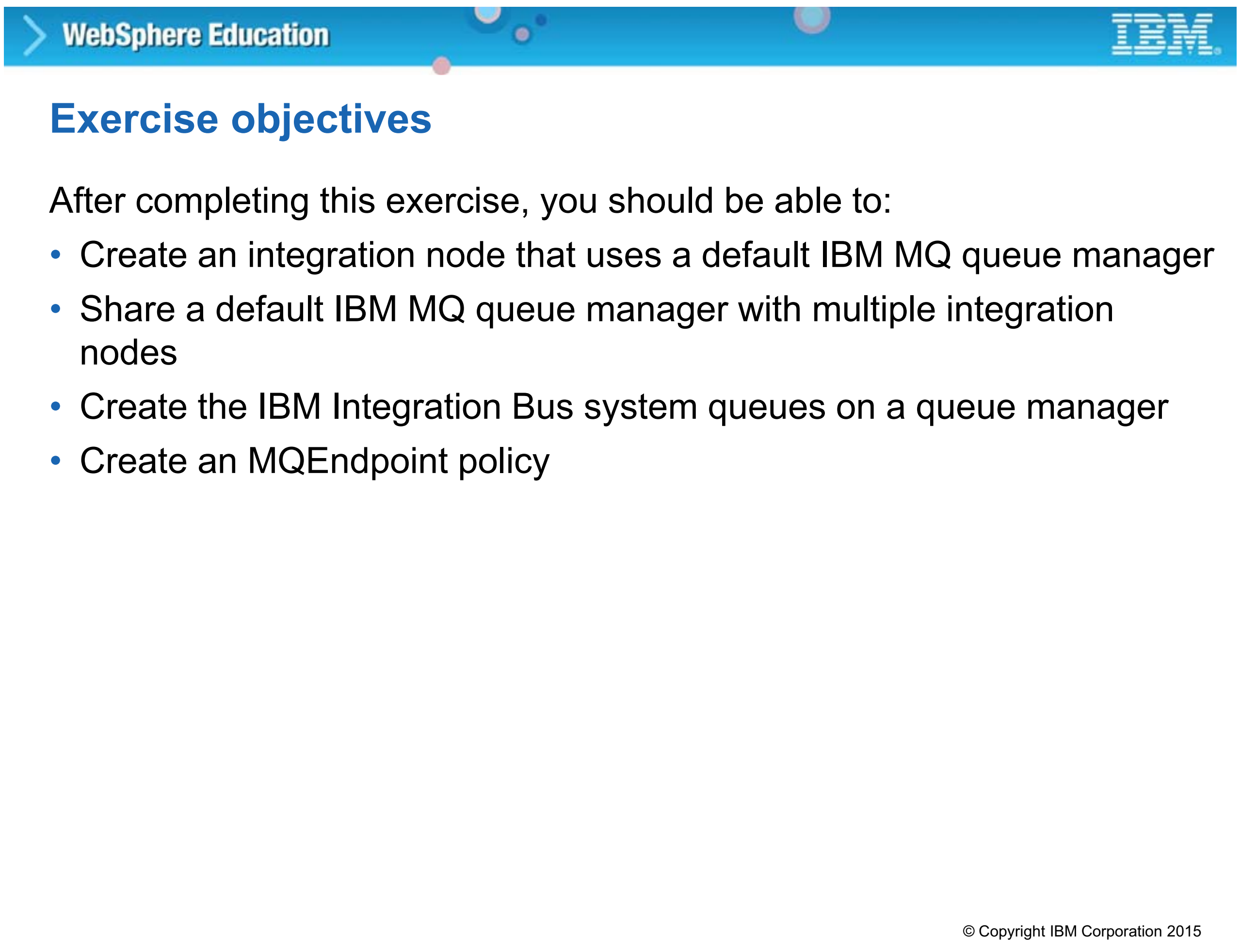
Connecting to IBM MQ

10.1

Figure 3-32. Exercise 2 WM6461.0

Notes:

In this exercise, you create an IBM MQ queue manager and an IBM Integration Bus integration node that uses the queue manager as a default queue manager. You create a second integration node with the same default queue manager as the first integration node. Finally, you create the IBM Integration Bus system queues on the queue manager and an MQEndpoint policy.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Create an integration node that uses a default IBM MQ queue manager
- Share a default IBM MQ queue manager with multiple integration nodes
- Create the IBM Integration Bus system queues on a queue manager
- Create an MQEndpoint policy

© Copyright IBM Corporation 2015

Figure 3-33. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 4. Administration in the IBM Integration Toolkit

What this unit is about

The IBM Integration Toolkit is the IBM Integration Bus development environment. This unit provides an overview of the administrative tasks available to application developers in the IBM Integration Toolkit and provides an overview of the development process.

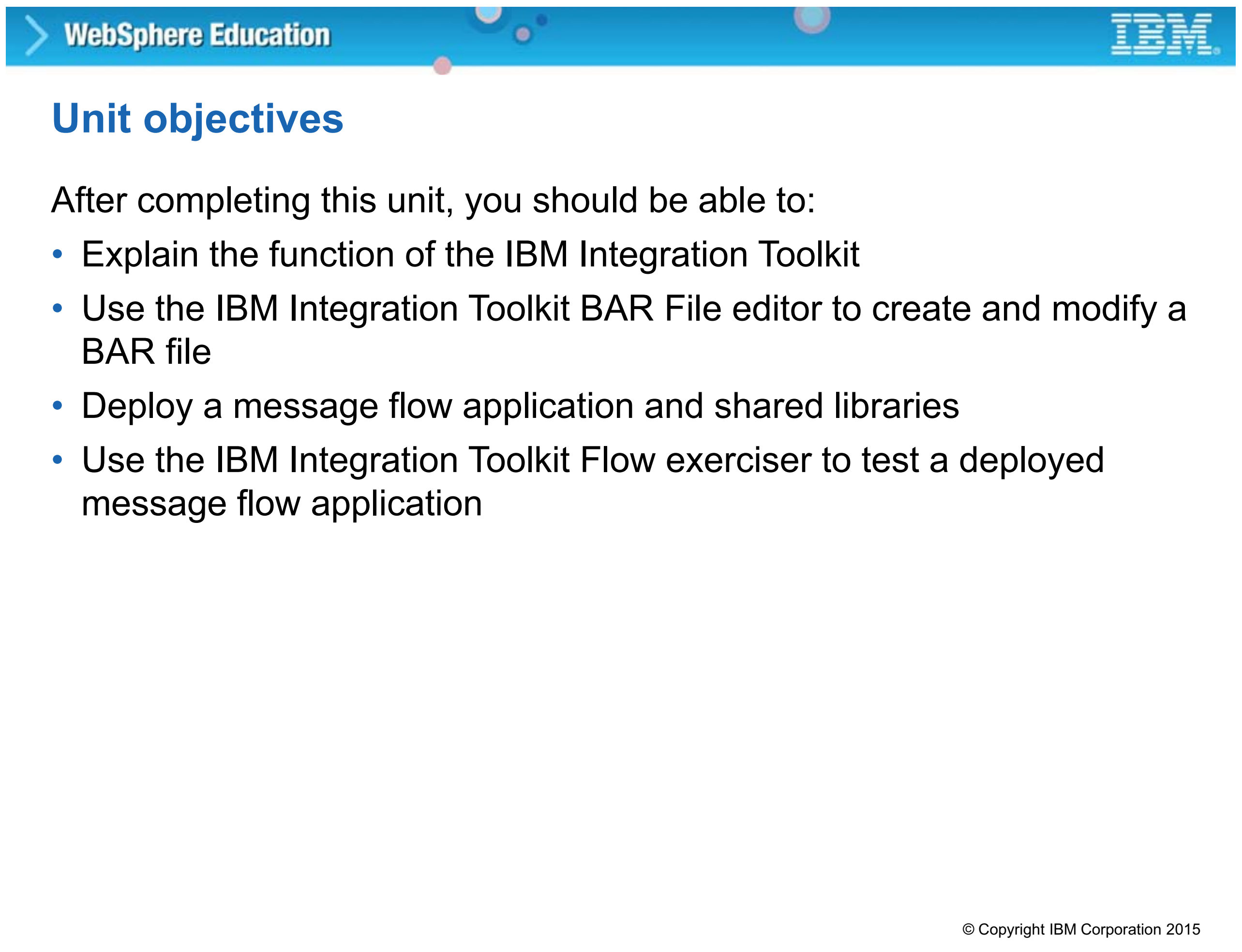
What you should be able to do

After completing this unit, you should be able to:

- Explain the function of the IBM Integration Toolkit
- Use the IBM Integration Toolkit BAR File editor to create and modify a BAR file
- Deploy a message flow application and shared libraries
- Use the IBM Integration Toolkit Flow exerciser to test a deployed message flow application

How you will check your progress

- Checkpoints
- Lab exercise: Using the IBM Integration Toolkit



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Explain the function of the IBM Integration Toolkit
- Use the IBM Integration Toolkit BAR File editor to create and modify a BAR file
- Deploy a message flow application and shared libraries
- Use the IBM Integration Toolkit Flow exerciser to test a deployed message flow application

© Copyright IBM Corporation 2015

Figure 4-1. Unit objectives WM6461.0

Notes:

The IBM Integration Toolkit, sometimes referred to as the “workbench”, is an integrated development and administrative environment.

Developers can create, deploy, test, and debug message flows and message models by using the IBM Integration Toolkit.

This unit explores the functions of the IBM Integration Toolkit. If you require in-depth training on developing and testing message flows, see courses WM666, *IBM Integration Bus V10.0 Application Development I*, and WM676, *IBM Integration Bus V10.0 Application Development II*.

4.1. IBM Integration Toolkit concepts

Development and runtime environment

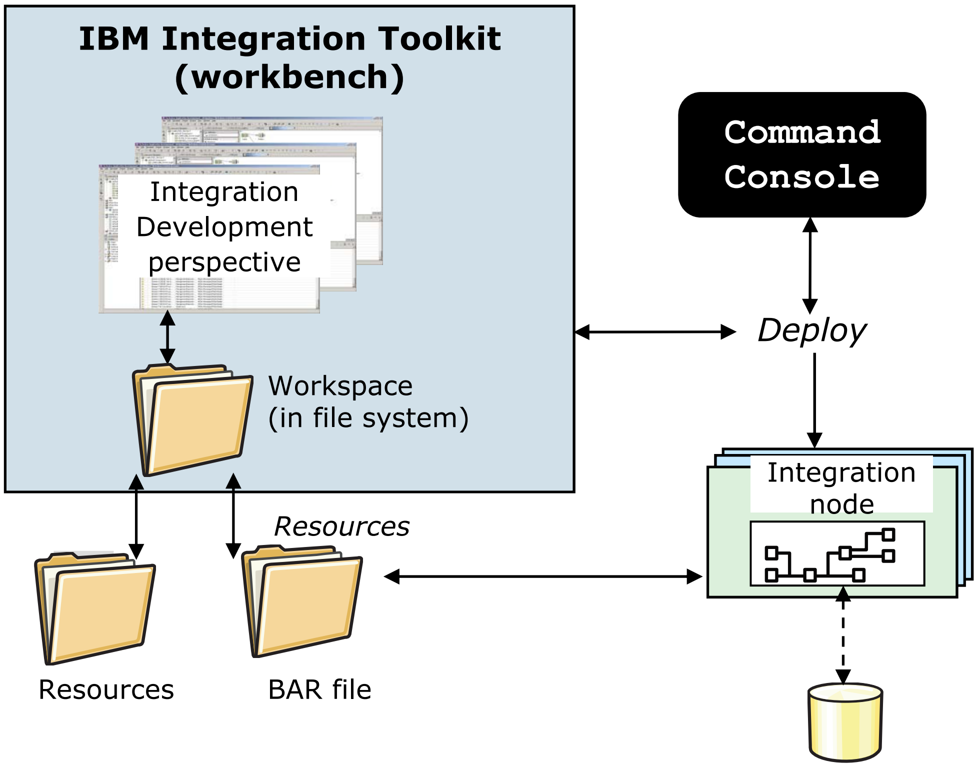

IBM Integration Toolkit (workbench)
Integration Development perspective
Workspace (in file system)
Command Console
Deploy
Integration node
Resources
Resources
BAR file

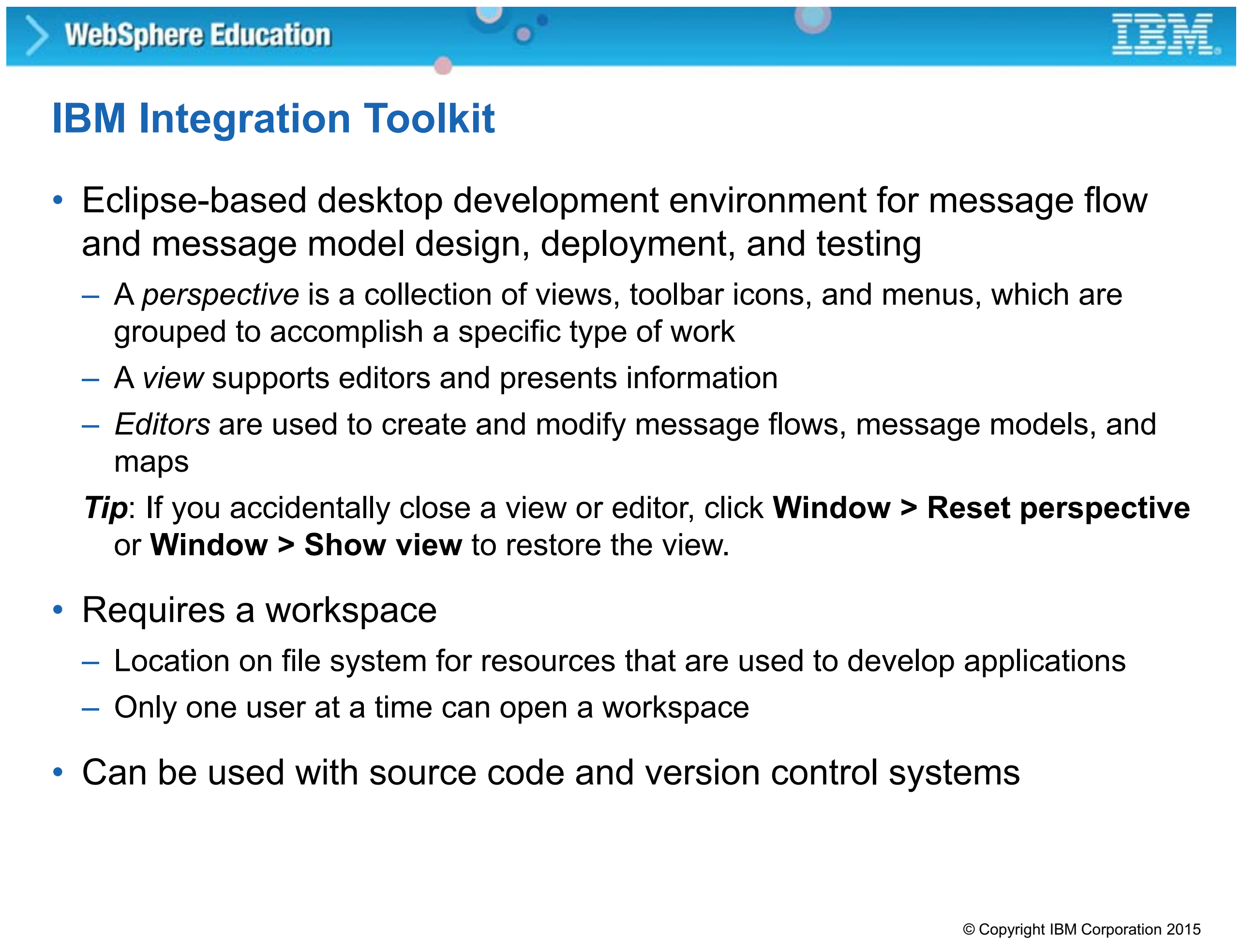
Figure 4-2. Development and runtime environment WM6461.0

Notes:

The IBM Integration Toolkit consists of a workbench window, which displays one or more perspectives. A *perspective* is a group of views and editors that are required to complete tasks that are associated with a role.

The applications, libraries, projects, folders, and files that exist in the workbench are called *resources*. Resources are stored collectively in a *workspace*. The workspace is a private work area for the individual developer. It holds the environment metadata and the loaded resource data in regular files. Every time that you save a resource, the changes are reflected in the file system. Normally you use your local file system, but you can choose to store your workspace in a network file system (NFS) as well.

To run any developed resource, it must be packaged into a *broker archive (BAR) file*. You then must deploy the BAR file to the integration server on an integration node.



WebSphere Education
IBM

IBM Integration Toolkit

- Eclipse-based desktop development environment for message flow and message model design, deployment, and testing
  - A *perspective* is a collection of views, toolbar icons, and menus, which are grouped to accomplish a specific type of work
  - A *view* supports editors and presents information
  - *Editors* are used to create and modify message flows, message models, and maps

  Tip: If you accidentally close a view or editor, click **Window > Reset perspective** or **Window > Show view** to restore the view.
- Requires a workspace
  - Location on file system for resources that are used to develop applications
  - Only one user at a time can open a workspace
- Can be used with source code and version control systems

Figure 4-3. IBM Integration Toolkit WM6461.0

Notes:

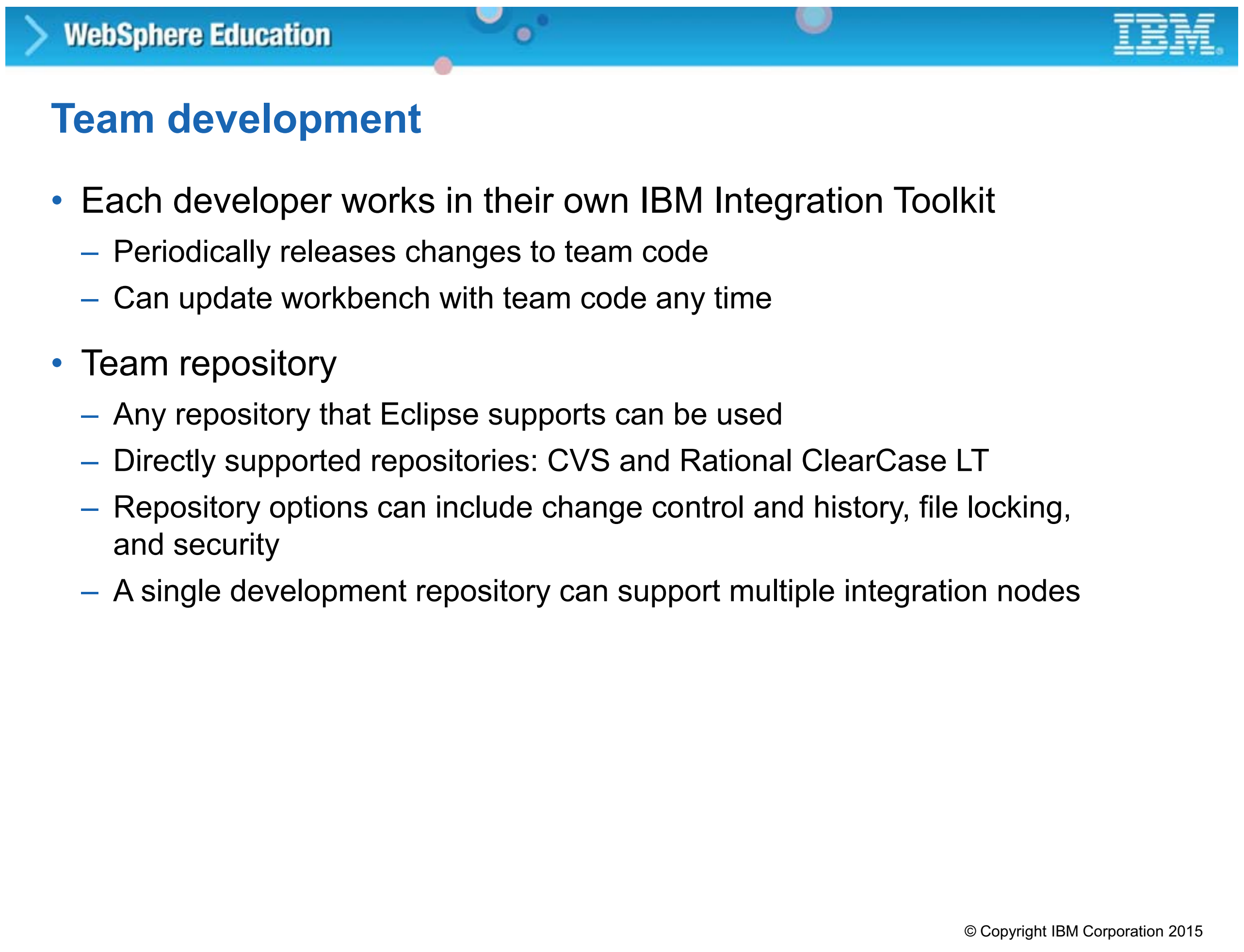
The IBM Integration Toolkit is an Eclipse-based integrated development environment for developing, deploying, and testing message flow applications and integration services.

The Eclipse environment has perspectives, views, and editors. A perspective controls initial view visibility, layout, and action visibility. The IBM Integration Toolkit provides standard perspectives for general resource navigation, online help, and team support tasks. The default perspective when you start the Integration Toolkit is the Integration Development perspective. More perspectives are supplied by other plug-ins.

Each perspective has its own views and editors that are arranged for presentation on the screen. Several different types of views and editors can be open at the same time within a perspective.

Views provide information about the currently selected object. For example, if you select a message processing node in the Message Flow editor, the properties of the node are shown in another view. Views have a simpler lifecycle than editors: modifications that are made in a view (such as changing a property value) are generally saved immediately, and the changes are reflected immediately in other related parts of the user interface.

Editors allow the user to open, edit, and save objects much like file-system-based tools. When active, an editor can contribute actions to the workbench menus and toolbar. The Integration Toolkit provides a standard message flow and message set development Other editors are supplied by plug-ins or applications.



WebSphere Education

Team development

- Each developer works in their own IBM Integration Toolkit
  - Periodically releases changes to team code
  - Can update workbench with team code any time
- Team repository
  - Any repository that Eclipse supports can be used
  - Directly supported repositories: CVS and Rational ClearCase LT
  - Repository options can include change control and history, file locking, and security
  - A single development repository can support multiple integration nodes

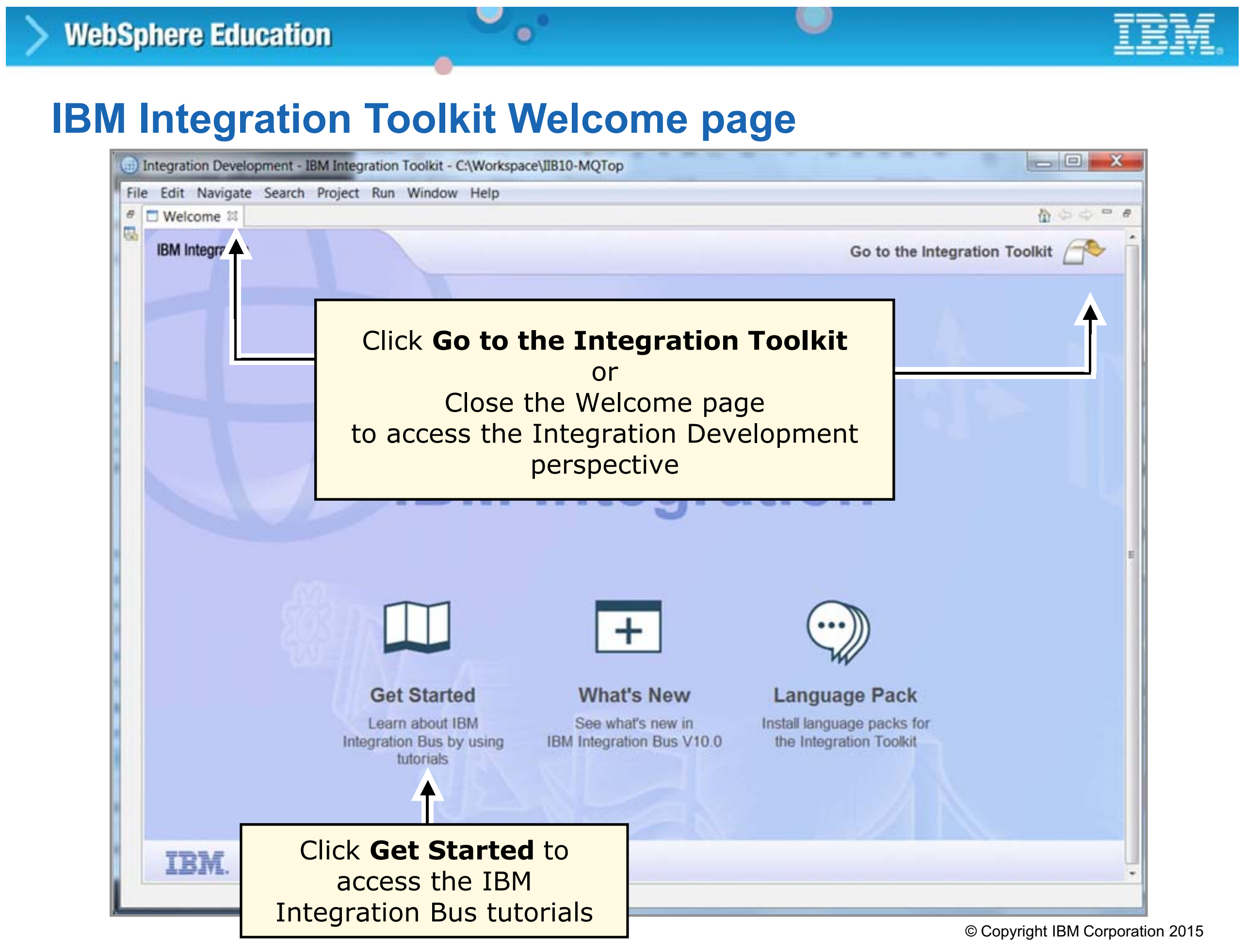
© Copyright IBM Corporation 2015

Figure 4-4. Team development WM6461.0

Notes:

With Integration Bus, each developer works in their own Toolkit instance. But the artifacts that the Toolkit creates are maintained as files so a source code management system or team repository can be used to manage the artifacts.

A source code management system or team repository is not part of the IBM Integration Toolkit, but it directly supports any Eclipse repository. For example, Concurrent Version System (CVS) is a simple open source software configuration management (SCM) system. It implements version control, parallel development, and lifecycle integration. Individual developers can use CVS, as can large, distributed teams.



IBM Integration Toolkit Welcome page

WebSphere Education
Integration Development - IBM Integration Toolkit - C:\Workspace\IIB10-MQTop
File Edit Navigate Search Project Run Window Help
Welcome
Go to the Integration Toolkit
Get Started
Learn about IBM Integration Bus by using tutorials
What's New
See what's new in IBM Integration Bus V10.0
Language Pack
Install language packs for the Integration Toolkit

Click **Go to the Integration Toolkit**
or
Close the Welcome page
to access the Integration Development perspective

Click **Get Started** to access the IBM Integration Bus tutorials

© Copyright IBM Corporation 2015

Figure 4-5. IBM Integration Toolkit Welcome page WM6461.0

Notes:

The Integration Toolkit is started in a Windows environment by clicking **IBM Integration Toolkit 10.0.0.0 > IBM Integration Toolkit 10.0.0.0** from the **Programs** menu.

The Welcome page, which is shown in the figure, is displayed the first time that the Integration Toolkit starts. You can also access the Welcome page at any time by clicking **Welcome** from the **Help** menu.

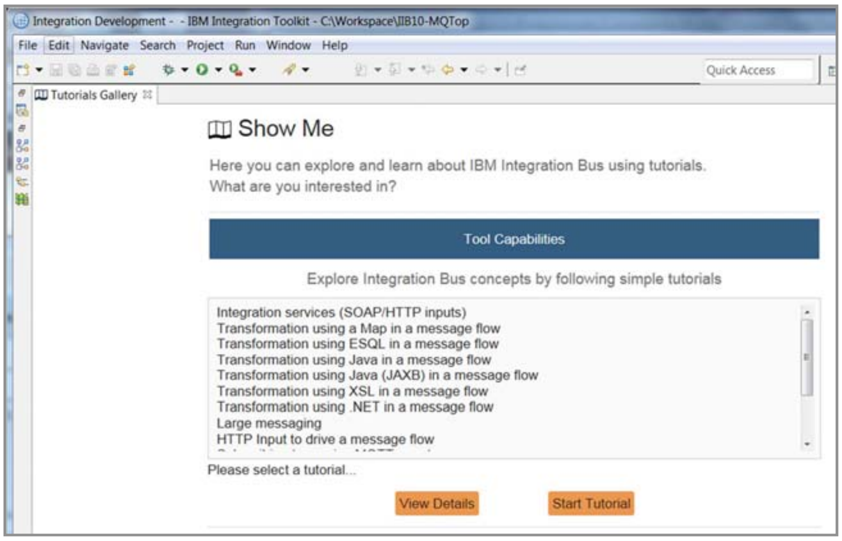
The Welcome topics are **Get Started**, **What's New**, and **Language Pack**. The **Get Started** topic includes links to tutorials for message flow development and testing.

Close the Welcome page to access the Integration Development perspective and begin message flow, integration services, and message model development.

WebSphere Education

IBM Integration Bus Tutorials Gallery

- Introduce the features that are available in Integration Bus and how to use them
  - End-to-end Integration Bus message flow applications
  - Demonstrate a specific feature of IBM Integration Bus such as file processing, message transformation, and modeling
- Access from the Welcome page or by clicking **Help > Tutorials Galley**


Integration Development - - IBM Integration Toolkit - C:\Workspace\IIB10-MQTop
File Edit Navigate Search Project Run Window Help
Quick Access
Tutorials Gallery
Tool Capabilities
View Details
Start Tutorial

Show Me

Here you can explore and learn about IBM Integration Bus using tutorials.
What are you interested in?

Explore Integration Bus concepts by following simple tutorials

Integration services (SOAP/HTTP inputs)
Transformation using a Map in a message flow
Transformation using ESQL in a message flow
Transformation using Java in a message flow
Transformation using Java (JAXB) in a message flow
Transformation using XSL in a message flow
Transformation using .NET in a message flow
Large messaging
HTTP Input to drive a message flow

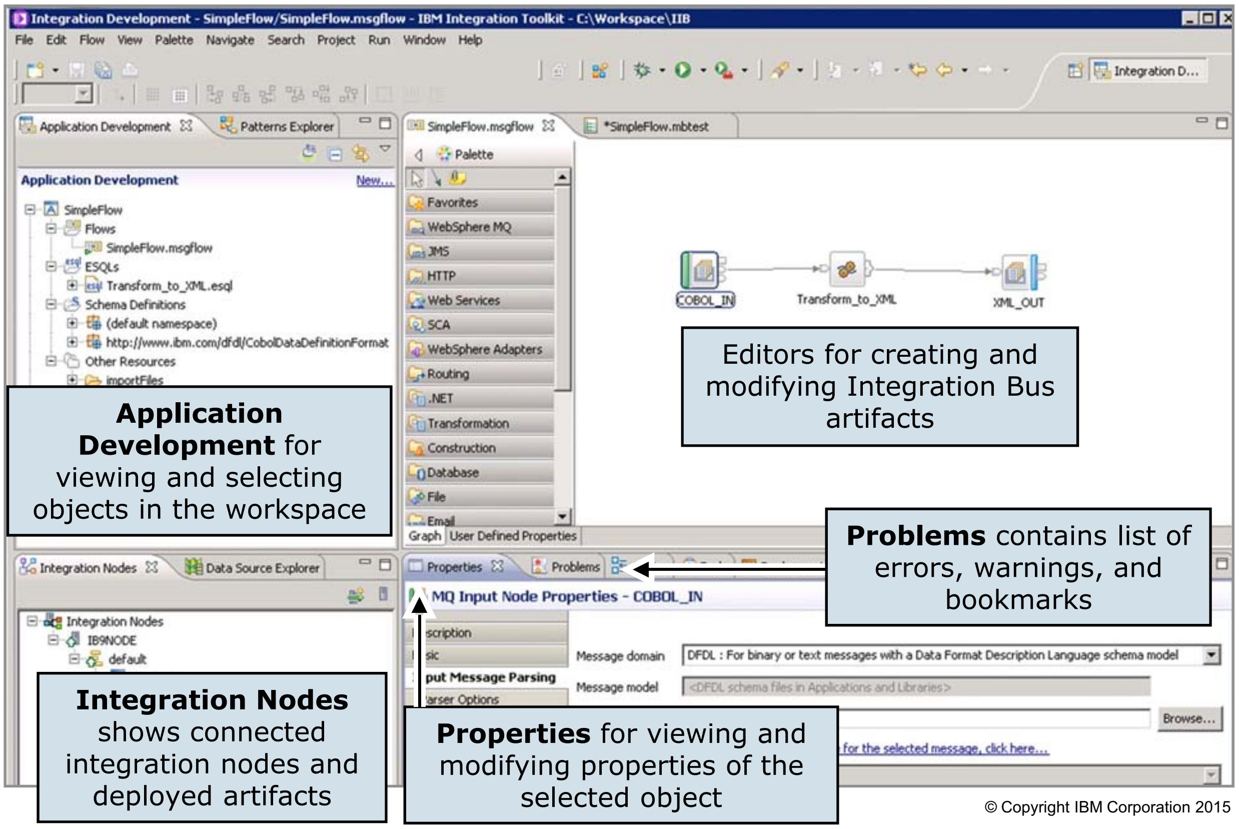
Please select a tutorial...

Figure 4-6. IBM Integration Bus Tutorials Gallery WM6461.0

Notes:

The Integration Toolkit provides many tutorial applications in the **Tutorials Gallery**. Unlike a message flow pattern, which acts as a configurable template that you can customize to your needs, a tutorial is a complete working example of an Integration Bus solution. You can deploy it and see how the example runs in the Integration Bus environment.

Integration Development views



Integration Development - SimpleFlow/SimpleFlow.msgflow - IBM Integration Toolkit - C:\Workspace\IIB
Application Development
Patterns Explorer
SimpleFlow.msgflow
*SimpleFlow.mbtest
Palette
SimpleFlow
Flows
ESQLs
Transform_to_XML.esql
Schema Definitions
Other Resources
Favorites
WebSphere MQ
JMS
HTTP
Web Services
SCA
WebSphere Adapters
Routing
.NET
Transformation
Construction
Database
File
Email
COBOL_IN
Transform_to_XML
XML_OUT
Integration Nodes
Data Source Explorer
IB9NODE
default
Properties
Problems
MQ Input Node Properties - COBOL_IN
Message domain
DFDL : For binary or text messages with a Data Format Description Language schema model
Message model
Browse...
Editors for creating and modifying Integration Bus artifacts
Application Development for viewing and selecting objects in the workspace
Problems contains list of errors, warnings, and bookmarks
Integration Nodes shows connected integration nodes and deployed artifacts
Properties for viewing and modifying properties of the selected object
© Copyright IBM Corporation 2015

Figure 4-7. Integration Development views WM6461.0

Notes:

The figure shows some of the most common views that are available in the Integration Development perspective:

- The Application Development view is used to display and select resources in a workspace. Right-clicking most objects in the Navigator opens a menu. Double-clicking most objects opens the object in an editor.
- Object editors open in the Editors view on separate tabs. The example in the figure shows the Message Flow editor. The Message Flow editor contains the message flow nodes in a palette and an area for developing message flows and wiring message nodes together.

Many tabbed views are provided in the lower right of the Integration Development perspective:

- The **Outline** view provides an outline of currently edited objects.
- The **Properties** view is used to view and modify the configuration properties for the currently selected object.
- The **Problems** view contains the list of current errors or warnings.

WebSphere Education IBM

Integration Toolkit containers

- **Applications** contains all the resources that are required to create a complete solution
  - Can include message flows, message models, JAR files, and XML stylesheets
  - **Integration service** is a specialized application with a defined interface and structure that acts as a container for a web services solution
  - **REST API** is an application that implements a RESTful interface that is defined by importing a Swagger 2.0 document
- **Library** is a logical grouping of related reusable resources (message models, code, maps, and subflows) that an application uses
- **Integration project**
  - Container in which you create and maintain all the resources that are associated with one or more message flows
  - Shown in the **Independent Resources** folder in the **Application Development** view

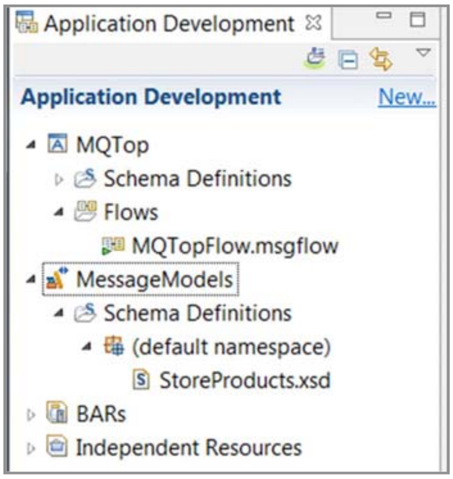


Figure 4-8. Integration Toolkit containers WM6461.0

Notes:

Integration Bus provides several types of container projects that a developer can use to organize message flow development resources.

An *application* is a container for the resources that are required to create a solution, and that should be deployed and managed together. Applications provide encapsulation, and resources within an application are packaged and deployed as a single unit, and can be managed as a single unit.

An *integration service* is a special type of application that has a defined interface and implementation. Services provide assisted development to provide web services in an integration node. Like applications, services provide encapsulation and are deployed and managed as a single unit. For more information, see Integration services.

In Integration Bus, a *REST API* is a specialized application that can be used to expose integrations as a RESTful web service that HTTP clients can call.

A *library* is a logical grouping of reusable routines and resources such as subflows, ESQL modules, message definitions, maps, and Java utilities. A library is useful to group resources for reuse or ease of management. A library is also packaged and deployed as a single unit.

WebSphere Education

Exporting to create a Project Interchange file

- Create a Project Interchange file to export solution resources from the workspace
- **File > Export > IBM Integration > Project Interchange** copies resources from the workspace to:
  - The file system
  - A compressed file
  - Any other file container that Eclipse supports
- Eclipse operation
  - No transformation
  - No IBM Integration Bus involvement

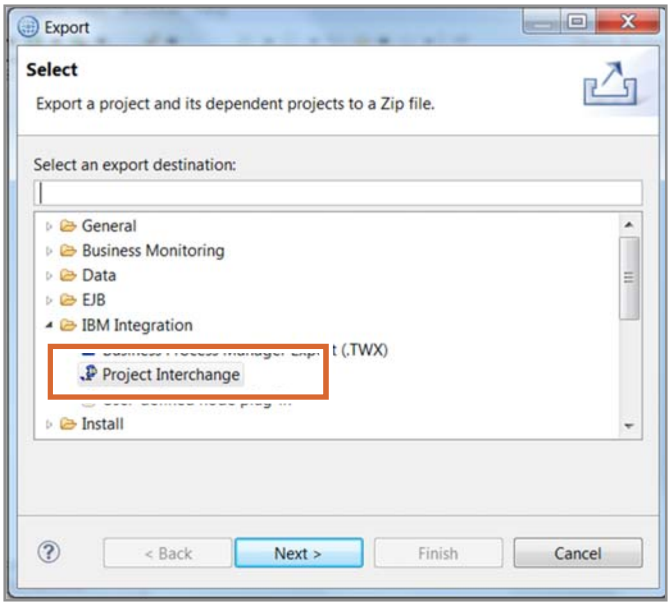


Figure 4-9. Exporting to create a Project Interchange file WM6461.0

Notes:

Integration Bus projects can be imported or exported. Exporting is the best choice for transferring files from a workspace to ensure that all components are included and categorized correctly.

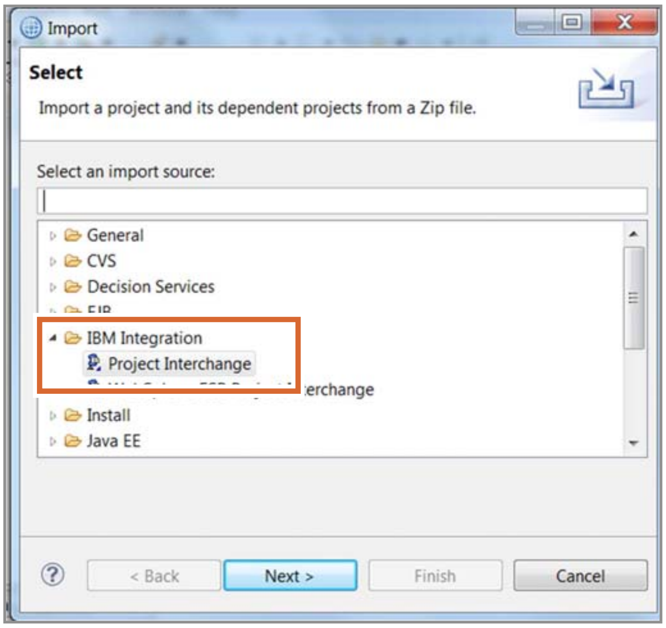
To export a project, application, or library to a Project Interchange file:

1. Click **File > Export**.
2. In the **Export** wizard, expand the **Other** folder as shown in the figure.
3. Select **Project Interchange**, and then click **Next**.
4. Select the objects to include in the export file, enter the name and location of the project interchange file, and then click **Finish**. The project interchange file is created as a compressed file.

WebSphere Education

Importing a Project Interchange file

- **File > Import > IBM Integration > Project Interchange** copies solution resources from a Project Interchange file into the existing workspace
- Can import Project Interchange files from previous versions of WebSphere Message Broker or IBM Integration Bus


Import

Select

Import a project and its dependent projects from a Zip file.

Select an import source:

Figure 4-10. Importing a Project Interchange file WM6461.0

Notes:

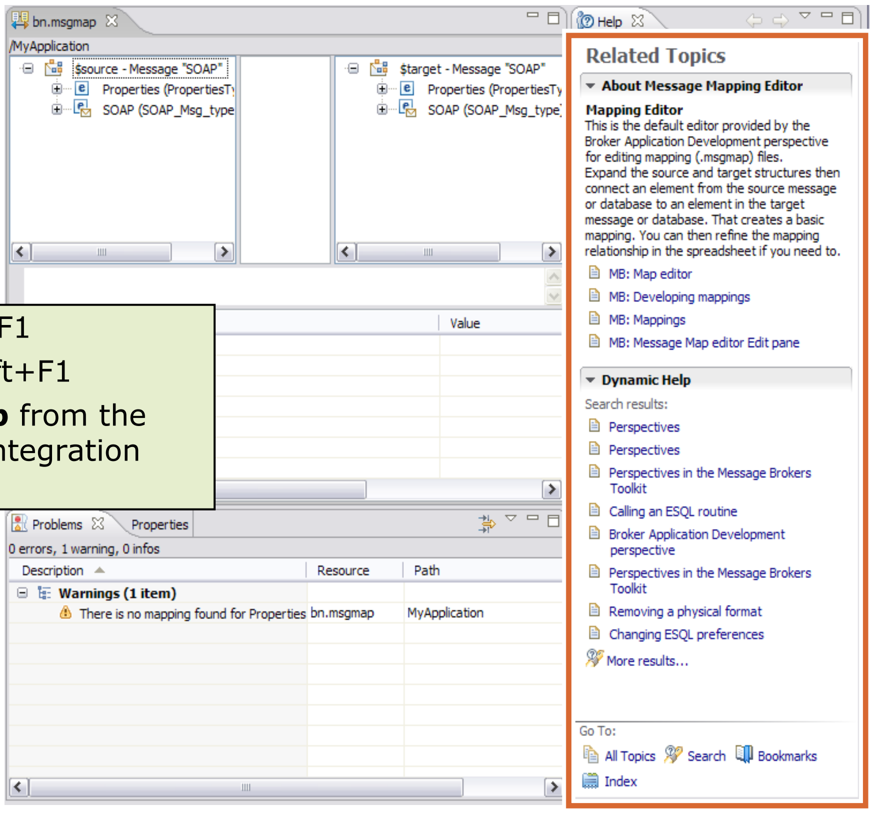
You must use the Integration Toolkit import and export functions to move application, library, or project resources to ensure that all references correctly reflect the new organization.

As shown in the figure, there are a many import options. For most of the importing that you do in this course, you use Project Interchange files to import exercise objects into a workspace.

WebSphere Education

Integrated help

- On Windows, press F1
- On Linux, press Shift+F1
- Click **Dynamic Help** from the **Help** menu in the Integration Toolkit



0 errors, 1 warning, 0 infos

There is no mapping found for Properties | bn.msgmap | MyApplication

Mapping Editor
This is the default editor provided by the Broker Application Development perspective for editing mapping (.msgmap) files.
Expand the source and target structures then connect an element from the source message or database to an element in the target message or database. That creates a basic mapping. You can then refine the mapping relationship in the spreadsheet if you need to.

- MB: Map editor
- MB: Developing mappings
- MB: Mappings
- MB: Message Map editor Edit pane

- Perspectives
- Perspectives
- Perspectives in the Message Brokers Toolkit
- Calling an ESQL routine
- Broker Application Development perspective
- Perspectives in the Message Brokers Toolkit
- Removing a physical format
- Changing ESQL preferences
- More results...

© Copyright IBM Corporation 2015

Figure 4-11. Integrated help WM6461.0

Notes:

A **Help** view is provided in the Integration Toolkit. To open the **Help** view in the Integration Toolkit, press F1 on Windows, or Shift+F1 on Linux, or click **Dynamic Help** from the **Help** menu.

The default view for **Help** contains **Related Topics**. In the top section, there is a small description of the interface element, such as a view, or an object, such as a node, that you last selected with the mouse. This description also contains links to topics in the IBM Knowledge Center. Click a link to open any topic in the Help view. Right-click the topic link and click **Open in Help Contents** to open the topic in the IBM Knowledge Center. The topic ***Help*** opens in a separate window.

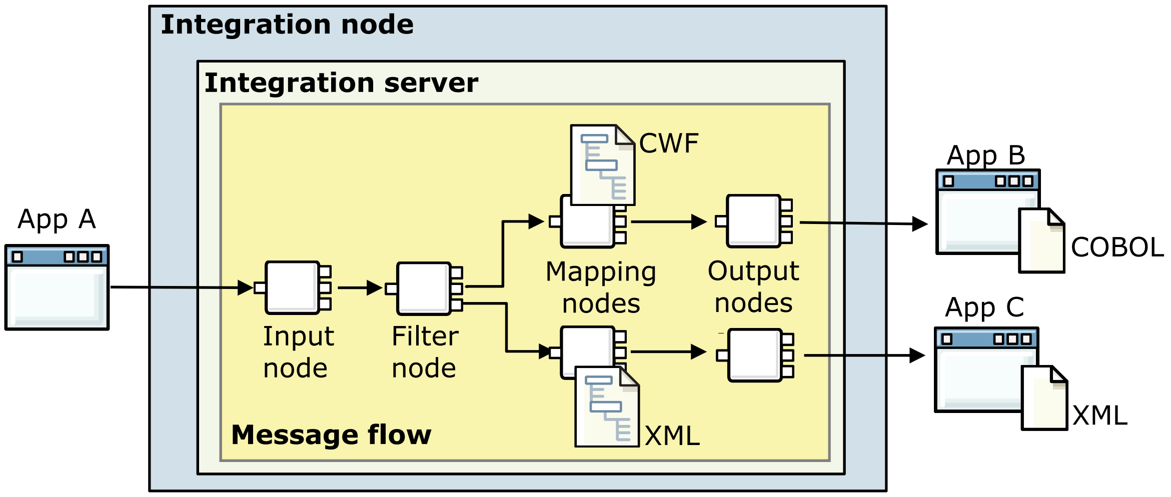
The bottom section of the **Help** view contains relevant links from the IBM Knowledge Center. These links are generated by using a search on the values of the perspective that you are in and the name of the interface element that you selected. The relevance of the search results is variable and depends on the interface element that is selected.

When you click a new interface element, the content of the **Related Topics** changes to match the new object.

4.2. Basic configuration tasks

WebSphere Education IBM

Main components



- **Integration node** routes, transforms, and enriches in-flight messages as determined by message flows and message models
- **Integration server** is a named grouping of message flows
- **Message flow** is a sequence of operations
- **Message flow nodes** define the operations
- **Message models** define structure and properties of the data

Figure 4-12. Main components WM6461.0

Notes:

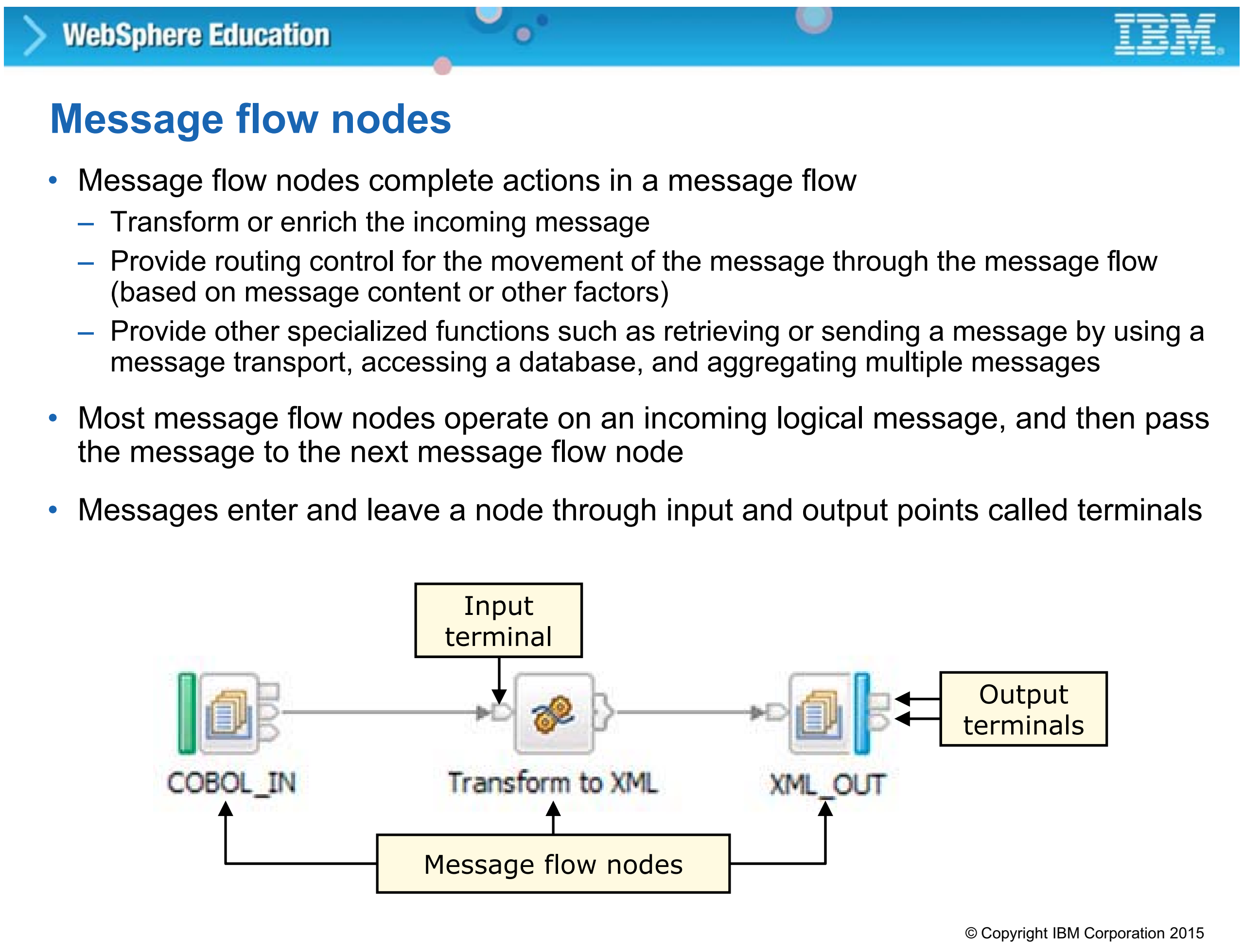
As the number of applications that are being interconnected increases, a service bus is required to provide more application connectivity services to implement an efficient and effective system. As shown in the figure, an Integration Bus integration node provides the service bus functions and centralizes routing and transformation.

In Integration Bus, a *message flow* defines the sequence of operations on a message by a series of message processing nodes. The actions are defined in terms of the message format, its content, and the results of individual actions along the message flow.

A message flow can route messages from sender to recipient based on the content of the message. However, notice that the direction of the flow is, as it was with basic messaging, from the source application to the destination application. In other words, it is assumed that the destination wants, expects, and handles all messages sent to it.

Messages can also be transformed from one format to another before being delivered, perhaps to accommodate the different requirements of the sender and the recipient. The message flow can also transform messages by modifying, combining, adding, or removing data fields. Information can be mapped between messages and databases. More complex manipulation of message data can be achieved by writing code, for example in Extended SQL (ESQL) or Java, within configurable nodes.

An integration server is a named grouping of message flows that are assigned to an integration node. The integration node enforces a degree of isolation between message flows in distinct integration servers by ensuring that they run in separate address spaces, or as unique processes.



Message flow nodes

- Message flow nodes complete actions in a message flow
  - Transform or enrich the incoming message
  - Provide routing control for the movement of the message through the message flow (based on message content or other factors)
  - Provide other specialized functions such as retrieving or sending a message by using a message transport, accessing a database, and aggregating multiple messages
- Most message flow nodes operate on an incoming logical message, and then pass the message to the next message flow node
- Messages enter and leave a node through input and output points called terminals

Figure 4-13. Message flow nodes WM6461.0

Notes:

A message flow node is a self-contained module that completes an action in the message flow. Most message flow nodes act on an incoming message, either by modifying the message or by using the message content to change the routing in the message flow.

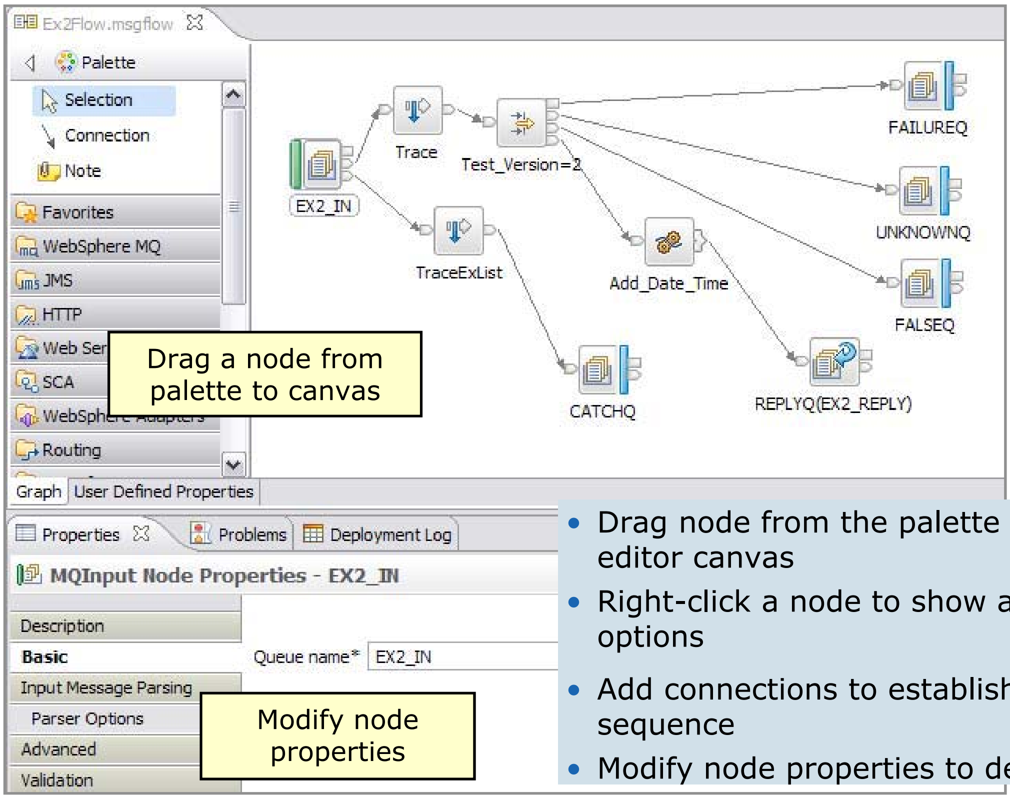
Some types of message flow nodes alter the flow of messages by querying the contents of the message, or by doing operations such as retrieving rows from a relational database. Other message flow nodes access message transports to send or receive messages, such as through IBM MQ, web services, or files. Other message flow nodes exchange data with enterprise information systems.

In most instances, a message enters the message flow node through an input terminal, and exits it through one or more output terminals. Not all message flow nodes have all terminals. Some have only output terminals but no input terminals, while others have only an input terminal. Some message flow nodes have multiple output terminals, and send the message through one or more terminals, depending on the processing that occurs within the node. With some message flow nodes, the developer can add terminals to allow multiple routing paths.

To connect message flow nodes together, you create a connection that is called a wire. Wiring two nodes together provides a path on which a message can flow. When wiring nodes in a message flow, be careful that you are wiring the correct nodes and terminals. Wiring the wrong terminals results in a message flow that does not process messages as expected. It is also possible to send a message through a terminal that is unwired, which discards the message with no warning from the integration node.

WebSphere Education

Create or modify a message flow



- Drag node from the palette to add it to the editor canvas
- Right-click a node to show available options
- Add connections to establish the flow sequence
- Modify node properties to define actions

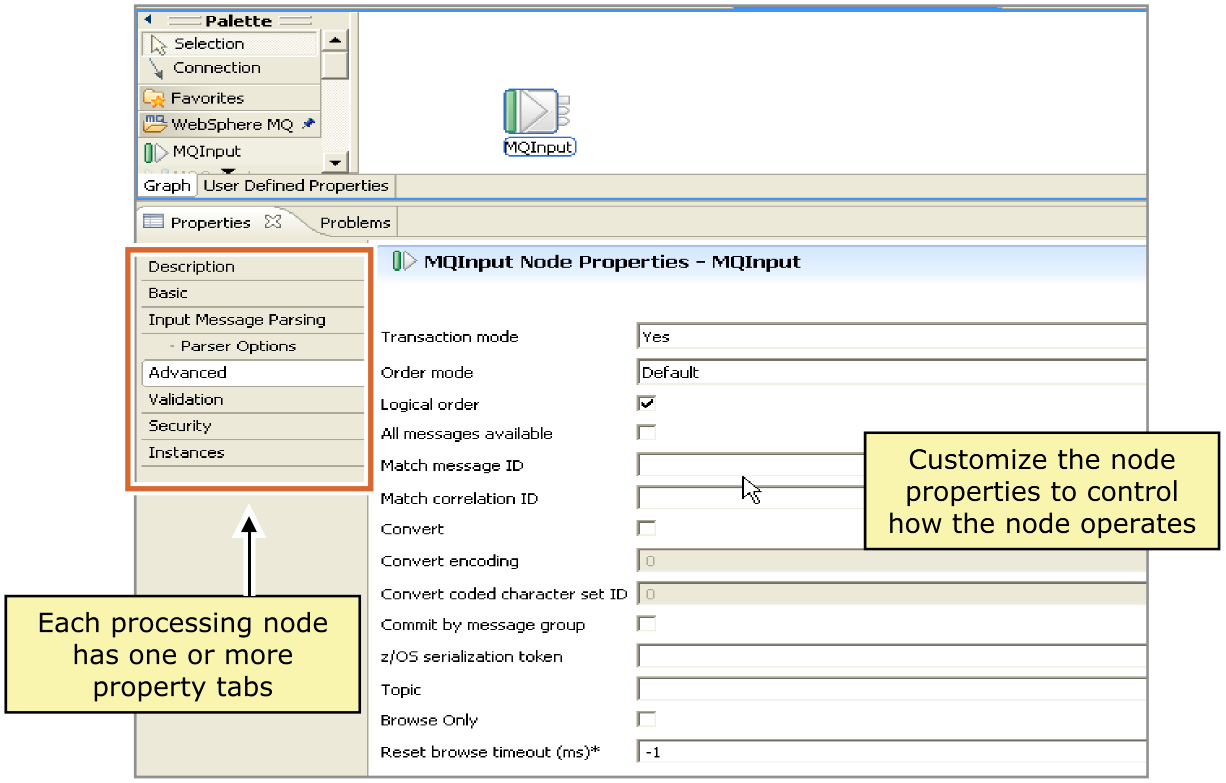
Figure 4-14. Create or modify a message flow WM6461.0

Notes:

Message flows are made up of message processing nodes, represented as icons and positioned in the Message Flow editor. Connections between the nodes establish their runtime sequence. Each node contributes to the overall purpose or intent of the message flow through node properties.

Creating and customizing a message flow, or altering an existing message flow, is procedural in nature. As the figure suggests, defining a message flow requires the completion of many steps, not necessarily in a specific order.

Message flow node properties



MQInput Node Properties - MQInput

Transaction mode	Yes
Order mode	Default
Logical order	✓
All messages available	☐
Match message ID	
Match correlation ID	
Convert	☐
Convert encoding	0
Convert coded character set ID	0
Commit by message group	☐
z/OS serialization token	
Topic	
Browse Only	☐
Reset browse timeout (ms)*	-1

© Copyright IBM Corporation 2015

Figure 4-15. Message flow node properties WM6461.0

Notes:

Message flow nodes names convey their intent and purpose. For example, an MQInput node starts a message flow by getting a message from an IBM MQ queue.

The figure shows the **Properties** pages that are obtained by clicking the node instance in the Message Flow editor. The **Properties** view is displayed, by default, at the bottom of the Integration Development perspective.

By continuing through the several **Properties** pages, the developer can customize the node behavior. As expected, customization options vary among nodes and are directly associated with the intents and purposes of each node. For example, one of the node properties of an MQInput node is the name of the IBM MQ queue from which it is to obtain messages.

WebSphere Education IBM

Managing the integration node in the Integration Toolkit

- In the **Integration Nodes** view:
  - Create a local integration node or connect to a remote integration node
  - Start and stop an integration node, integration server, or message flow
  - Create and delete integration servers
  - Start the IBM Integration web user interface
  - Deploy applications, libraries, message flows, BAR files
  - Verify the status of integration nodes, integration servers, and message flows

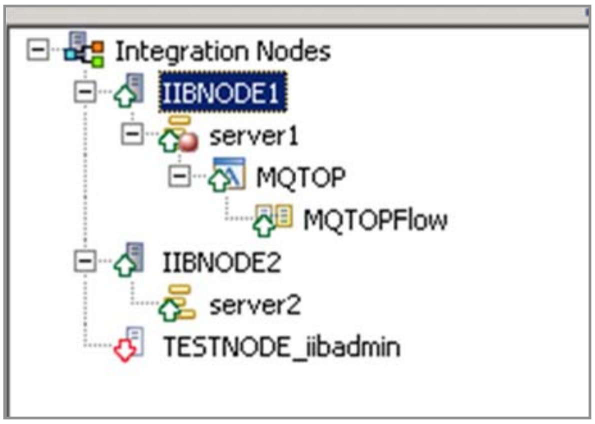


Figure 4-16. Managing the integration node in the Integration Toolkit WM6461.0

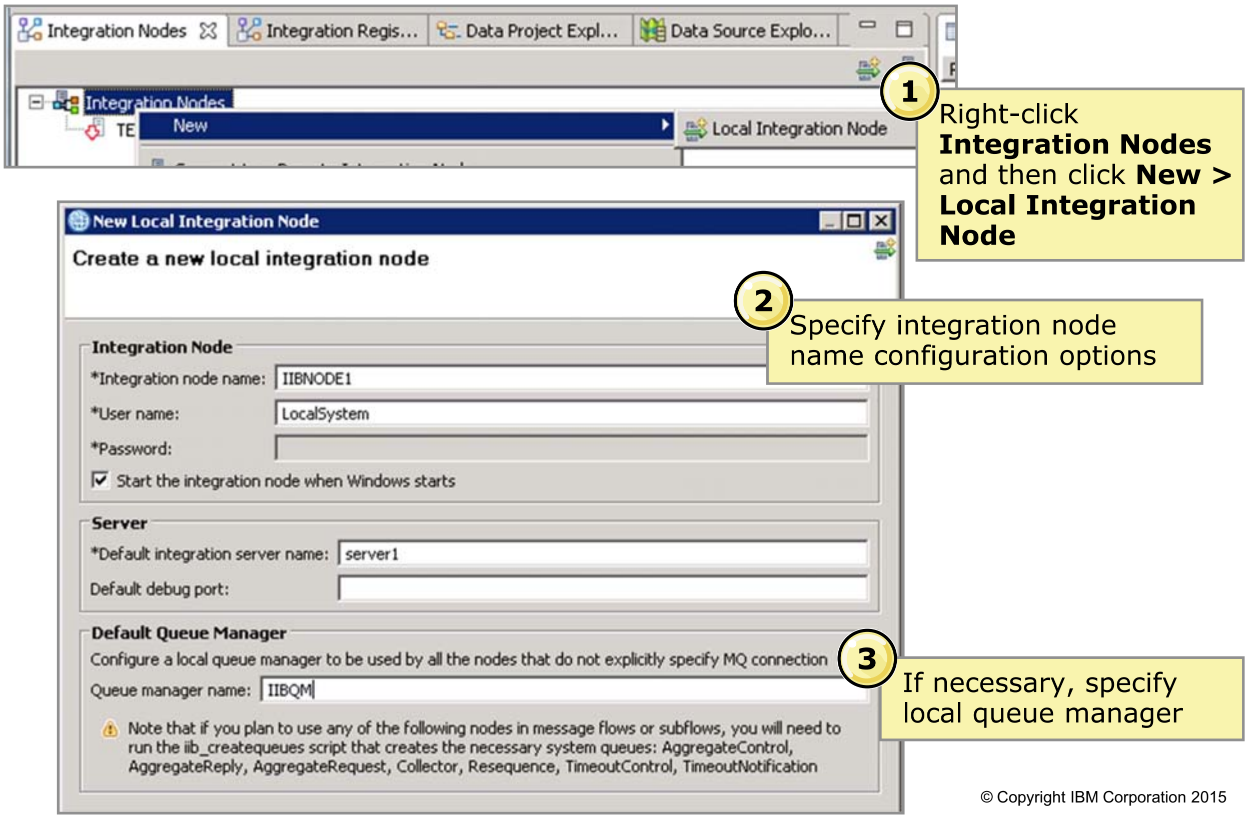
Notes:

Use the **Integration Nodes** view in the Integration Toolkit to create and work with integration nodes in the Integration Toolkit.

Integration nodes that are created on the local system are automatically displayed in the **Integration Nodes** view. Remote integration nodes can be added to the **Integration Nodes** view.

When you open or switch to the **Integration Nodes** view, the Integration Toolkit attempts to connect to integration nodes on the local system and to any remote integration nodes.

Creating an integration node in the IBM Integration Toolkit


Integration Nodes
Integration Regis...
Data Project Expl...
Data Source Explo...
Integration Nodes
TE
New
Local Integration Node
1
Right-click **Integration Nodes** and then click **New > Local Integration Node**
New Local Integration Node
2
Specify integration node name configuration options
3
If necessary, specify local queue manager

Create a new local integration node

Integration Node

*Integration node name: IIBNODE1

*User name: LocalSystem

*Password:

☑ Start the integration node when Windows starts

Server

*Default integration server name: server1

Default debug port:

Default Queue Manager

Configure a local queue manager to be used by all the nodes that do not explicitly specify MQ connection

Queue manager name: IIBQM

Note that if you plan to use any of the following nodes in message flows or subflows, you will need to run the iib_createqueues script that creates the necessary system queues: AggregateControl, AggregateReply, AggregateRequest, Collector, Resequence, TimeoutControl, TimeoutNotification

© Copyright IBM Corporation 2015

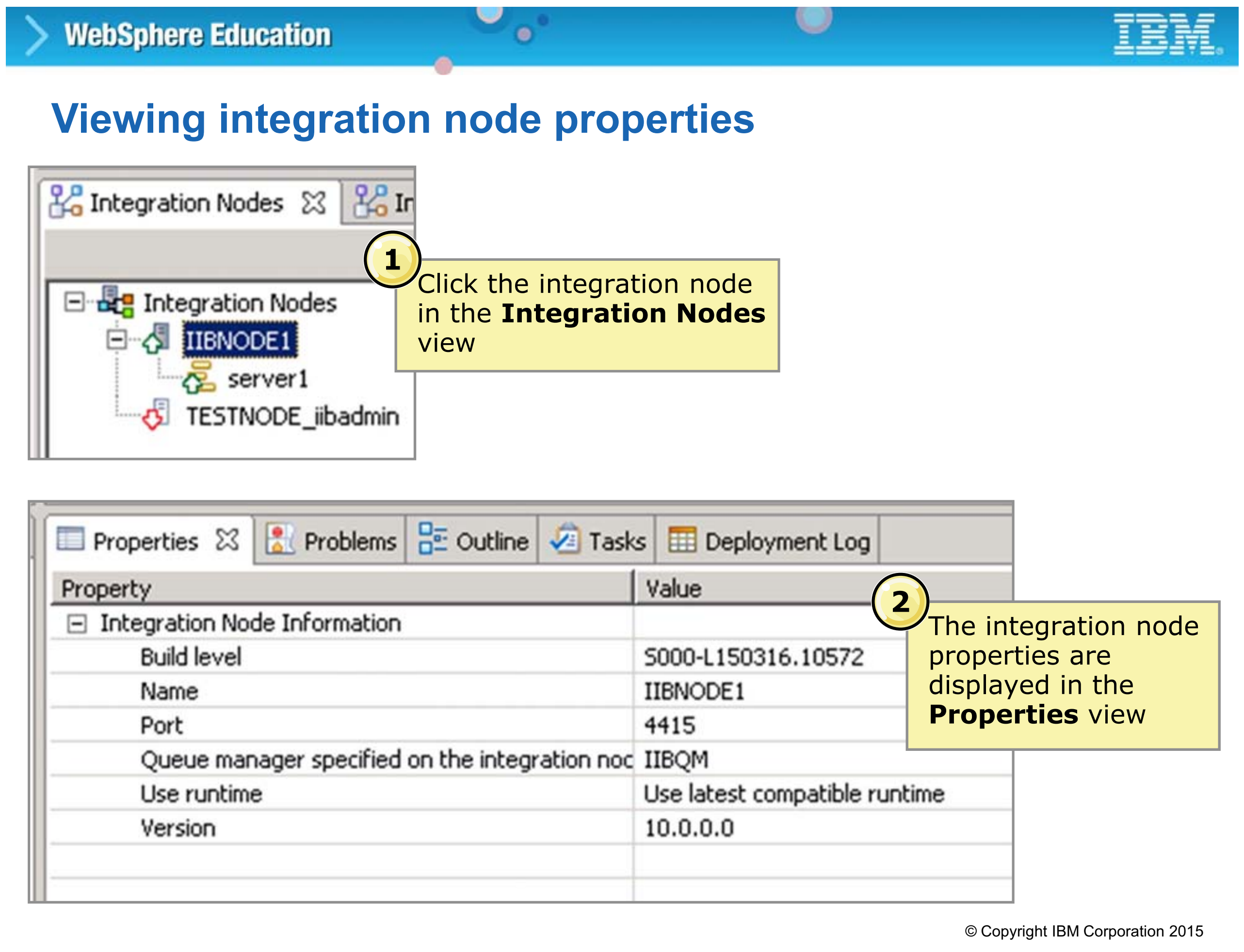
Figure 4-17. Creating an integration node in the IBM Integration Toolkit WM6461.0

Notes:

If you have the necessary permissions, you can create a local integration node by using the Integration Toolkit.

If necessary, you can create the local integration node to use a default IBM MQ queue manager.

Viewing integration node properties


WebSphere Education
IBM
Integration Nodes
Integration Nodes
IIBNODE1
server1
TESTNODE_iibadmin
1
Click the integration node in the **Integration Nodes** view
Properties
Problems
Outline
Tasks
Deployment Log
2
The integration node properties are displayed in the **Properties** view
© Copyright IBM Corporation 2015

Property	Value
Integration Node Information	
Build level	S000-L150316.10572
Name	IIBNODE1
Port	4415
Queue manager specified on the integration noc	IIBQM
Use runtime	Use latest compatible runtime
Version	10.0.0.0

Figure 4-18. Viewing integration node properties WM6461.0

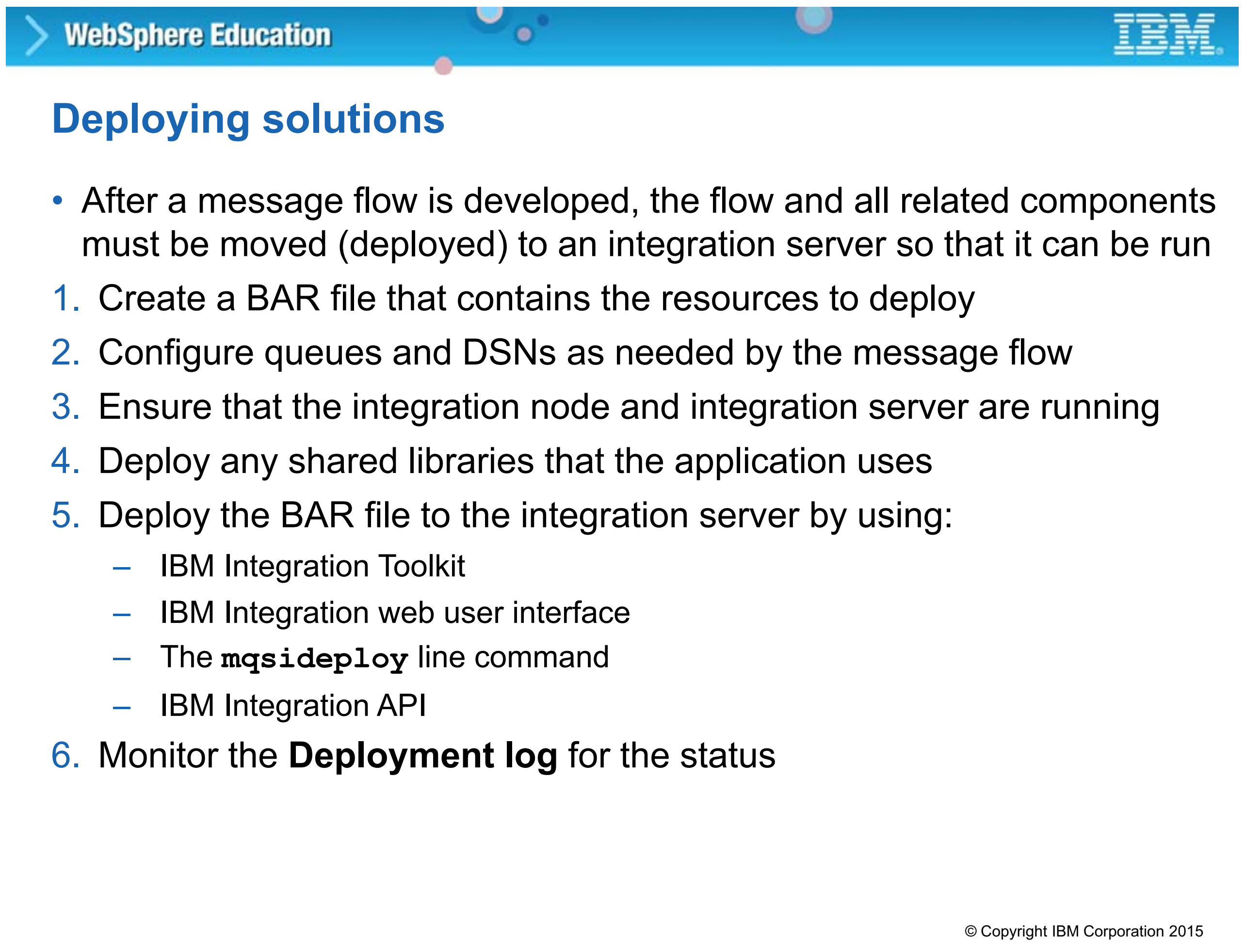
Notes:

You can view the integration node properties in the Integration Toolkit.

You can also view the integration server, and message flow application properties in the Integration Toolkit.

To view the integration server properties, click the integration server in the **Integration Nodes** view. The integration server properties are displayed on the **Properties** tab.

4.3. Deploying message flows from the Integration Toolkit



WebSphere Education

Deploying solutions

- After a message flow is developed, the flow and all related components must be moved (deployed) to an integration server so that it can be run

1. Create a BAR file that contains the resources to deploy
2. Configure queues and DSNs as needed by the message flow
3. Ensure that the integration node and integration server are running
4. Deploy any shared libraries that the application uses
5. Deploy the BAR file to the integration server by using:
   - IBM Integration Toolkit
   - IBM Integration web user interface
   - The **`mqsideploy`** line command
   - IBM Integration API
6. Monitor the **Deployment log** for the status

© Copyright IBM Corporation 2015

Figure 4-19. Deploying solutions WM6461.0

Notes:

To run a message flow, you must move it to an integration server. This process is called *deployment*.

When you use the Integration Toolkit Flow exerciser to test the flow, the application is deployed automatically. If you are not using the Flow exerciser or need more control over the deployment process, you must create a BAR file and deploy it manually.

The BAR file contains the resources for the application. Resources can include message flows, message models, and ESQL code.

After the BAR file is created, you deploy it to an integration server. During deployment, the BAR file is unpacked and the message flow components are placed into the integration server libraries.

This figure summarizes the steps for deploying Integration Bus solutions. The next pages describe these steps in more detail.

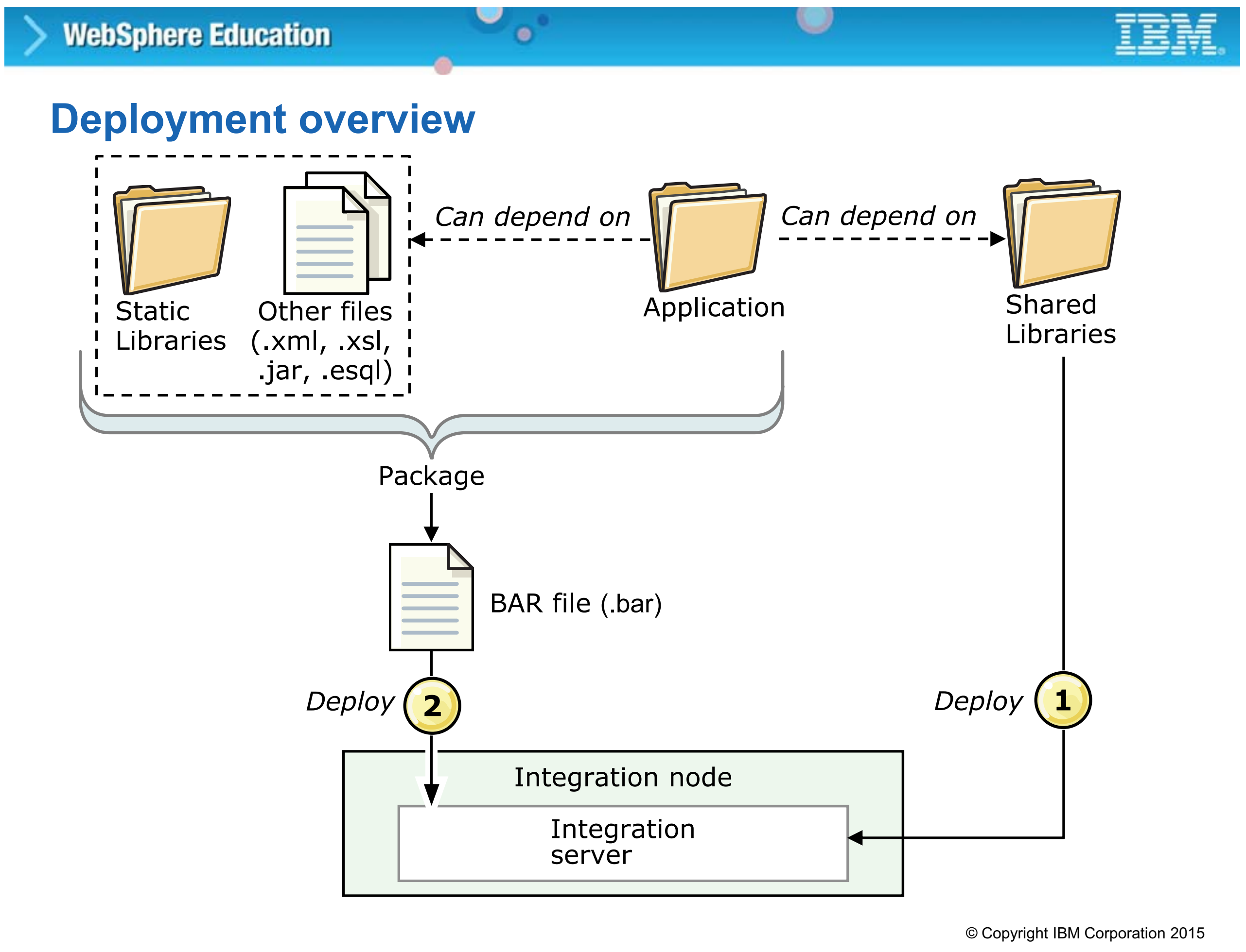


Figure 4-20. Deployment overview WM6461.0

Notes:

Deployment is the process of moving Integration Bus applications from the Integration Toolkit to an integration server.

A deployment operation takes some time; exactly how much depends on the nature of network connections and the volume of information that is being exchanged. Administrators have some control over the volume of information that is exchanged in that they can select between a complete or incremental deployment.

- In a **complete** deployment, files that are already deployed to the integration server are removed before the complete contents of the BAR file is deployed. So, nothing is left in the integration server from any previous deployment. Do *not* use a complete deployment if you want to merge the existing contents of the integration server with the contents of the BAR file.
- In an **incremental** deployment, deployed files are added to the integration server. Newer versions of the files overwrite existing ones in the integration server. Do *not* use an incremental deployment if you want to clear the contents of the integration server before the BAR file is deployed.

As you might expect, efficiency can be achieved regarding the packaging of the deployment data. Reducing the volume, by compression, reduces the exchange time.

WebSphere Education

BAR file contents

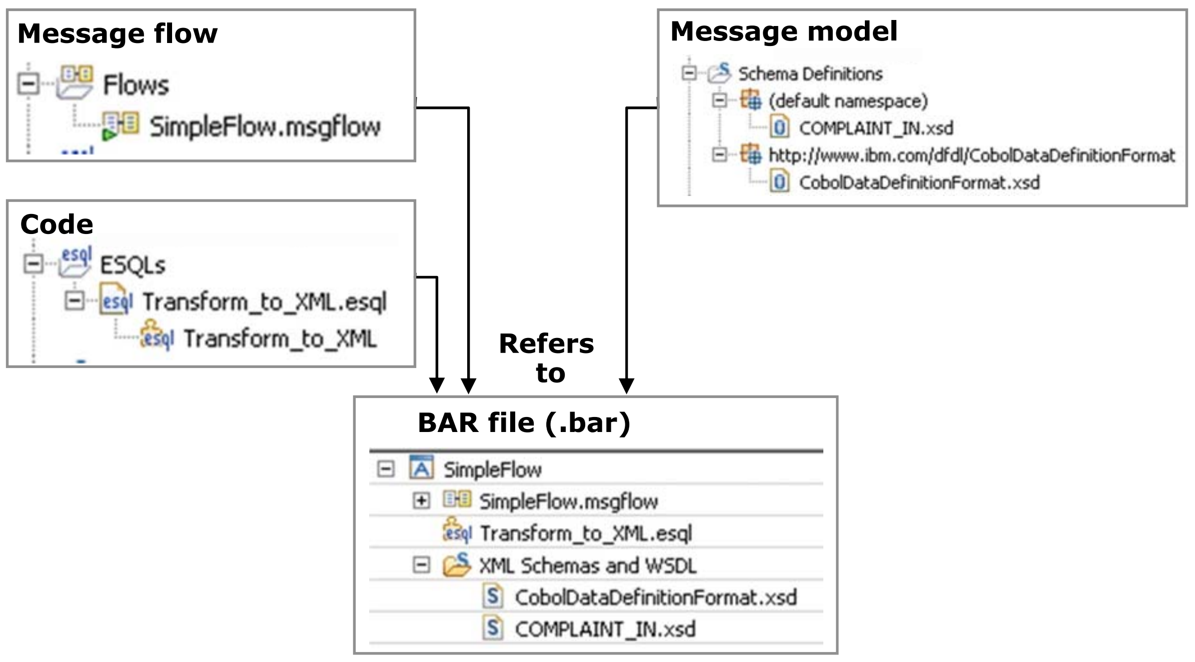


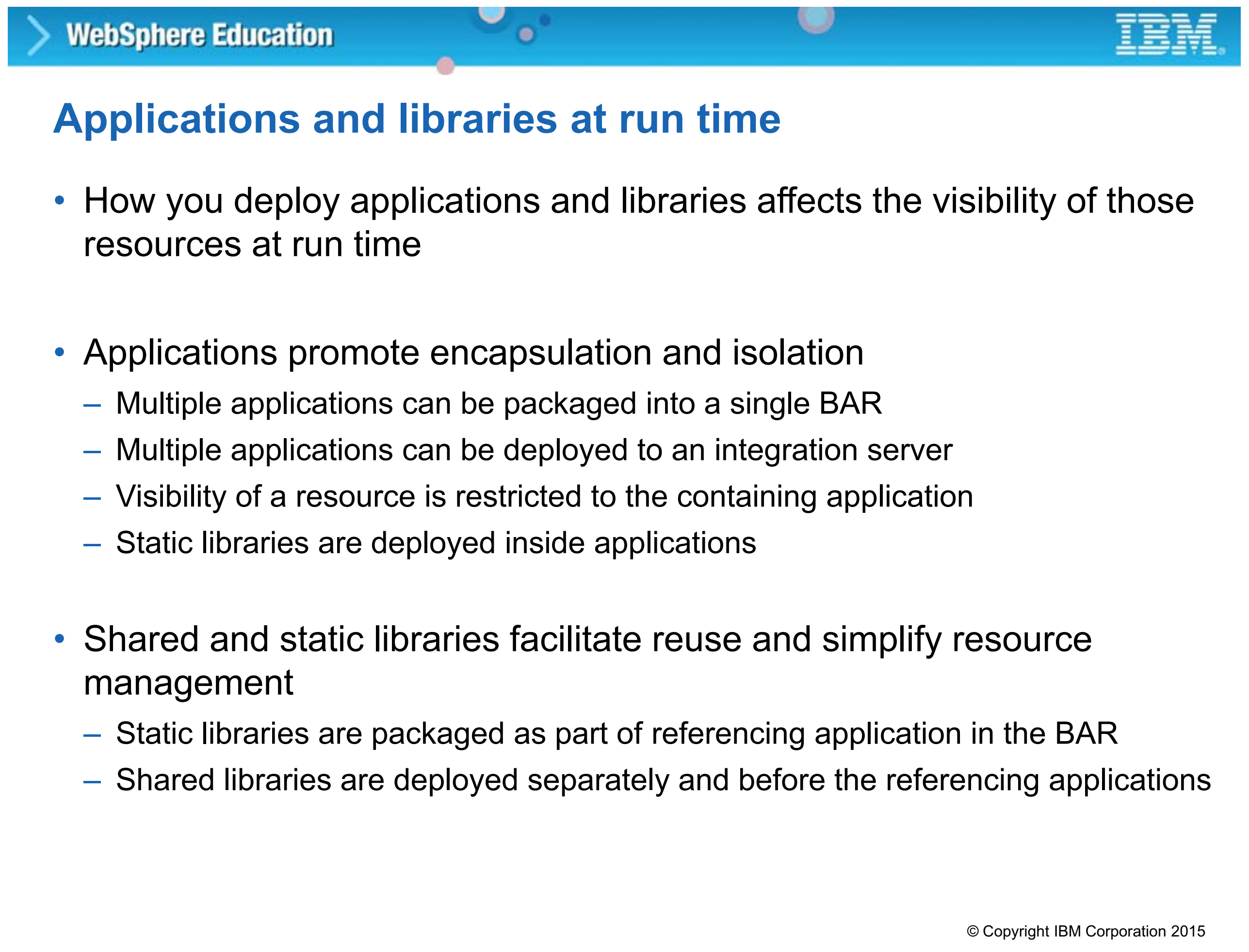
Figure 4-21. BAR file contents WM6461.0

Notes:

The BAR file can contain many different files such as:

- A message flow (`.msgflow`) file for each message flow
- Message model schemas
- XML files (`.xml`) and stylesheets (`.xsl` files) for use with the XSLTransform node
- ESQL files for use with the Compute node or JAR flies for use with JavaCompute node

The BAR file also contains `broker.xml` file. This file is called the *broker deployment* descriptor. This file, in XML format, in the `META-INF` folder of the `.zip` file and can be modified by using a text editor or shell script.



WebSphere Education

Applications and libraries at run time

- How you deploy applications and libraries affects the visibility of those resources at run time
- Applications promote encapsulation and isolation
  - Multiple applications can be packaged into a single BAR
  - Multiple applications can be deployed to an integration server
  - Visibility of a resource is restricted to the containing application
  - Static libraries are deployed inside applications
- Shared and static libraries facilitate reuse and simplify resource management
  - Static libraries are packaged as part of referencing application in the BAR
  - Shared libraries are deployed separately and before the referencing applications

© Copyright IBM Corporation 2015

Figure 4-22. Applications and libraries at run time WM6461.0

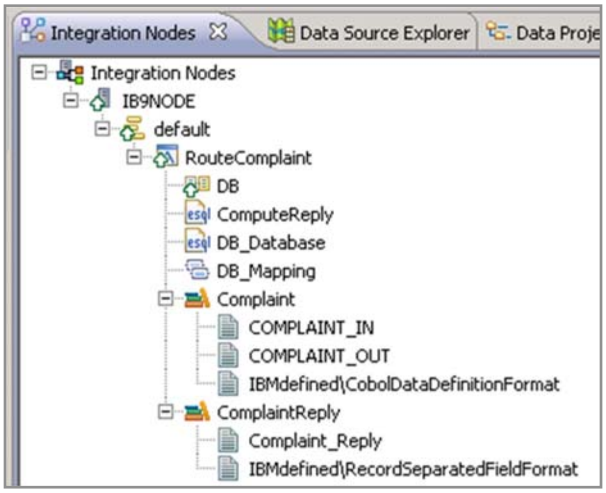
Notes:

In Integration Bus, you store development artifacts in containers that are known as *applications* and *libraries*. How you deploy applications and libraries affects the visibility of those resources at run time.

WebSphere Education IBM

Applications and libraries on the integration node

- Each application is an independent container within the integration server
- When one or more applications references a static library, a copy of the library is deployed to each application container that references that library, which provides a certain level of isolation
- When an update to a static library is deployed to an application, it flows only inside the affected application area
- Shared libraries are deployed and maintained separately
- All applications and libraries are stored as unmodified files on the work path of the target integration node



```
Integration Nodes
  IB9NODE
    default
      RouteComplaint
        DB
        ComputeReply
        DB_Database
        DB_Mapping
        Complaint
          COMPLAINT_IN
          COMPLAINT_OUT
          IBMdefined\CobolDataDefinitionFormat
        ComplaintReply
          Complaint_Reply
          IBMdefined\RecordSeparatedFieldFormat
```

Figure 4-23. Applications and libraries on the integration node WM6461.0

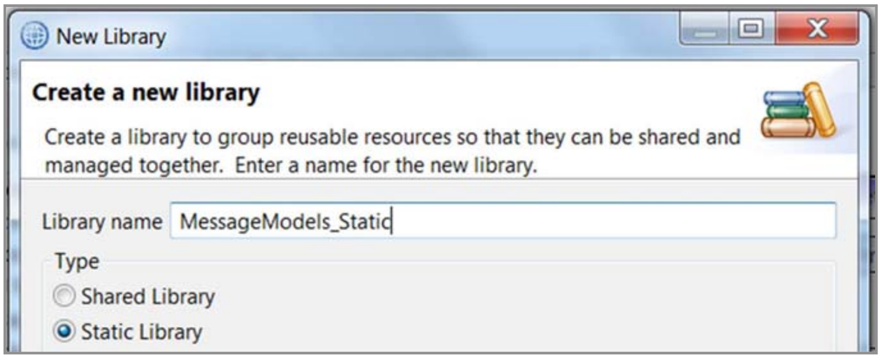
Notes:

In a production environment, you might have multiple Integration Bus applications that use the same resources or reference the same libraries. This figure describes the behavior of applications and libraries at run time.

WebSphere Education IBM

Static libraries

- Application gets its own copy of the library and the resources that it contains
- To update a library, every application that uses the library must be repackaged and redeployed with the updates
- During packaging and deployment process, the library hierarchy is ignored and the libraries are flattened into a list



New Library

Create a new library

Create a library to group reusable resources so that they can be shared and managed together. Enter a name for the new library.

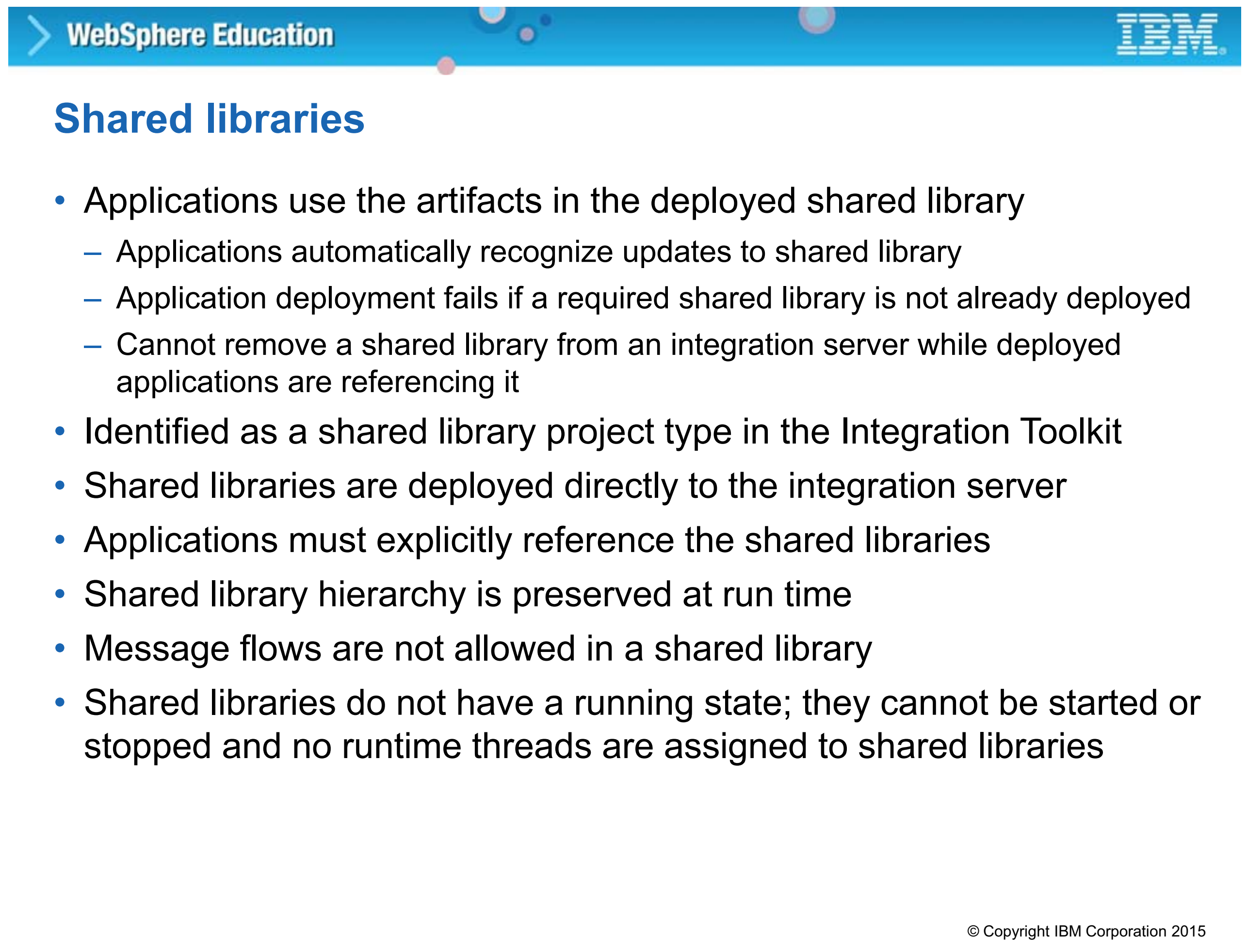
Library name: MessageModels_Static

Type
- Shared Library
- [✓ Static Library]

Figure 4-24. Static libraries WM6461.0

Notes:

One or more applications can reference a static library. Changes that are made to the library in the Integration Toolkit are available to all applications that reference that library. However, when the applications are packaged into a BAR file and deployed, each application has its own copy of the library and the resources that are contained in it. If you update a static library, you must repackage and redeploy each application that references that library.



WebSphere Education

Shared libraries

- Applications use the artifacts in the deployed shared library
  - Applications automatically recognize updates to shared library
  - Application deployment fails if a required shared library is not already deployed
  - Cannot remove a shared library from an integration server while deployed applications are referencing it
- Identified as a shared library project type in the Integration Toolkit
- Shared libraries are deployed directly to the integration server
- Applications must explicitly reference the shared libraries
- Shared library hierarchy is preserved at run time
- Message flows are not allowed in a shared library
- Shared libraries do not have a running state; they cannot be started or stopped and no runtime threads are assigned to shared libraries

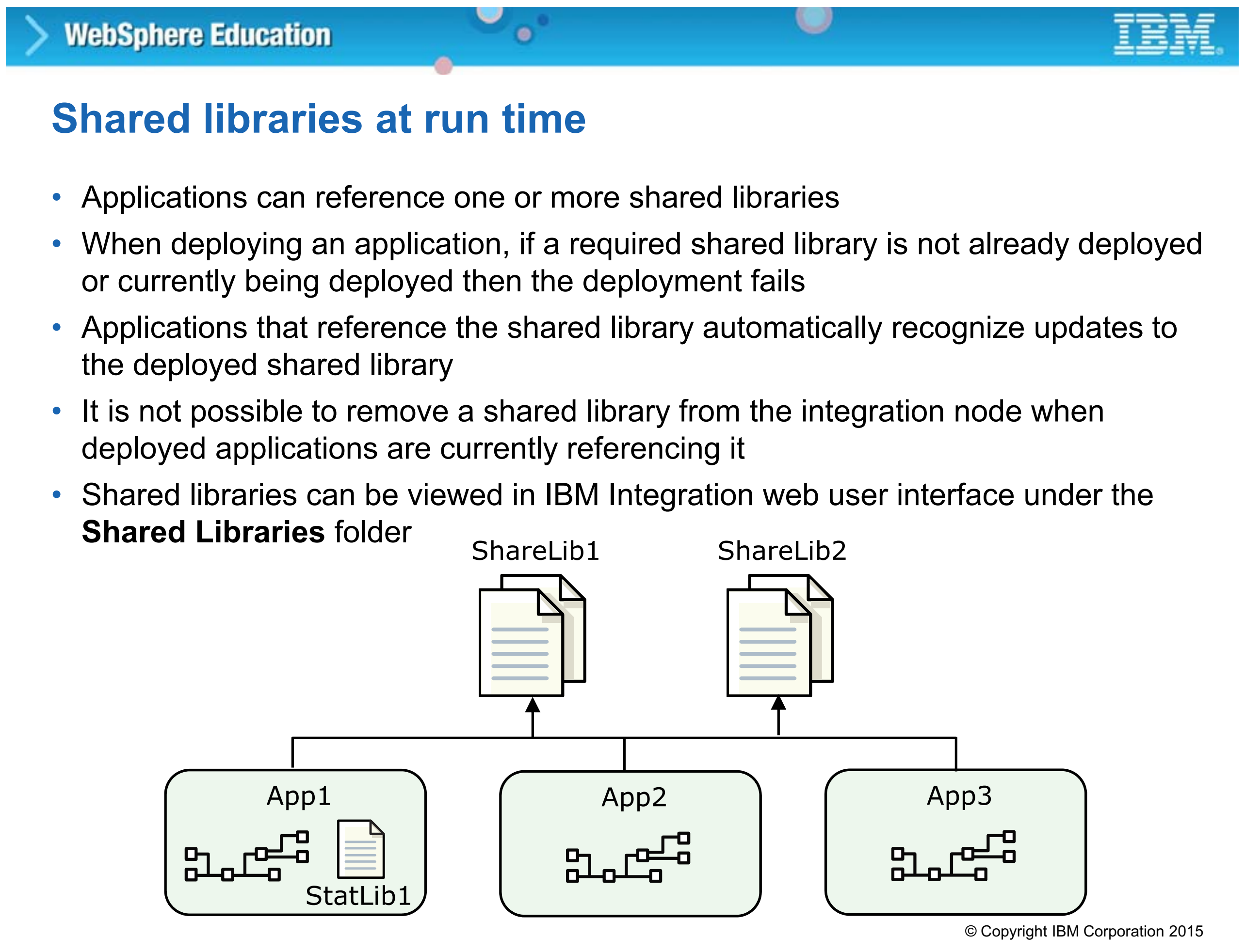
Figure 4-25. Shared libraries WM6461.0

Notes:

A shared library can be deployed directly to an integration server. Any application can reference the resources in that deployed shared library. If that shared library is updated, all referencing applications recognize the changes immediately.

A shared library must be deployed with or before an application that references it. If you deploy resources by adding them to a BAR file, ensure that you include any shared libraries that your application references.

Similarly, you cannot remove a deployed shared library while deployed applications are referencing it. Shared libraries can reference other shared libraries, but not static libraries. Similarly, a static library cannot reference a shared library.



WebSphere Education IBM

Shared libraries at run time

- Applications can reference one or more shared libraries
- When deploying an application, if a required shared library is not already deployed or currently being deployed then the deployment fails
- Applications that reference the shared library automatically recognize updates to the deployed shared library
- It is not possible to remove a shared library from the integration node when deployed applications are currently referencing it
- Shared libraries can be viewed in IBM Integration web user interface under the **Shared Libraries** folder

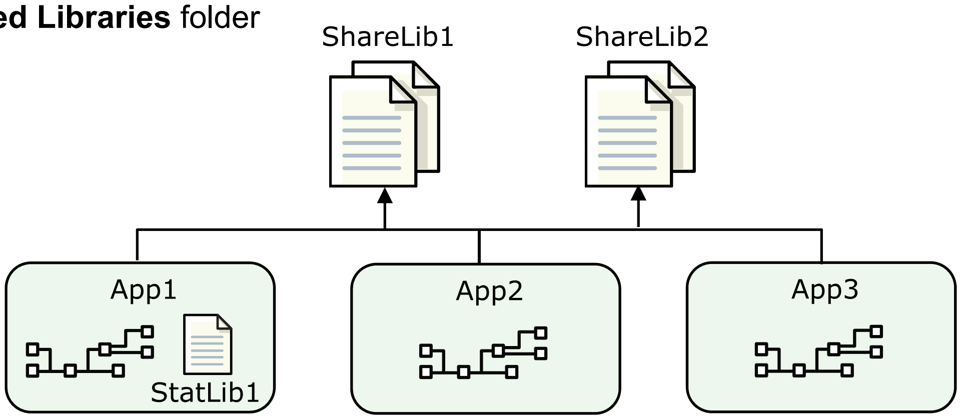


Figure 4-26. Shared libraries at run time WM6461.0

Notes:

A shared library can be deployed directly to an integration server. Any application can reference the resources in that deployed shared library. If that shared library is updated, the referencing applications immediately recognize the changes.

A shared library must be deployed with or before an application that references it.

An administrator can view the shared libraries that are deployed to an integration server by using the Integration web user interface.

Broker schemas (namespaces)

- Integration project subdirectory that organizes objects into a namespace to prevent duplicate names in the workspace or in the integration node
  - Defined as the relative path from the project source directory to the flow name
  - Can be added to an object in the Integration Toolkit by right-click the object and then clicking **New > Broker Schema**
- Broker schema is propagated to the integration node with deployment
- Required in shared libraries for some files (ESQL files, subflows) to avoid conflicts between files in shared library and applications

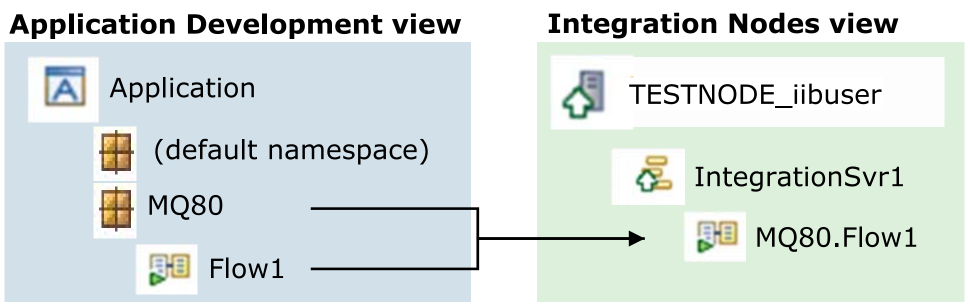
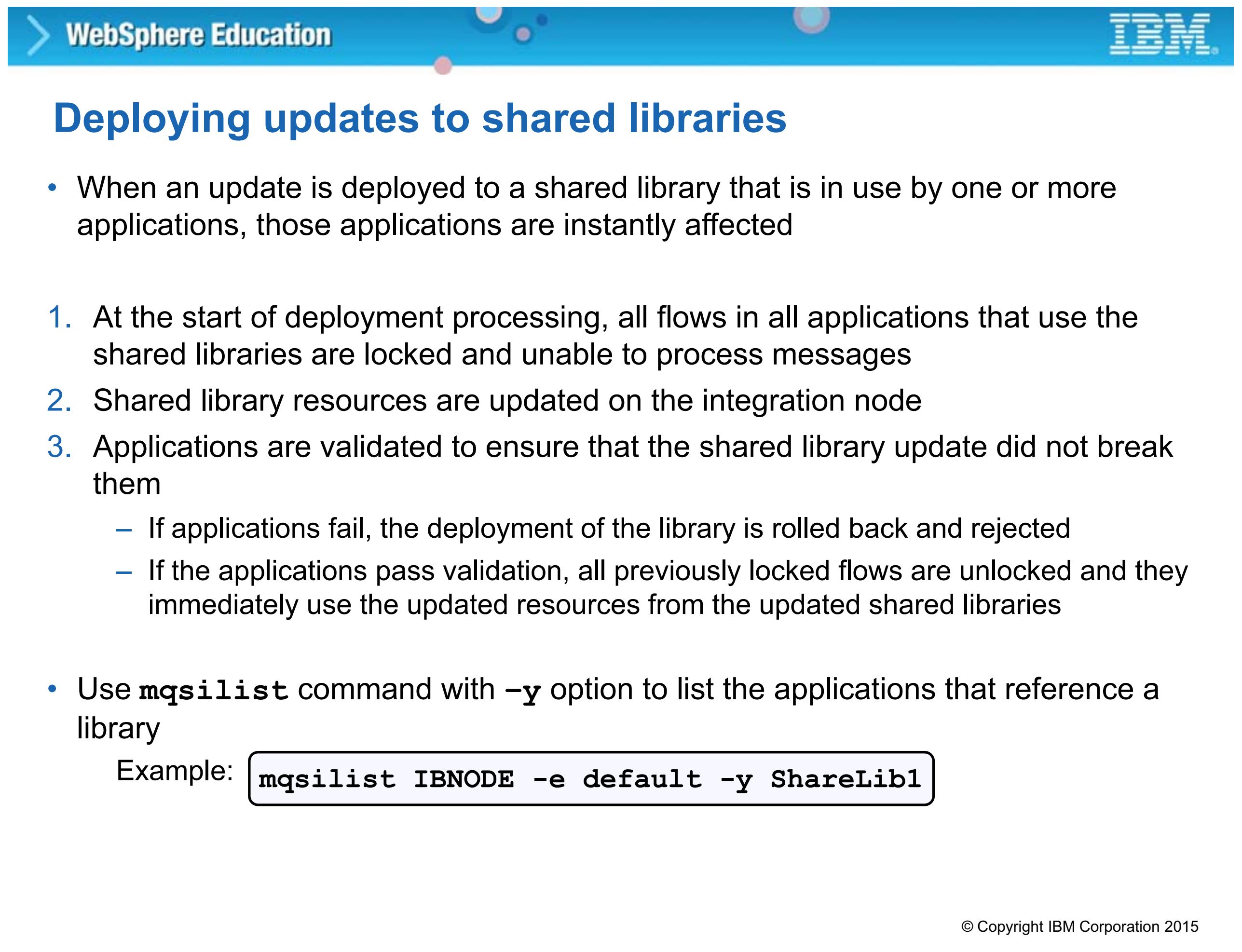


Figure 4-27. Broker schemas (namespaces) WM6461.0

Notes:

Integration Bus objects are identified by using a combination of a namespace and a name, referred to as a fully qualified name. The use of fully qualified names makes it easy to identify and locate objects, and to correct broken references. For example, if an object is renamed, all references to that object are broken. If an object is substituted for another object with the same name, all the broken references are corrected. This concept, called name linking, is important when working in a team environment. It means that you can share files in a repository and concurrently modify, add, and delete objects in the message flow application. When you integrate the various parts of the application, you can detect and resolve broken references to objects that were previously moved, renamed, or deleted.

The Integration Bus namespace is referred to as a *broker schema.* It is defined as the relative path from the project source directory to the flow name. When you first create a message flow project, a default broker schema that is named `(default)` is created within the project. Because each broker schema represents a unique name scope, you can create two different message flows that share a name within two different broker schemas. The broker schemas ensure that the two message flows are recognized as separate resources. The two message flows with the same base name, are considered unique.



WebSphere Education

Deploying updates to shared libraries

- When an update is deployed to a shared library that is in use by one or more applications, those applications are instantly affected

1. At the start of deployment processing, all flows in all applications that use the shared libraries are locked and unable to process messages
2. Shared library resources are updated on the integration node
3. Applications are validated to ensure that the shared library update did not break them
   - If applications fail, the deployment of the library is rolled back and rejected
   - If the applications pass validation, all previously locked flows are unlocked and they immediately use the updated resources from the updated shared libraries

- Use **`mqsilist`** command with **`-y`** option to list the applications that reference a library

  Example:

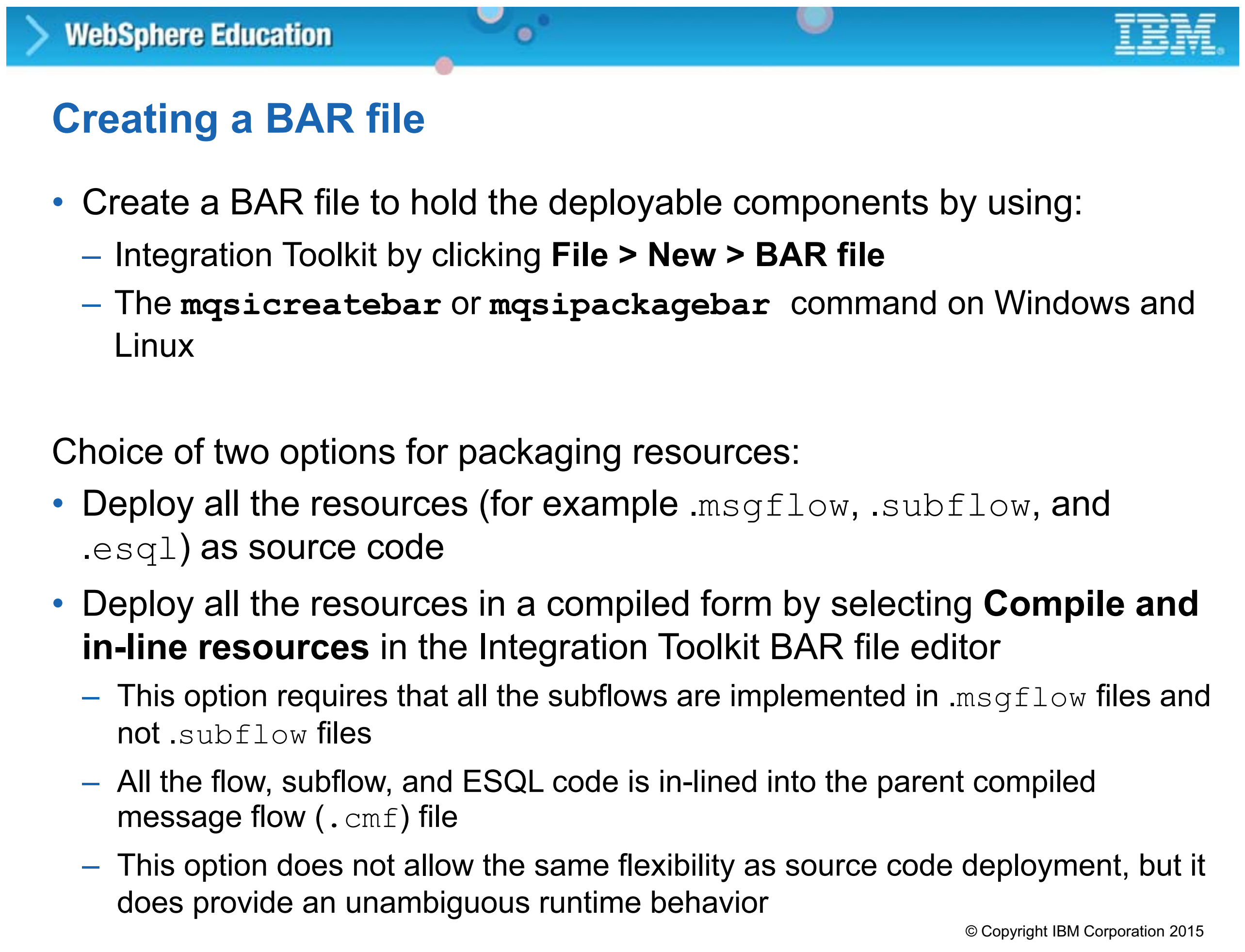
```
mqsilist IBNODE -e default -y ShareLib1
```

Figure 4-28. Deploying updates to shared libraries WM6461.0

Notes:

Use care when deploying updates to shared libraries. When an update is deployed to shared library, all applications that use the shared libraries are affected.

This figure describes the internal process that occurs when you deploy an update to a shared library.



WebSphere Education
IBM

Creating a BAR file

- Create a BAR file to hold the deployable components by using:
  – Integration Toolkit by clicking **File > New > BAR file**
  – The **`mqsicreatebar`** or **`mqsipackagebar`** command on Windows and Linux

Choice of two options for packaging resources:

- Deploy all the resources (for example `.msgflow`, `.subflow`, and `.esql`) as source code
- Deploy all the resources in a compiled form by selecting **Compile and in-line resources** in the Integration Toolkit BAR file editor
  – This option requires that all the subflows are implemented in `.msgflow` files and not `.subflow` files
  – All the flow, subflow, and ESQL code is in-lined into the parent compiled message flow (`.cmf`) file
  – This option does not allow the same flexibility as source code deployment, but it does provide an unambiguous runtime behavior

Figure 4-29. Creating a BAR file WM6461.0

Notes:

The message flow and all related components that the message flow uses must be deployed to the integration server. To move components, you create a BAR file, and move all the required resources into it.

When you add files to the BAR file, they can be automatically compiled as part of the process. JAR files that the JavaCompute node requires in message flows are added automatically from your Java project. When the BAR file is created, it is automatically compressed and prepared for deployment.

If your BAR file contains a mixture of resources that are compiled and resources that are not compiled, you might see unexpected results. For example, if you select the **Compile and in-line resources** option to create a BAR file that contains an ESQL file and a message flow, the ESQL is embedded in the compiled message flow (`.cmf`) file. If you then update the ESQL and add it to the BAR file with the **Compile and in-line resources** option not selected, the ESQL file is added as an individual resource, but the `.cmf` file uses the original ESQL because the original ESQL remains embedded in the `.cmf` file. To avoid confusion, select one of the following deployment options:

- Deploy all the resources (for example `.msgflow`, `.subflow`, and `.esql`) as source code.

- Deploy all the resources in the compiled form. This deployment method requires that all the subflows are implemented in `.msgflow` files. As a result, all the flow, subflow, and ESQL code is in-lined into the parent CMF file. This method does not allow the same flexibility as the source code deployment, but it does provide an unambiguous runtime behavior

Warning

If a BAR file contains a mixture of resources that are compiled and resources that are not compiled, you might see unexpected results.

When you add a message flow to a BAR file, you can add the flow as a `.msgflow` file, or as a compiled message flow (defined in a `.cmf` file). You cannot add the same message flow to a BAR file as both a `.cmf` file and a `.msgflow` file. If your flow contains one of the following nodes, you cannot add the flow as a `.msgflow` file:

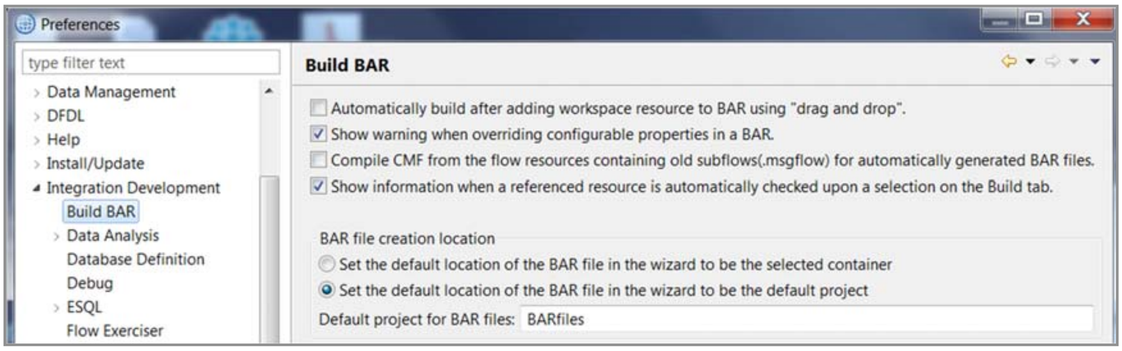
- A user-defined node that is created from a subflow
- A subflow node that represents a subflow that is defined in a message flow file
- A WebSphere Message Broker Version 7.0 Mapping node
- An MQOptimizedFlow node

WebSphere Education IBM

Creating the BAR file in the Integration Toolkit

- From the IBM Integration Toolkit:
  1. Click **File > New > BAR File**
  2. Specify the name of the BAR file
  3. Add components to the BAR file by using the editor, or drag components from the Application Development view

- By default, the BAR file is created in the **BarFiles** project
  – To change this value in the IBM Integration Toolkit, click **Window > Preferences**, expand **Integration Development**, and then select **Build BAR**



Build BAR

☐ Automatically build after adding workspace resource to BAR using "drag and drop".
☑ Show warning when overriding configurable properties in a BAR.
☐ Compile CMF from the flow resources containing old subflows(.msgflow) for automatically generated BAR files.
☑ Show information when a referenced resource is automatically checked upon a selection on the Build tab.

BAR file creation location
○ Set the default location of the BAR file in the wizard to be the selected container
● Set the default location of the BAR file in the wizard to be the default project

Default project for BAR files: BARfiles

© Copyright IBM Corporation 2015

Figure 4-30. Creating the BAR file in the Integration Toolkit WM6461.0

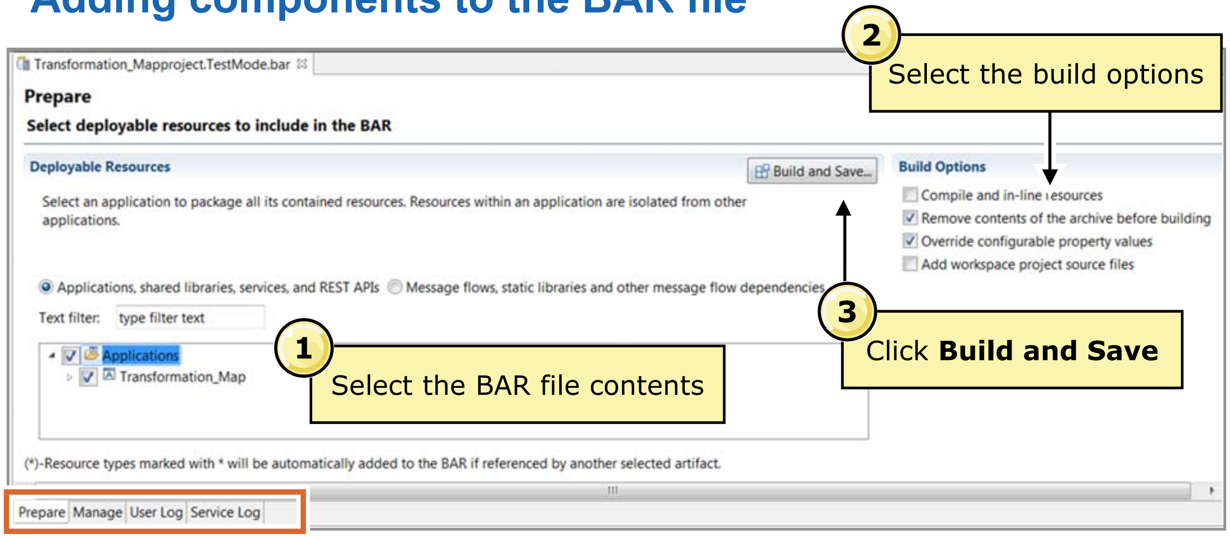
Notes:

The figure describes one of the ways to create a BAR file in the Integration Toolkit.

By default the BAR file is created in a special **BarFiles** project.

After the BAR file is created, you can use the BAR file editor to add and remove components and change the deployment options. You can also add components to the BAR file in the **Application Development** view by dragging them and dropping them into the BAR file.

Adding components to the BAR file


Transformation_Mapproject.TestMode.bar
Prepare
Select deployable resources to include in the BAR
Deployable Resources
Build and Save...
Build Options
Select an application to package all its contained resources. Resources within an application are isolated from other applications.
Compile and in-line resources
Remove contents of the archive before building
Override configurable property values
Add workspace project source files
Applications, shared libraries, services, and REST APIs
Message flows, static libraries and other message flow dependencies
Text filter: type filter text
Applications
Transformation_Map
(*)-Resource types marked with * will be automatically added to the BAR if referenced by another selected artifact.
Prepare
Manage
User Log
Service Log
1 Select the BAR file contents
2 Select the build options
3 Click Build and Save

- **Prepare** view build options include:
  - **Include source files** to include source files in the BAR file
  - **Remove contents of the archive before building** to remove all existing contents of the BAR file before building the new BAR file
  - **Override configurable properties values** to override the values from the message flow

Figure 4-31. Adding components to the BAR file WM6461.0

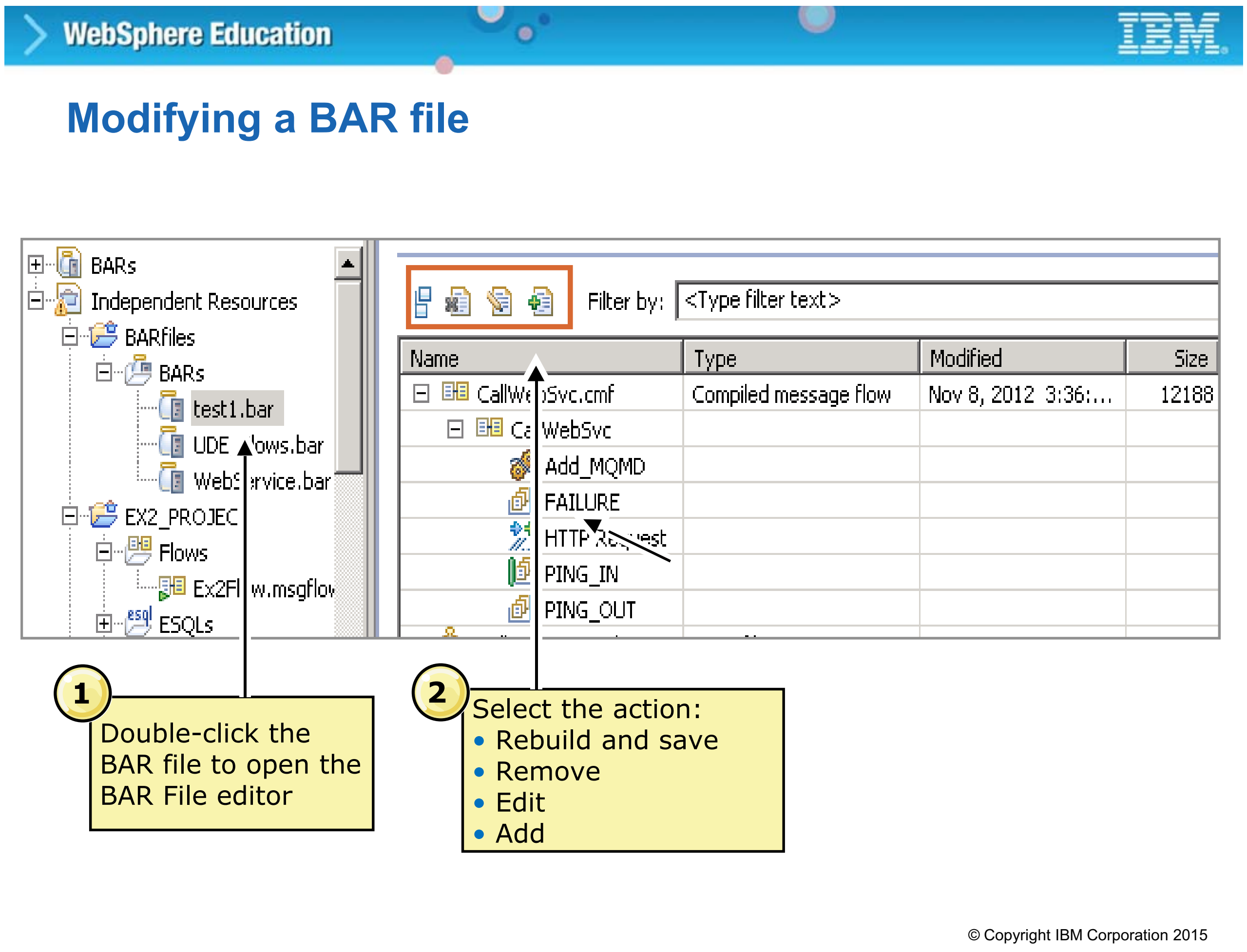
Notes:

As shown in the figure, you populate the BAR file with artifacts from the workspace on the BAR file editor **Prepare** tab by checking the resources that are required and clicking **Build and save**.

By default, the **Prepare** tab shows all the resources that are available in the workspace. It is possible to limit the list to only those resources that are contained within a particular working set. If you define a working set that contains only the resources that are needed for a particular application, you can select this working set on the BAR file builder. The BAR file builder then shows the relevant resources only.

If you elect to include source files in the BAR file, these resources are stored in a separate folder, which is named `src`. A separate folder clearly distinguishes source components and the component that the integration node runtime environment uses.

Compile errors prevent components from being successfully deployed. If you receive a message about a failed deployment, be sure to consult the **Problems** view and correct any issues. However, components that display warning messages are included in the BAR file.



Modifying a BAR file

1 Double-click the BAR file to open the BAR File editor

2 Select the action:
- Rebuild and save
- Remove
- Edit
- Add

Figure 4-32. Modifying a BAR file WM6461.0

Notes:

After the BAR file is created, you can use the BAR file editor to add and remove components and change the deployment options. You can also add components to the BAR file in the Application Development view by dragging them into the BAR file.

To modify a BAR file:

1. Double-click the file with the `.bar` extension to open the BAR File editor.

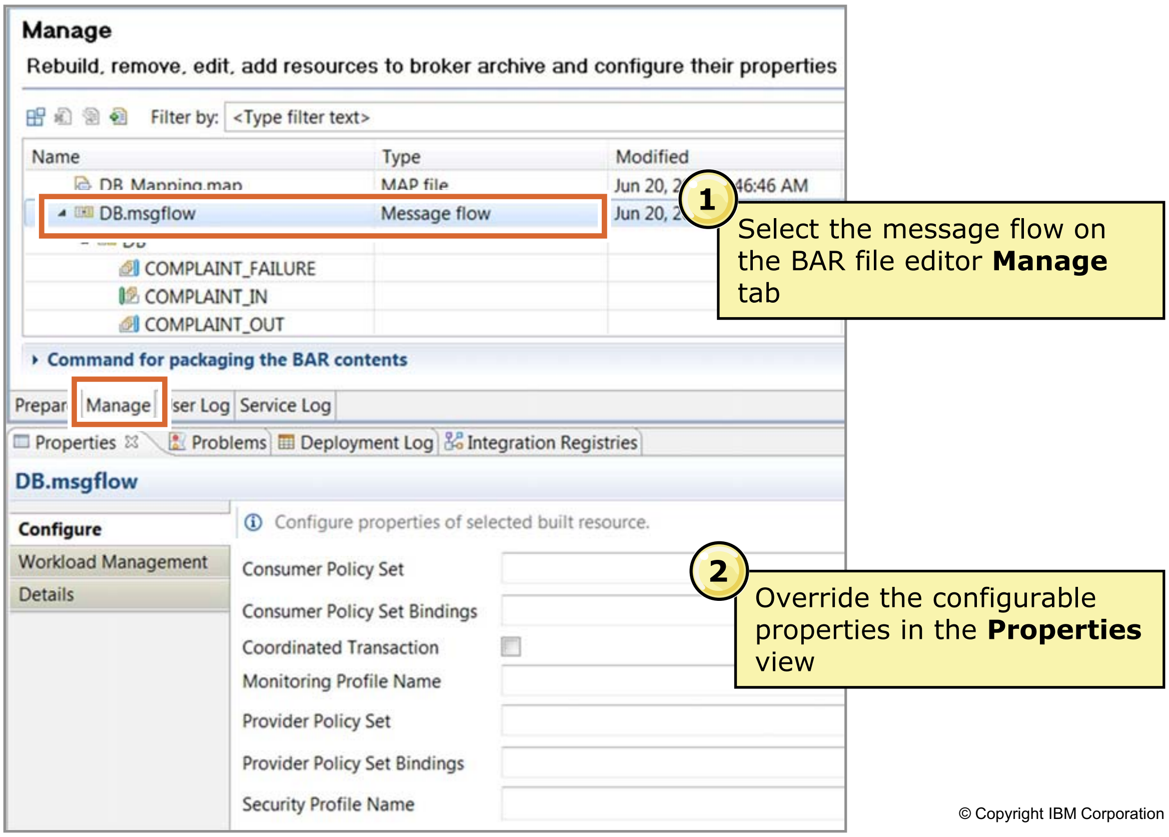
   Go to the **Manage** tab. The **Manage** tab displays the message flows, message models, and other files that are currently in the BAR file. Use the icons to add, edit, and remove files from the BAR file.

2. Select the action. The action icons on the toolbar in the **Manage** tab are used to:

   - Build a BAR file.
   - Remove message flows, message definitions, or other items from the BAR file.
   - Edit selected message flows, message definitions, or other items in the BAR file.
   - Add message flows, message definitions, or other items to the BAR file.

WebSphere Education

BAR file message flow properties



1 Select the message flow on the BAR file editor **Manage** tab

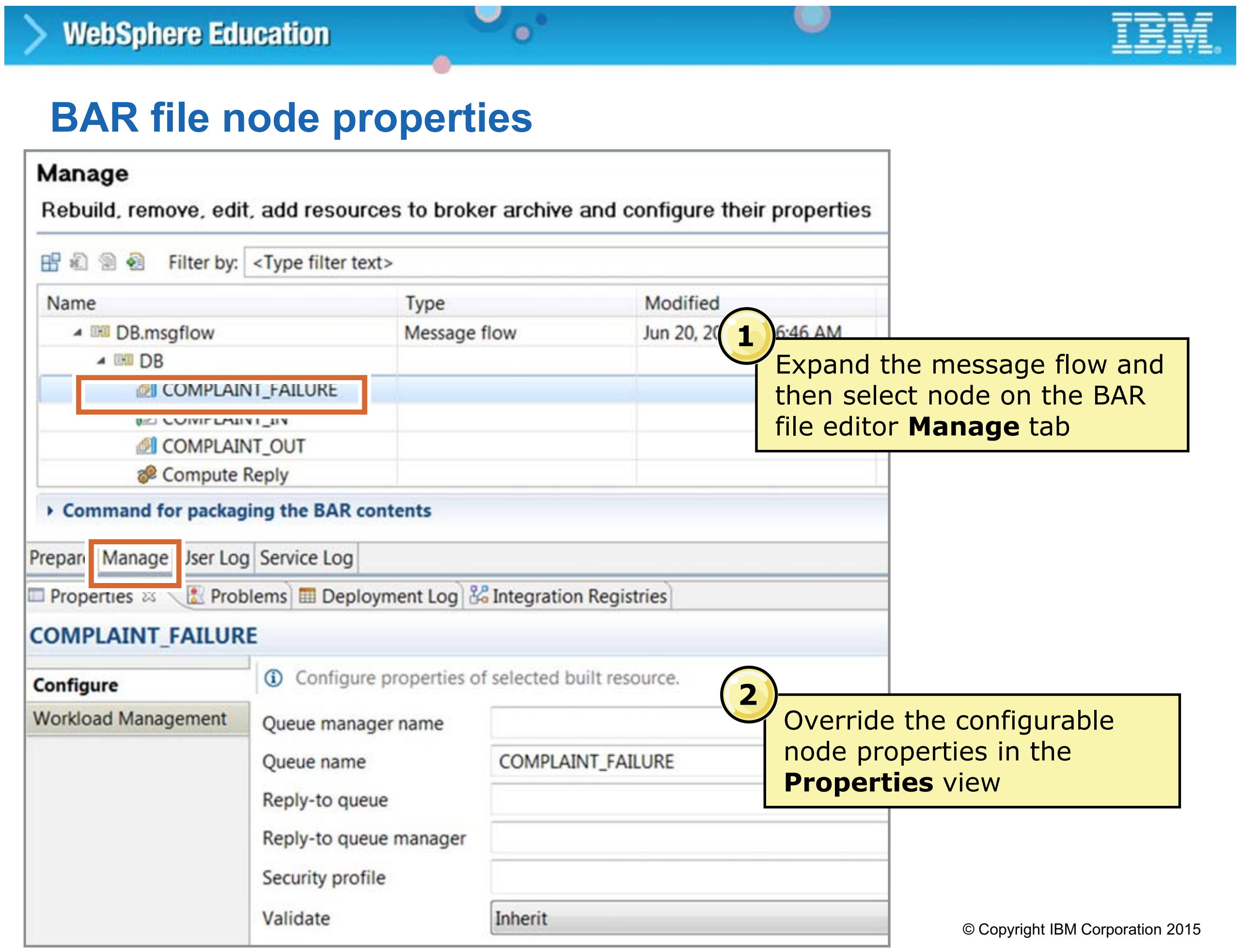
2 Override the configurable properties in the **Properties** view

Figure 4-33. BAR file message flow properties WM6461.0

Notes:

In the **Manage** view of the BAR File editor, you can set variables that give the BAR file more or alternative runtime properties. In this way, the content of the BAR file (that is, message flows and message definitions) can remain unchanged but the attributes can vary from one package to another.

The properties that can be modified without changing the underlying message flow itself are known as *configurable properties*. Configurable properties eliminate the need to modify message flow source as the flow progresses through the lifecycle from development to test to production.



BAR file node properties

Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
DB.msgflow	Message flow	Jun 20, 20 6:46 AM
DB		
COMPLAINT_FAILURE		
COMPLAINT_IN		
COMPLAINT_OUT		
Compute Reply		

Command for packaging the BAR contents

COMPLAINT_FAILURE

Configure properties of selected built resource.

Queue manager name	
Queue name	COMPLAINT_FAILURE
Reply-to queue	
Reply-to queue manager	
Security profile	
Validate	Inherit

1 Expand the message flow and then select node on the BAR file editor **Manage** tab

2 Override the configurable node properties in the **Properties** view

Figure 4-34. BAR file node properties WM6461.0

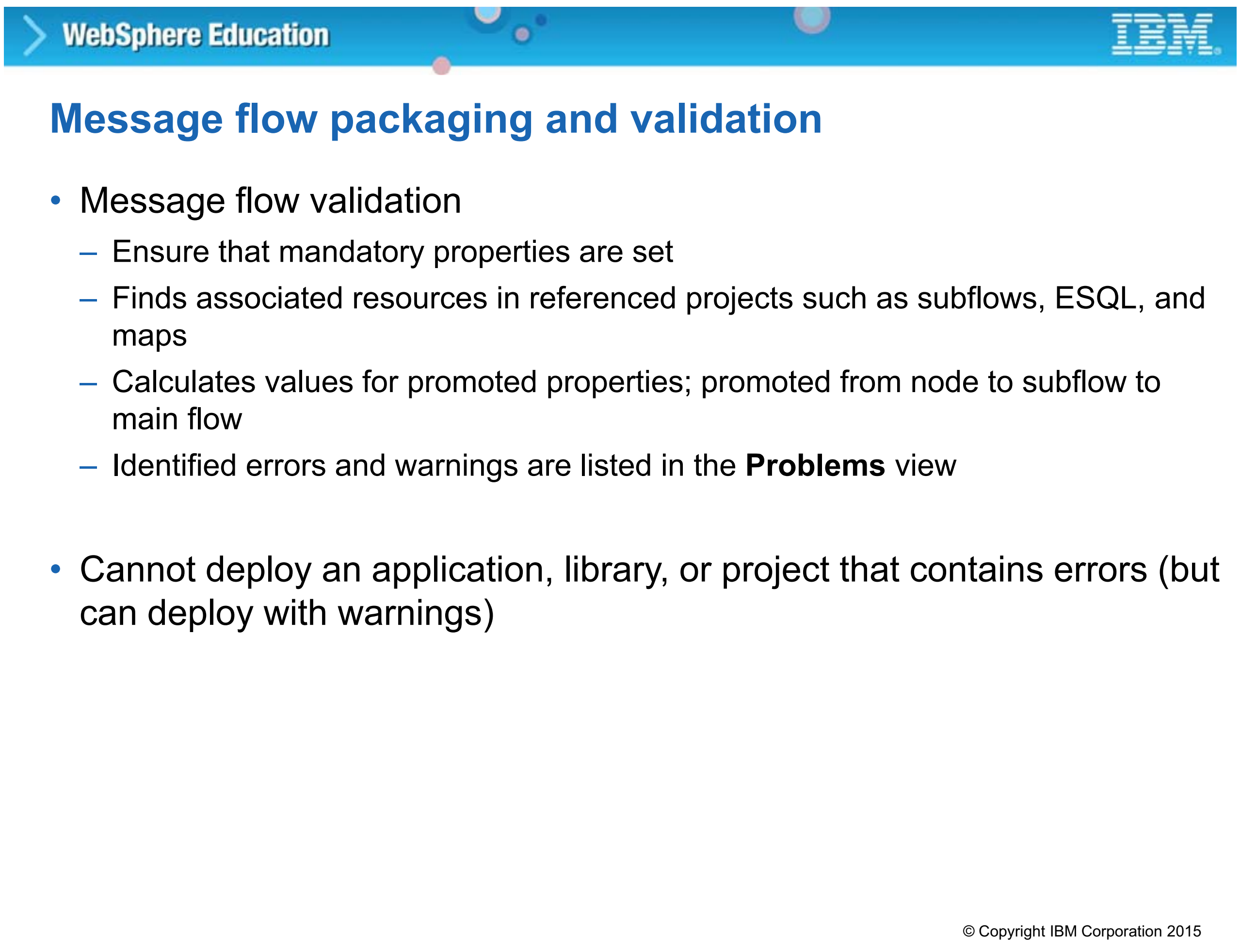
Notes:

You can also use the BAR file editor to modify some message processing node properties.

Consider the case of a simple message flow. It uses an input queue with a development name of Q_IN. On the production system, the corresponding input queue might be named COMPLAINT_FAILURE. On the **Manage** tab of the BAR file editor, right-clicking the MQInput node name displays the node properties where you can override the associated input queue. In this manner, an administrator can alter the message flow by replacing the input queue that the developer uses (Q_IN) with the queue name in the production system (COMPLAINT_FAILURE).

You can also view the properties that can be configured for your message flows by selecting **Configurable properties** from the **Filter by** list.

Be sure to save the BAR file after you change it.



Message flow packaging and validation

- Message flow validation
  - Ensure that mandatory properties are set
  - Finds associated resources in referenced projects such as subflows, ESQL, and maps
  - Calculates values for promoted properties; promoted from node to subflow to main flow
  - Identified errors and warnings are listed in the **Problems** view
- Cannot deploy an application, library, or project that contains errors (but can deploy with warnings)

© Copyright IBM Corporation 2015

Figure 4-35. Message flow packaging and validation WM6461.0

Notes:

The message flow is validated when you click **Build and Save** in the BAR file editor. Every resource is subject to incremental validation upon saving. Errors or warnings are detailed in the **Problems** view. In the **Application Development** view, you can also see the error chain up to the application level. Errors in an application prevent deployment.

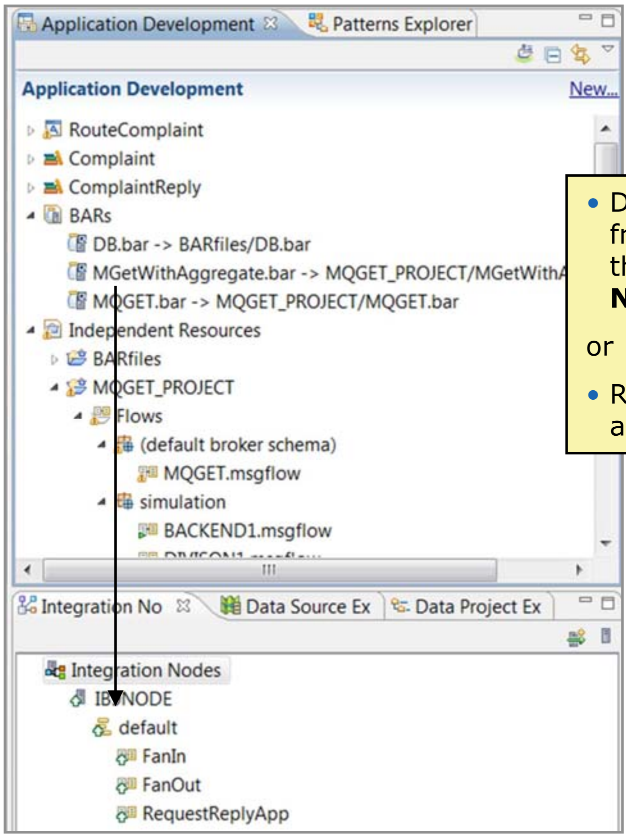
Double-clicking the error in the task view often takes the user to the location of the problem so that it can be fixed. The Task list can be filtered, for example, to show items from selected resource only.

The validator uses application properties to find dependent applications and libraries. They also find associated resources such as ESQL modules and maps. The compiler lists its results in the **Details** section of **Add to BAR** operation.

The external source code management system can add the version number, or you can use the utility `mqsicreatebar`. As an alternative to adding version information to the name of a resource, you can use the **Version** property of message flows and message sets.

WebSphere Education

Deploying in the IBM Integration Toolkit



- Drag the BAR file, application, or library from Application Development view onto the integration server in **Integration Nodes** view

or

- Right-click the BAR file, click **Deploy**, and specify the target integration server

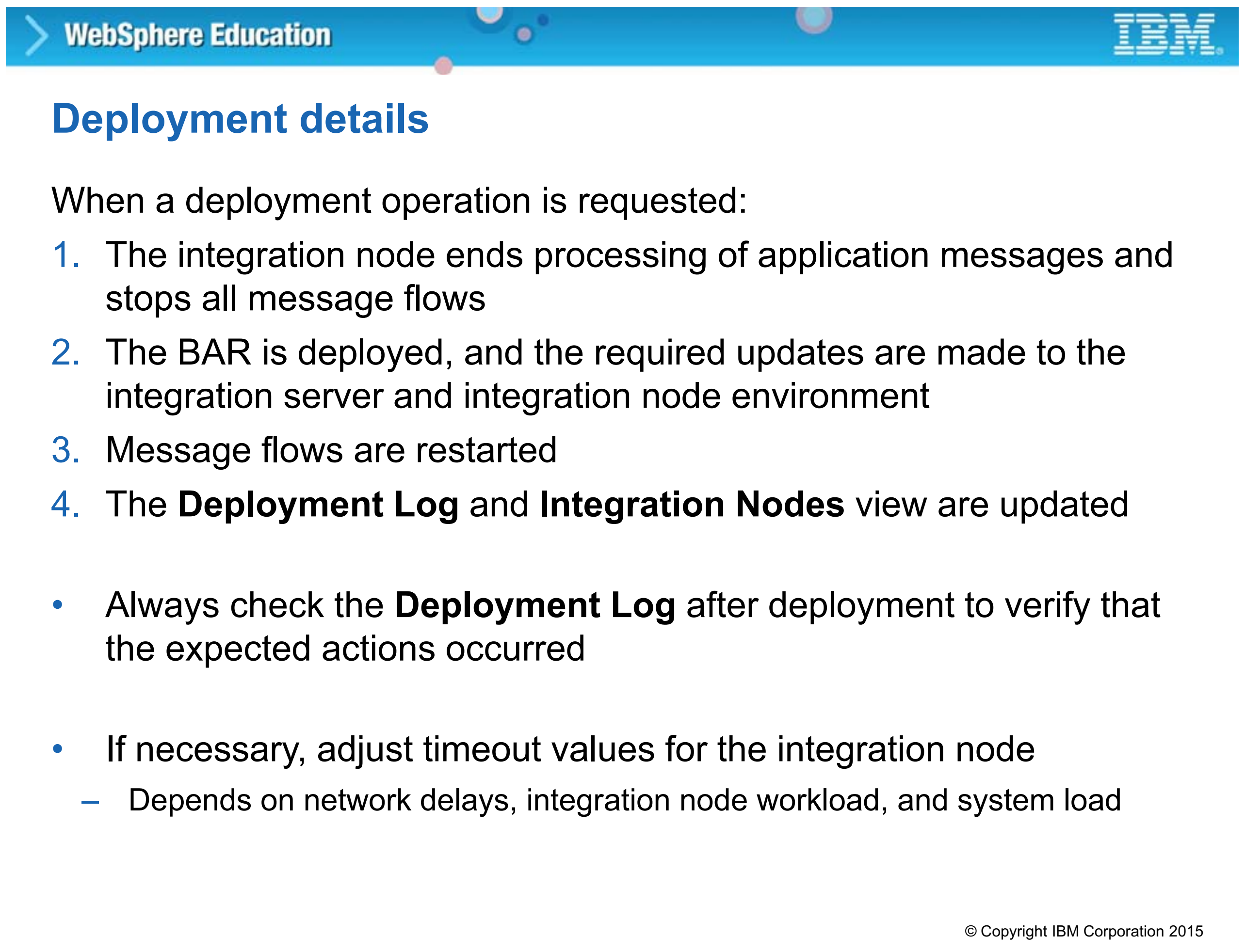
Figure 4-36. Deploying in the IBM Integration Toolkit WM6461.0

Notes:

There are many ways to deploy the BAR file to an integration server in the Integration Toolkit. One way is to select the BAR file in the **Application Development** navigator and drag it onto the integration server in the **Integration nodes** view. Alternatively, you can right-click an integration server to select a BAR file to deploy to the selected integration server.

You can also deploy message flow applications directly onto integration servers without having to build BAR files or change perspectives.

Deployment results are displayed synchronously in a **Deployment log**. With the **Deployment Log**, you can quickly verify that message flows are deployed and working as expected.



WebSphere Education

Deployment details

When a deployment operation is requested:

1. The integration node ends processing of application messages and stops all message flows
2. The BAR is deployed, and the required updates are made to the integration server and integration node environment
3. Message flows are restarted
4. The **Deployment Log** and **Integration Nodes** view are updated

- Always check the **Deployment Log** after deployment to verify that the expected actions occurred

- If necessary, adjust timeout values for the integration node
  - Depends on network delays, integration node workload, and system load

Figure 4-37. Deployment details WM6461.0

Notes:

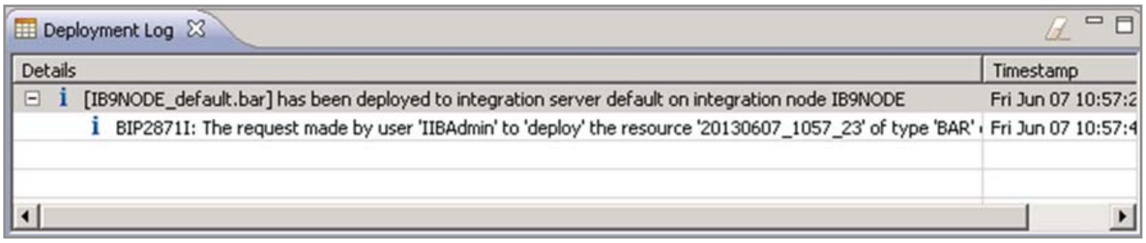
The figure lists the actions that occur when a BAR file is deployed to an integration server.

Be sure to check the **Deployment Log** in the Integration Toolkit or the **Administration Log** in the Integration web user interface after every deployment operation to ensure that the deployment was successful. Deployment messages are also shown in the system event log.

WebSphere Education IBM

Deployment Log

- Shows the result of deployment actions on an integration node
  - Name of the integration node from which deployment message originated
  - Detail of the deployment message
- **Clean Log** clears all log entries from the **Deployment Log** view, but does not delete the log entries from the administration log
- Messages are automatically cleared after 72 hours



Deployment Log

Details	Timestamp
[IB9NODE_default.bar] has been deployed to integration server default on integration node IB9NODE	Fri Jun 07 10:57:2
BIP2871I: The request made by user 'IIBAdmin' to 'deploy' the resource '20130607_1057_23' of type 'BAR'	Fri Jun 07 10:57:4

Figure 4-38. Deployment Log WM6461.0

Notes:

The **Deployment Log** in the Integration Toolkit shows the result of a deployment action on an integration node. The figure shows an example of a **Deployment Log**.

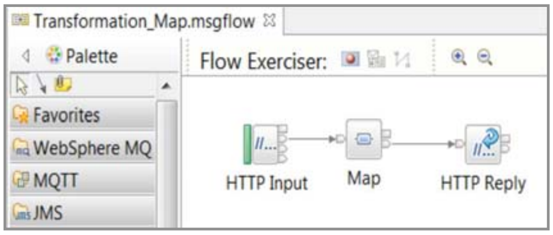
If you close the **Deployment Log** view, you can reopen it from the toolbar by clicking **Window > Show View > Deployment Log.**

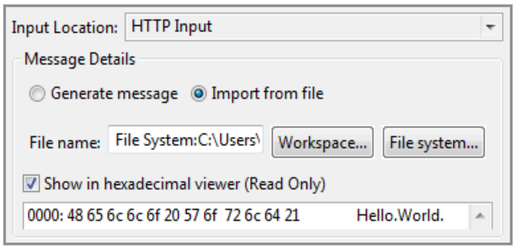
4.4. Testing message flows

WebSphere Education

Integrated testing with IBM Integration Toolkit

- Automatically creates and deploys the BAR file to selected integration server
- Client and direct injection options
  - Import, view, and edit test data
  - Inject messages over transports
  - Capture mock inputs for later replay
  - Build regression suites from test cases
- Observe captured data paths
  - Move back and forth
  - View all parts of the message assembly
  - View, start, and stop recording the data
  - Select from multiple injected messages in single data capture session




Input Location: HTTP Input

Message Details

Generate message [✓ Import from file]

File name: File System:C:\Users\ Workspace... File system...

☑ Show in hexadecimal viewer (Read Only)

```
0000: 48 65 6c 6c 6f 20 57 6f  72 6c 64 21          Hello.World.
```

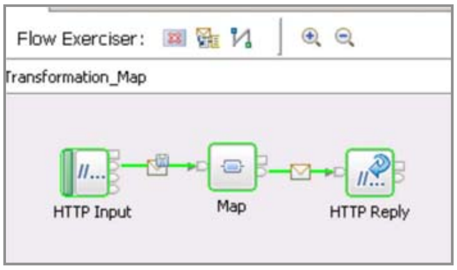


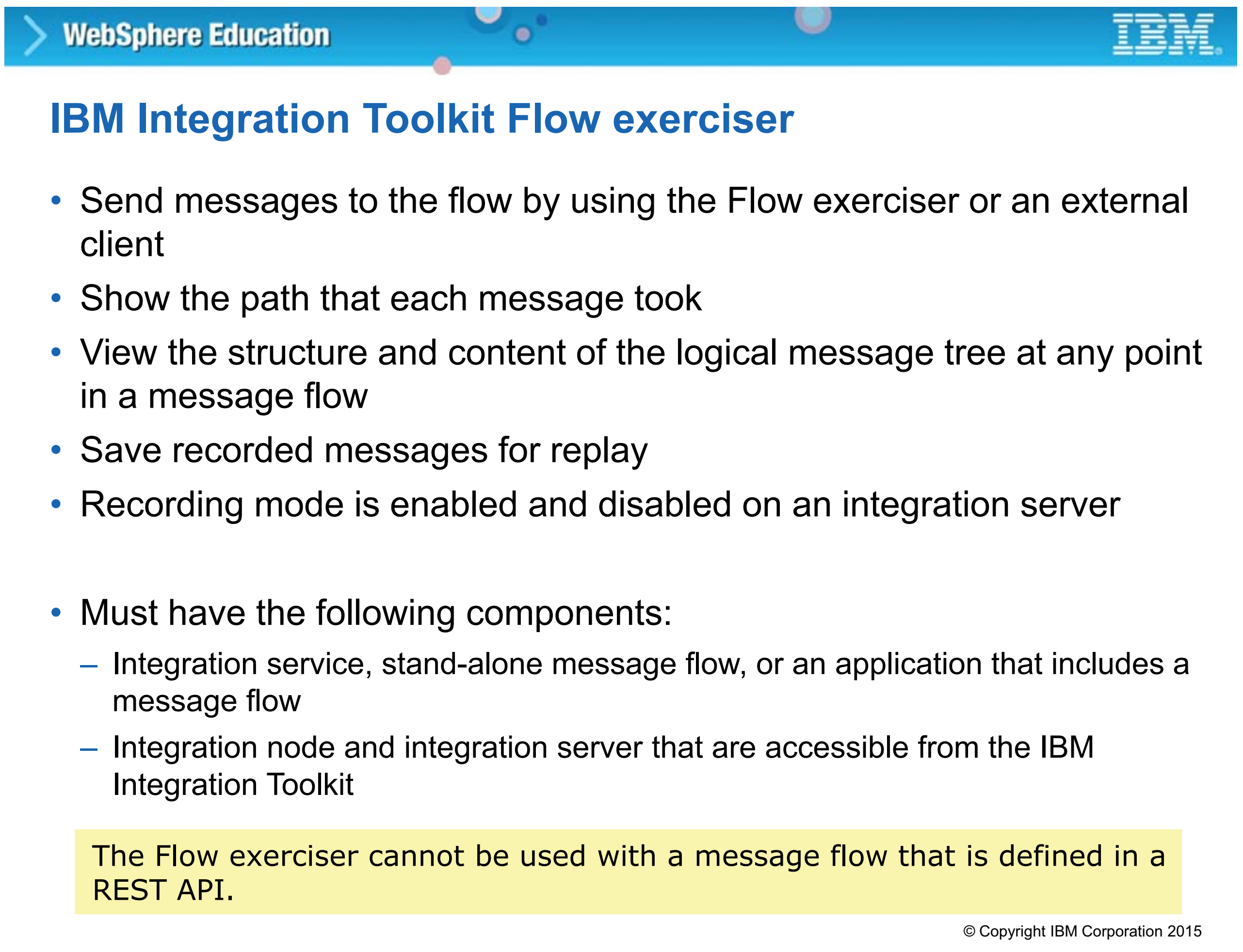
Figure 4-39. Integrated testing with IBM Integration Toolkit WM6461.0

Notes:

To check that a message flow or integration service is processing messages as expected, you can send messages to the flow by using the Flow exerciser in the Integration Toolkit. You can then use the Flow exerciser to show the path that each message took, and view the structure and content of the logical message tree at any point in a message flow.

You can send messages to the message flow by using one of the following options:

- Use an external tool or client to form and send one or more input messages to the flow. After the flow processes the input messages, click the **View path** icon in the Flow exerciser toolbar to highlight message paths on the flow.
- If you are using an integration service, or your message flow contains MQInput, HTTPInput, or SOAPInput nodes, click the **Send message** icon in the Flow exerciser toolbar. Then, you can use the **Send Message** dialog box to create an input message or select an existing input message or recorded message and send it to the flow.



WebSphere Education IBM.

IBM Integration Toolkit Flow exerciser

- Send messages to the flow by using the Flow exerciser or an external client
- Show the path that each message took
- View the structure and content of the logical message tree at any point in a message flow
- Save recorded messages for replay
- Recording mode is enabled and disabled on an integration server

- Must have the following components:
  - Integration service, stand-alone message flow, or an application that includes a message flow
  - Integration node and integration server that are accessible from the IBM Integration Toolkit

The Flow exerciser cannot be used with a message flow that is defined in a REST API.

© Copyright IBM Corporation 2015

Figure 4-40. IBM Integration Toolkit Flow exerciser WM6461.0

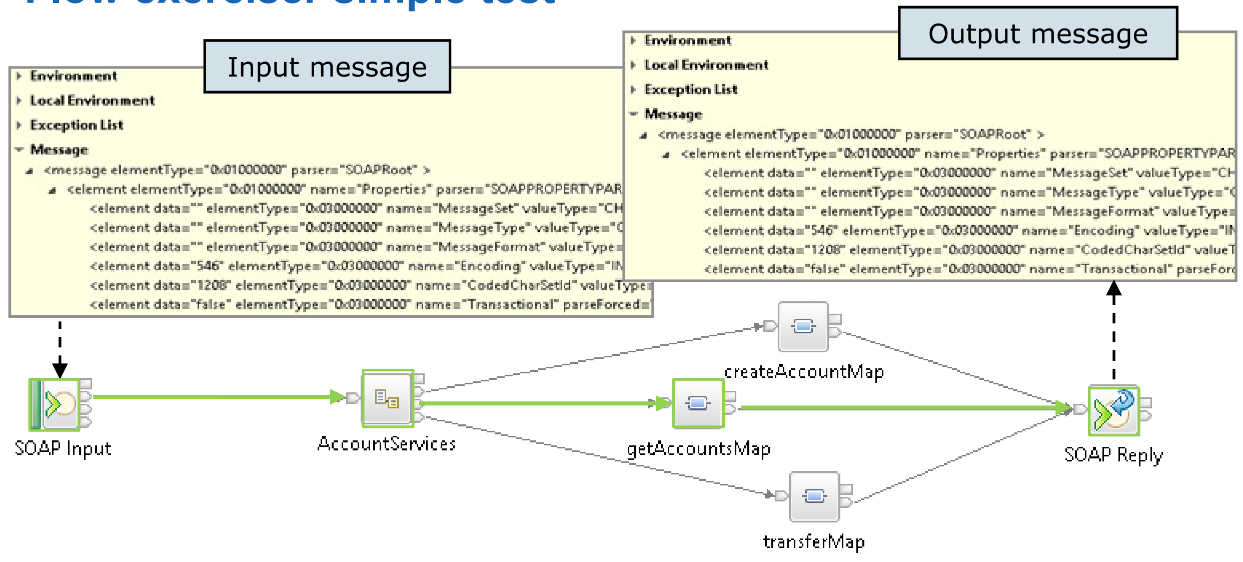
Notes:

Before you use the Flow exerciser, ensure that you have the following components:

- A resource such as an integration service, a stand-alone message flow, or an application that includes a message flow.
- An integration node and associated integration server that are accessible from the Integration Toolkit.

WebSphere Education

Flow exerciser simple test



```
Environment
Local Environment
Exception List
Message
  <message elementType="0x01000000" parser="SOAPRoot" >
    <element elementType="0x01000000" name="Properties" parser="SOAPPROPERTYPAR
      <element data="" elementType="0x03000000" name="MessageSet" valueType="CH
      <element data="" elementType="0x03000000" name="MessageType" valueType="C
      <element data="" elementType="0x03000000" name="MessageFormat" valueType=
      <element data="546" elementType="0x03000000" name="Encoding" valueType="IN
      <element data="1208" elementType="0x03000000" name="CodedCharSetId" valueType=
      <element data="false" elementType="0x03000000" name="Transactional" parseForced="
```

```
Environment
Local Environment
Exception List
Message
  <message elementType="0x01000000" parser="SOAPRoot" >
    <element elementType="0x01000000" name="Properties" parser="SOAPPROPERTYPAR
      <element data="" elementType="0x03000000" name="MessageSet" valueType="CH
      <element data="" elementType="0x03000000" name="MessageType" valueType="C
      <element data="" elementType="0x03000000" name="MessageFormat" valueType=
      <element data="546" elementType="0x03000000" name="Encoding" valueType="IN
      <element data="1208" elementType="0x03000000" name="CodedCharSetId" valueT
      <element data="false" elementType="0x03000000" name="Transactional" parseFor
```

1. Create and send a message to the input node of the message flow
2. Highlight the message path on the message flow and any subflows that are associated with the message flow
3. Display the content of the logical message tree for a message that passed through a connection in the message flow
4. Save the content of the logical message tree as a recorded message that you can send to the message flow later

Figure 4-41. Flow exerciser simple test WM6461.0

Notes:

This figure lists the basic steps for using the Flow exerciser for testing a message flow.

After you record the run of a flow (as shown by the highlighted paths in green), you can save any input messages for testing the flow later. When you send (inject) a message to the flow, the flow is initiated directly; it does not need an external transport to send an input message to the node.

Saved messages use the internal XML format, and can be viewed under **Other Resources** in the **Application Development** view. The internal XML format for any recorded message can also be viewed by clicking **Copy unformatted message** from the menu.

WebSphere Education

Flow exerciser toolbar

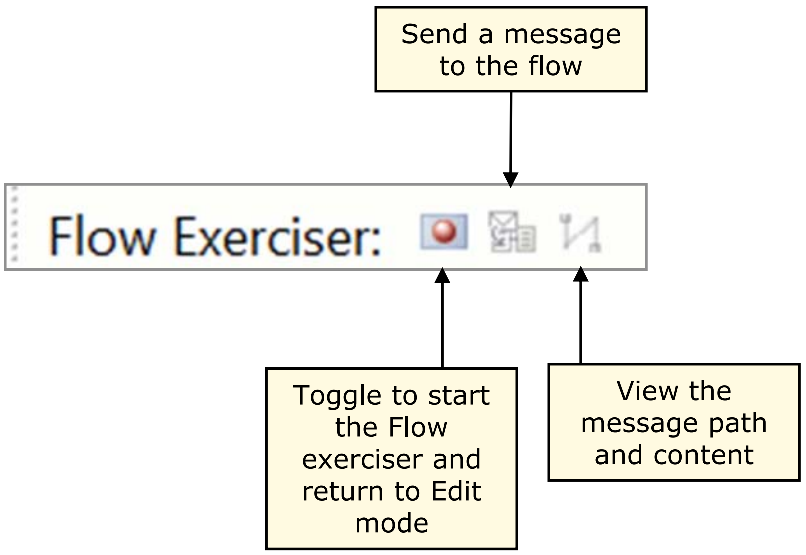

Send a message to the flow
Flow Exerciser:
Toggle to start the Flow exerciser and return to Edit mode
View the message path and content

Figure 4-42. Flow exerciser toolbar WM6461.0

Notes:

The Flow exerciser toolbar in the Message Flow editor, contains three icons.

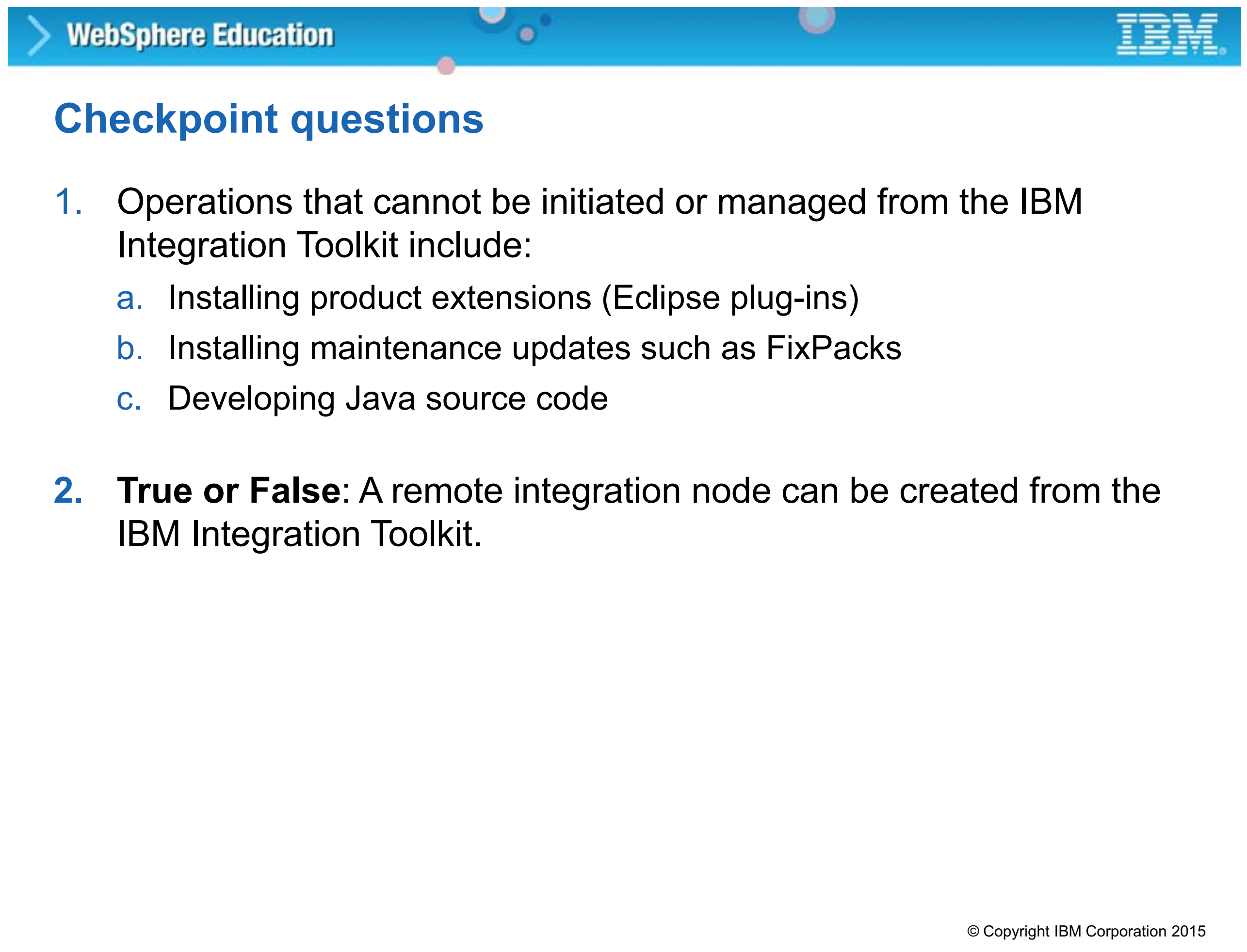
Unit summary

Having completed this unit, you should be able to:

- Explain the function of the IBM Integration Toolkit
- Use the IBM Integration Toolkit BAR File editor to create and modify a BAR file
- Deploy a message flow application and shared libraries
- Use the IBM Integration Toolkit Flow exerciser to test a deployed message flow application

Figure 4-43. Unit summary WM6461.0

Notes:



WebSphere Education IBM

Checkpoint questions

1. Operations that cannot be initiated or managed from the IBM Integration Toolkit include:
   a. Installing product extensions (Eclipse plug-ins)
   b. Installing maintenance updates such as FixPacks
   c. Developing Java source code

2. **True or False**: A remote integration node can be created from the IBM Integration Toolkit.

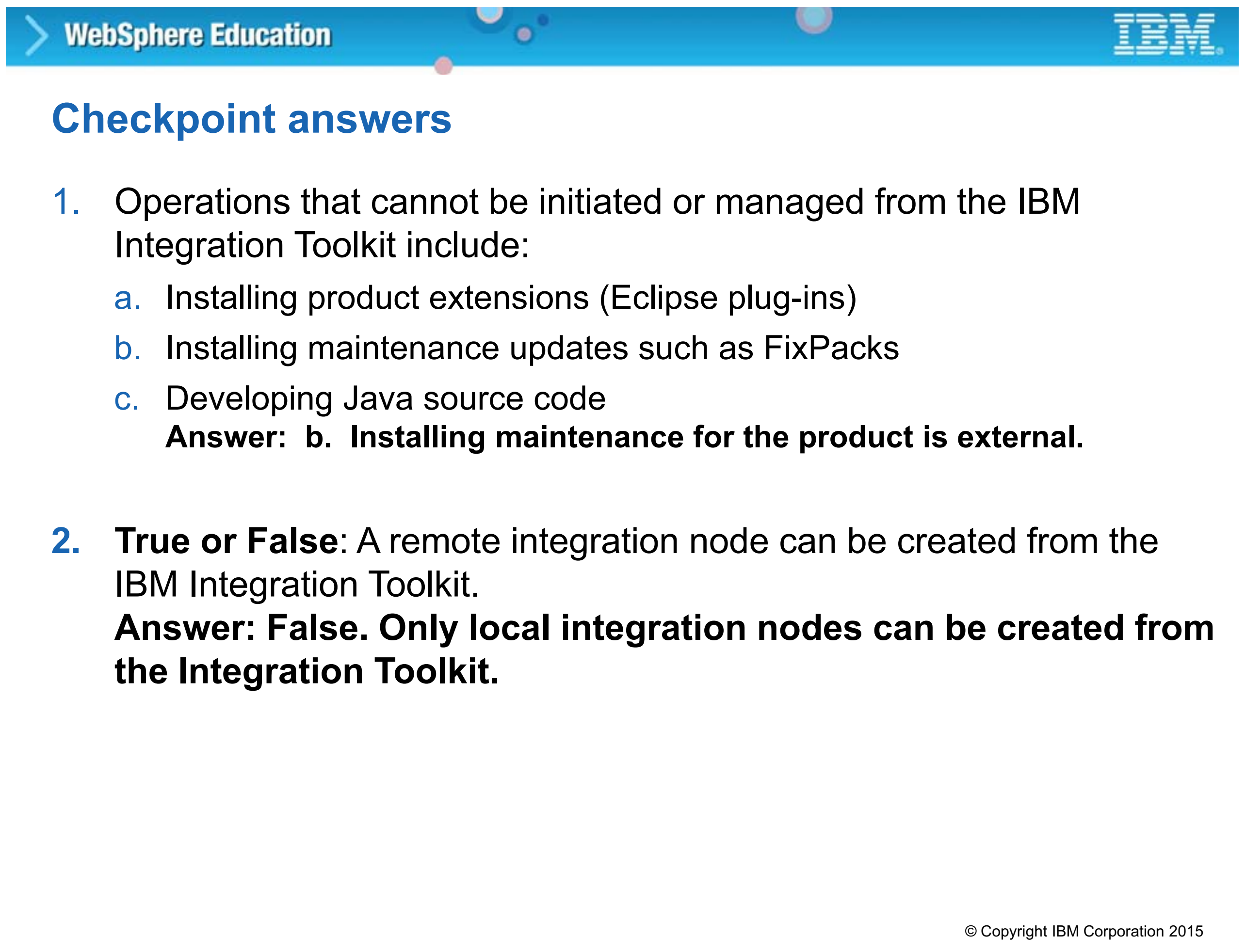
Figure 4-44. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



Checkpoint answers

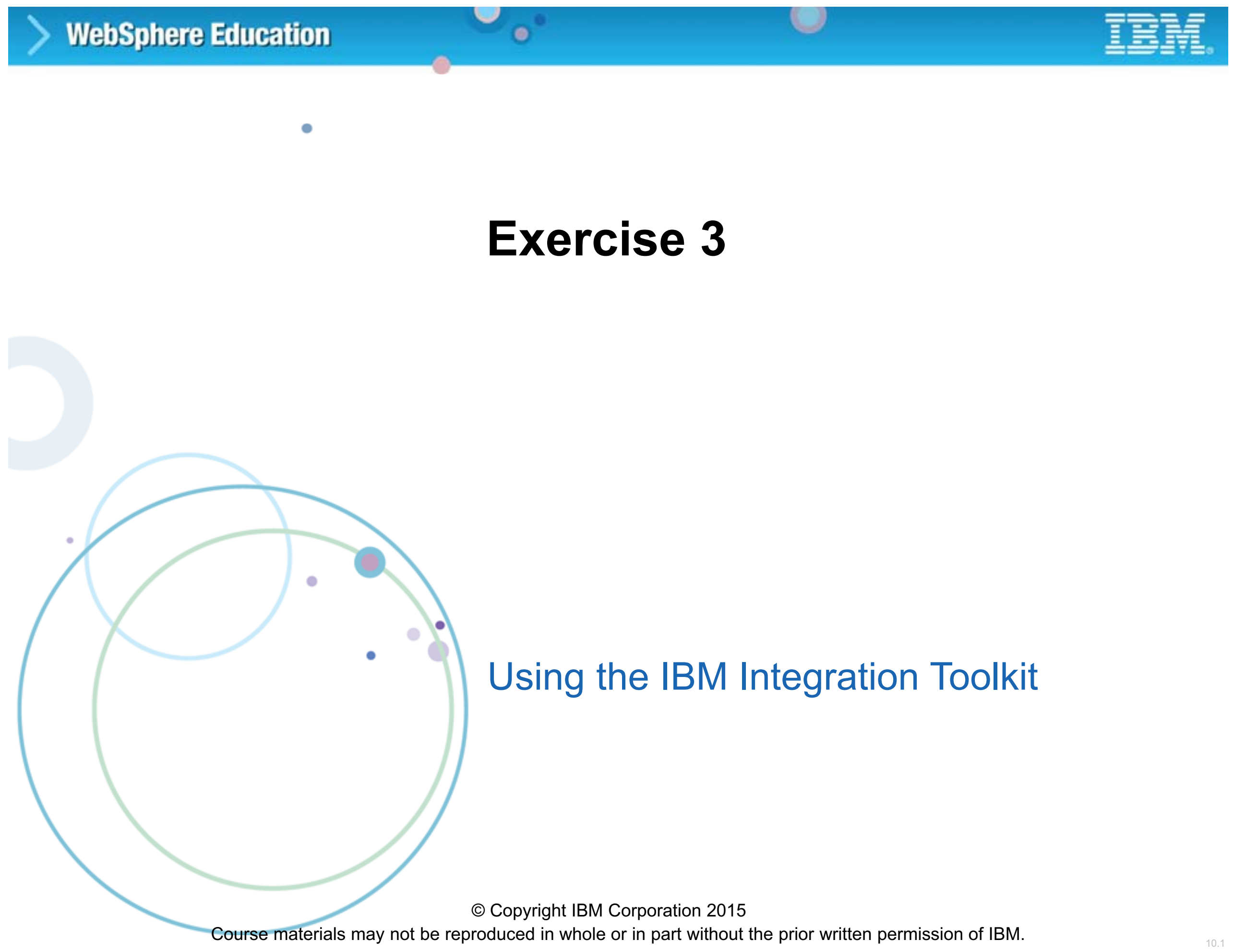
1. Operations that cannot be initiated or managed from the IBM Integration Toolkit include:
   a. Installing product extensions (Eclipse plug-ins)
   b. Installing maintenance updates such as FixPacks
   c. Developing Java source code
   Answer: b. Installing maintenance for the product is external.

2. **True or False**: A remote integration node can be created from the IBM Integration Toolkit.
   Answer: False. Only local integration nodes can be created from the Integration Toolkit.

© Copyright IBM Corporation 2015

Figure 4-45. Checkpoint answers WM6461.0

Notes:



WebSphere Education

Exercise 3

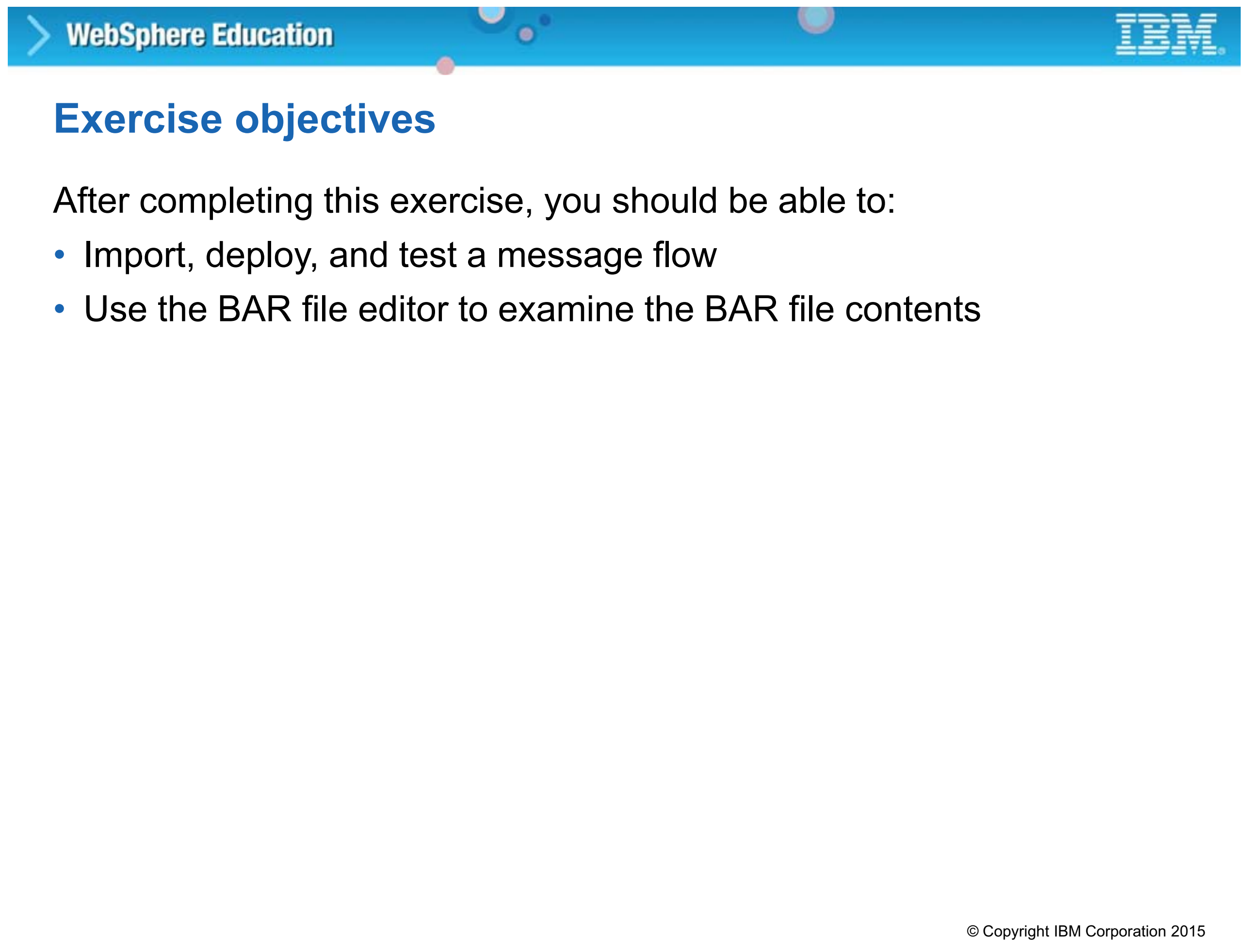
Using the IBM Integration Toolkit

10.1

Figure 4-46. Exercise 3 WM6461.0

Notes:

In this exercise, you use import, deploy, and test a message flow to learn the administrative tasks available in the development environment.


WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:
- Import, deploy, and test a message flow
- Use the BAR file editor to examine the BAR file contents

© Copyright IBM Corporation 2015

Figure 4-47. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 5. Administration basics

What this unit is about

In this unit, you learn how to use commands, the IBM Integration web user interface, and the IBM Integration API Exerciser to configure and administer the integration node and integration node components. The unit also describes how to back up and restore critical resources.

What you should be able to do

After completing this unit, you should be able to:

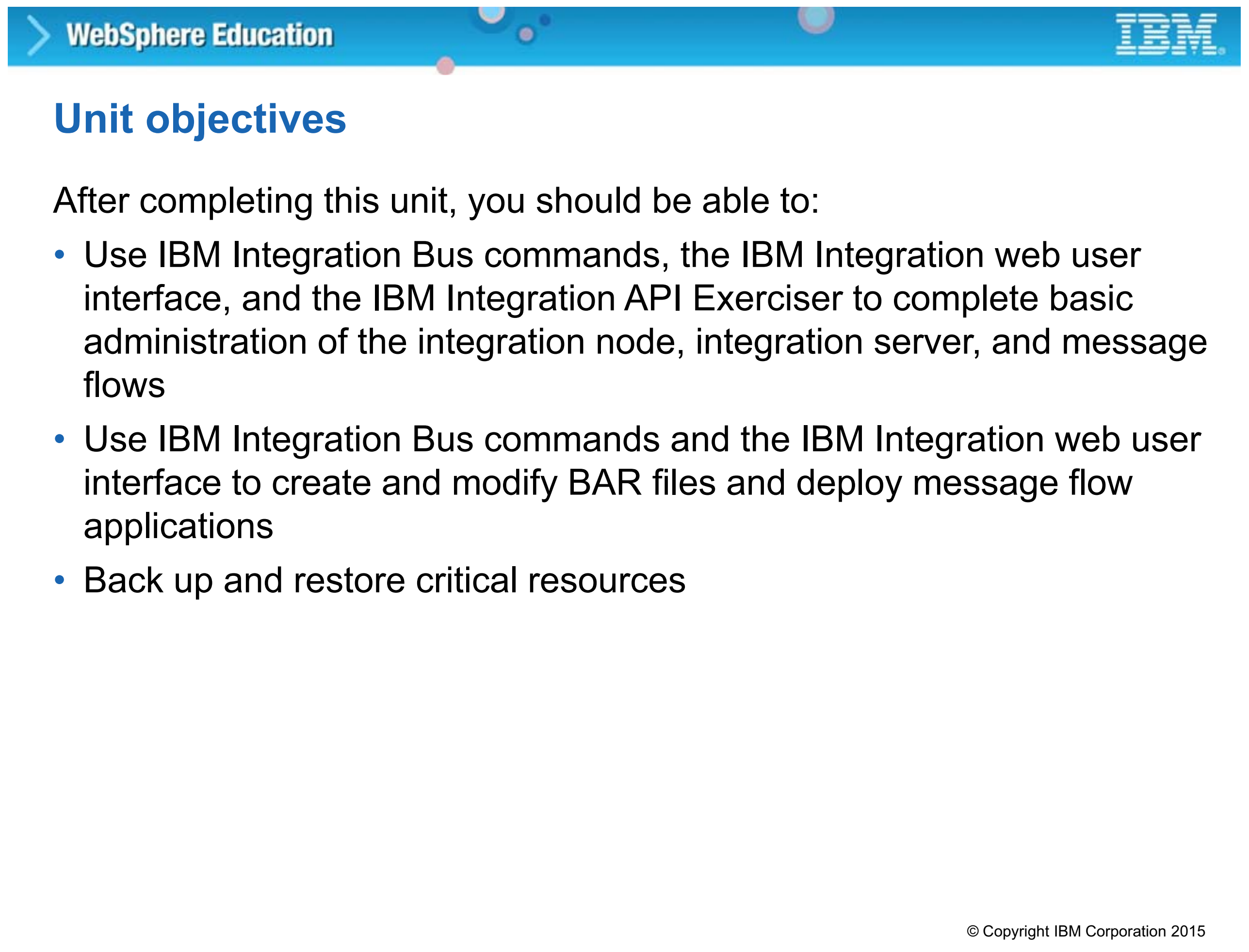
- Use IBM Integration Bus commands, the IBM Integration web user interface, and the IBM Integration API Exerciser to complete basic administration of the integration node, integration server, and message flows
- Use IBM Integration Bus commands and the IBM Integration web user interface to create and modify BAR files and deploy message flow applications
- Back up and restore critical resources

How you will check your progress

- Checkpoint questions
- Exercise 4, Administering the runtime components

References

IBM Knowledge Center



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Use IBM Integration Bus commands, the IBM Integration web user interface, and the IBM Integration API Exerciser to complete basic administration of the integration node, integration server, and message flows
- Use IBM Integration Bus commands and the IBM Integration web user interface to create and modify BAR files and deploy message flow applications
- Back up and restore critical resources

© Copyright IBM Corporation 2015

Figure 5-1. Unit objectives WM6461.0

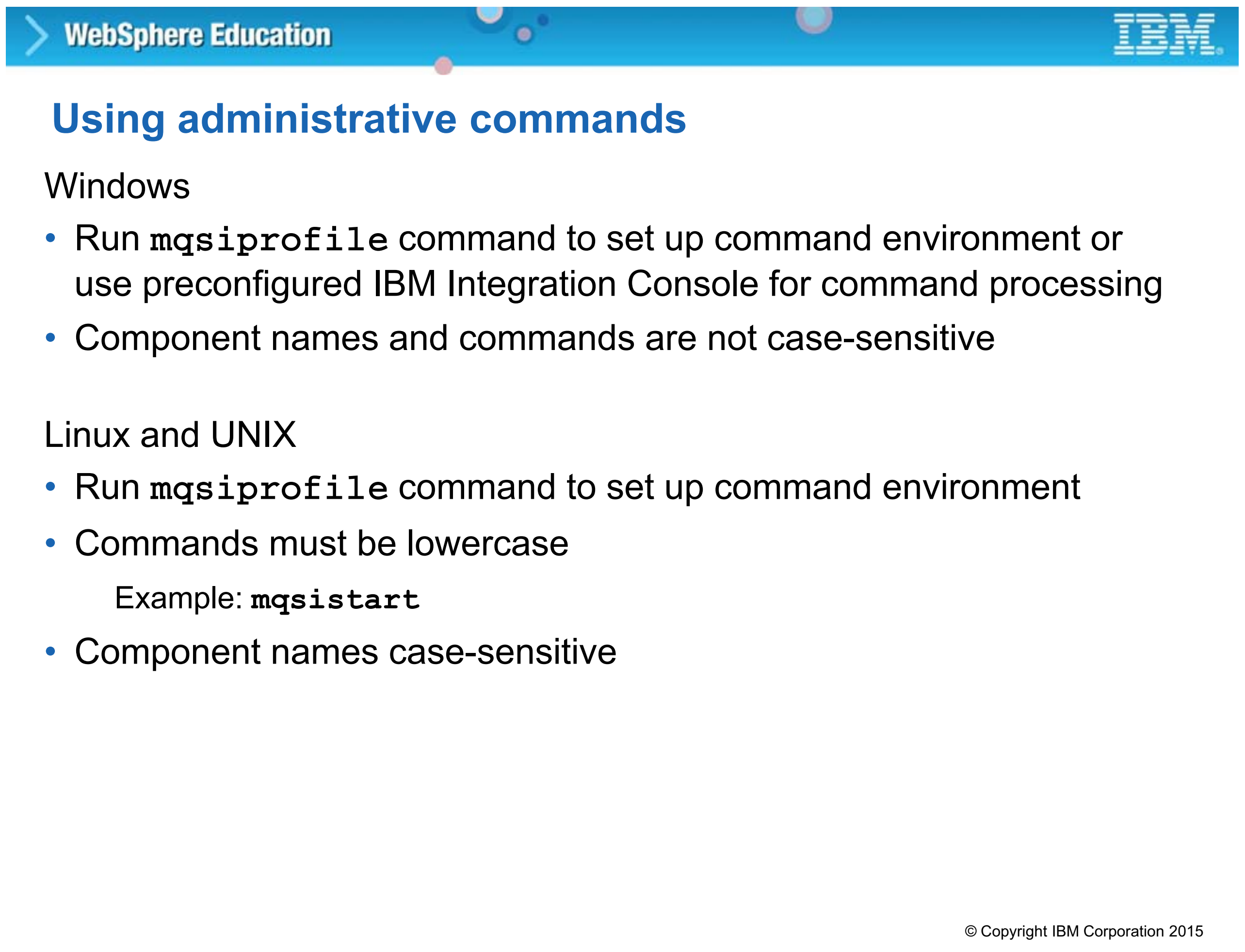
Notes:

This unit describes basic administration by using the IBM Integration Bus administration interfaces.

- With the IBM Integration Bus command console, you can use commands administer Integration Bus components.
- The IBM Integration web user interface provides a web-based administration interface to an integration node.

This unit also includes a topic on backup and restore utilities for integration node resources.

5.1. IBM Integration Bus commands


WebSphere Education
IBM

Using administrative commands

Windows

- Run **mqsiprofile** command to set up command environment or use preconfigured IBM Integration Console for command processing
- Component names and commands are not case-sensitive

Linux and UNIX

- Run **mqsiprofile** command to set up command environment
- Commands must be lowercase

  Example: **mqsistart**
- Component names case-sensitive

© Copyright IBM Corporation 2015

Figure 5-2. Using administrative commands WM6461.0

Notes:

All IBM Integration Bus functions can be controlled by using commands. These commands begin with “mqsi” or “iib”.

The IBM Integration Bus command processor is essentially the same for Windows, Linux, and UNIX. In all cases, the command environment (such as file paths and environment variables) must be initialized after product installation, and also before you run a command, such as before you start an integration node. The `mqsiprofile` command initializes the environment.

On Windows, you can start the IBM Integration Console, which runs the `mqsiprofile` command in the background, then displays a Windows command shell from which you enter the administration commands.

In the UNIX and Linux environments, you must manually run the `mqsiprofile` command to set up the command environment; you can then run administration commands.

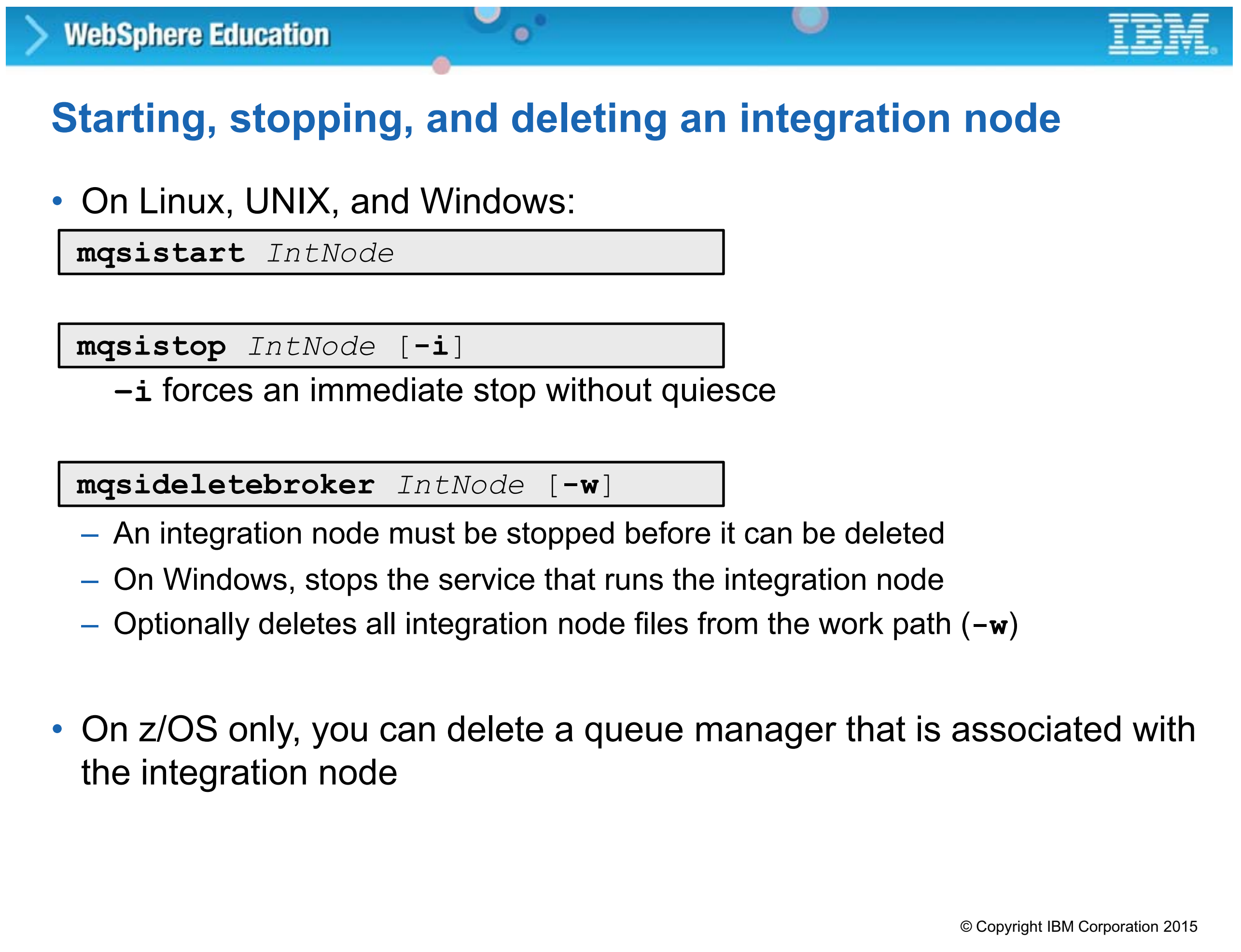
Before running the `mqsiprofile` command, be sure to change the contents of the `.profile` file by adding this line:

```
/opt/ibm/mqsi/10.0.0.0/bin/mqsiprofile
```

Warning

Commands are case-sensitive on the Linux and UNIX.

The `mqsiprofile` command is found in the `bin` directory of the installation directory. Do not change the location of the `mqsiprofile` command, since it can be replaced or updated when you apply product maintenance.



WebSphere Education

Starting, stopping, and deleting an integration node

- On Linux, UNIX, and Windows:

```
mqsistart IntNode
```

```
mqsistop IntNode [-i]
```

-i forces an immediate stop without quiesce

```
mqsideletebroker IntNode [-w]
```

  – An integration node must be stopped before it can be deleted
  – On Windows, stops the service that runs the integration node
  – Optionally deletes all integration node files from the work path (**-w**)

- On z/OS only, you can delete a queue manager that is associated with the integration node

© Copyright IBM Corporation 2015

Figure 5-3. Starting, stopping, and deleting an integration node WM6461.0

Notes:

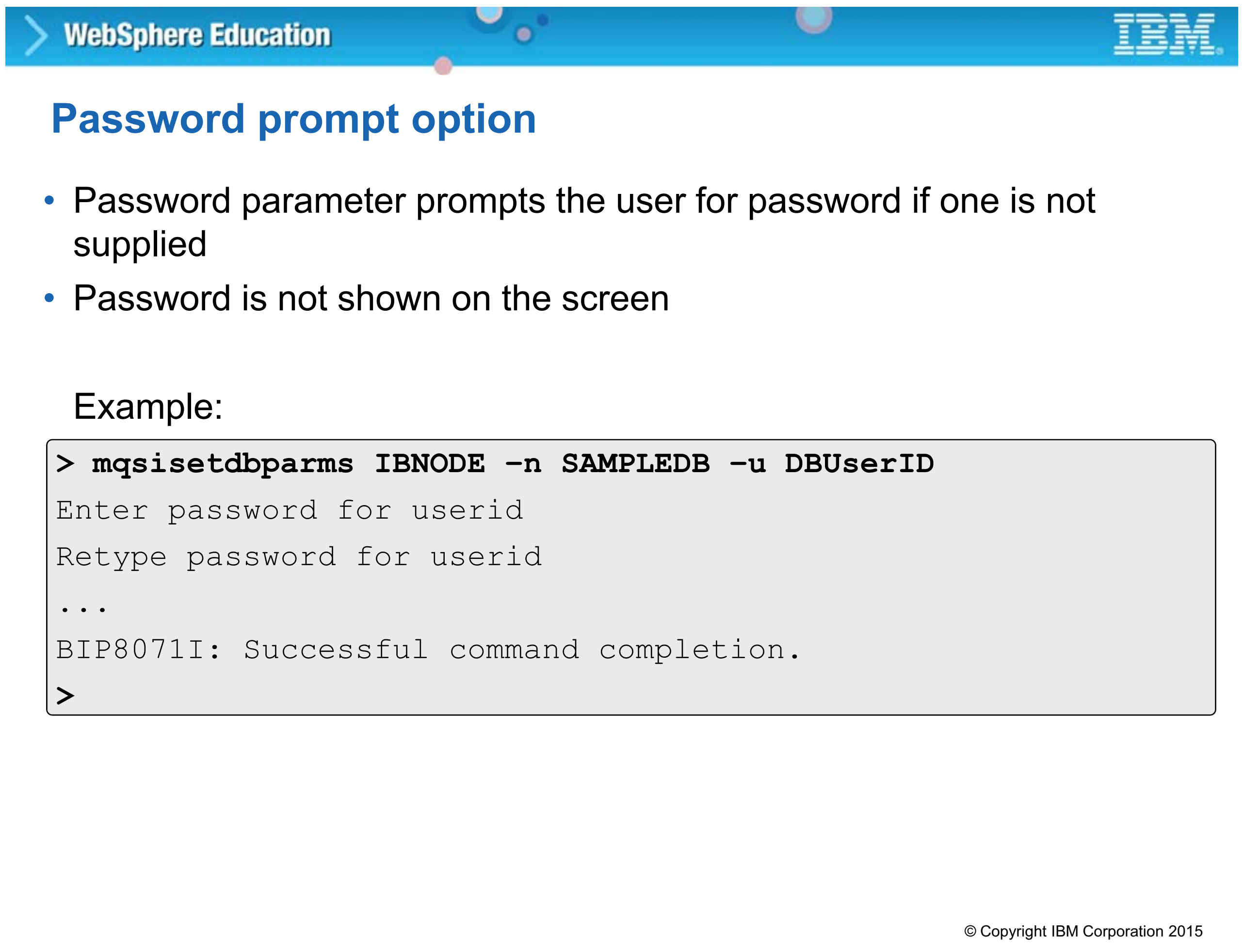
In earlier units, you learned how to create an integration node. After an integration node is created, you must know how to start and stop it. You also must know how to delete it.

The **`mqsistop -i`** option immediately stops the integration node. Specify this flag only if attempting to stop the integration node without `-i` did not stop the integration node.

With the **`mqsideletebroker`** command, you can stop and remove all work path files that are associated with the integration node (`-w`).

Be sure that the integration node is stopped before attempting to delete it. If an integration node is associated with queue manager, deleting an integration node also removes all integration node SYSTEM queues.

On Windows, the integration node service is also deleted when the integration node is deleted.



WebSphere Education

Password prompt option

- Password parameter prompts the user for password if one is not supplied
- Password is not shown on the screen

Example:

```
> mqsisetdbparms IBNODE -n SAMPLEDB -u DBUserID
Enter password for userid
Retype password for userid
...
BIP8071I: Successful command completion.
>
```

Figure 5-4. Password prompt option WM6461.0

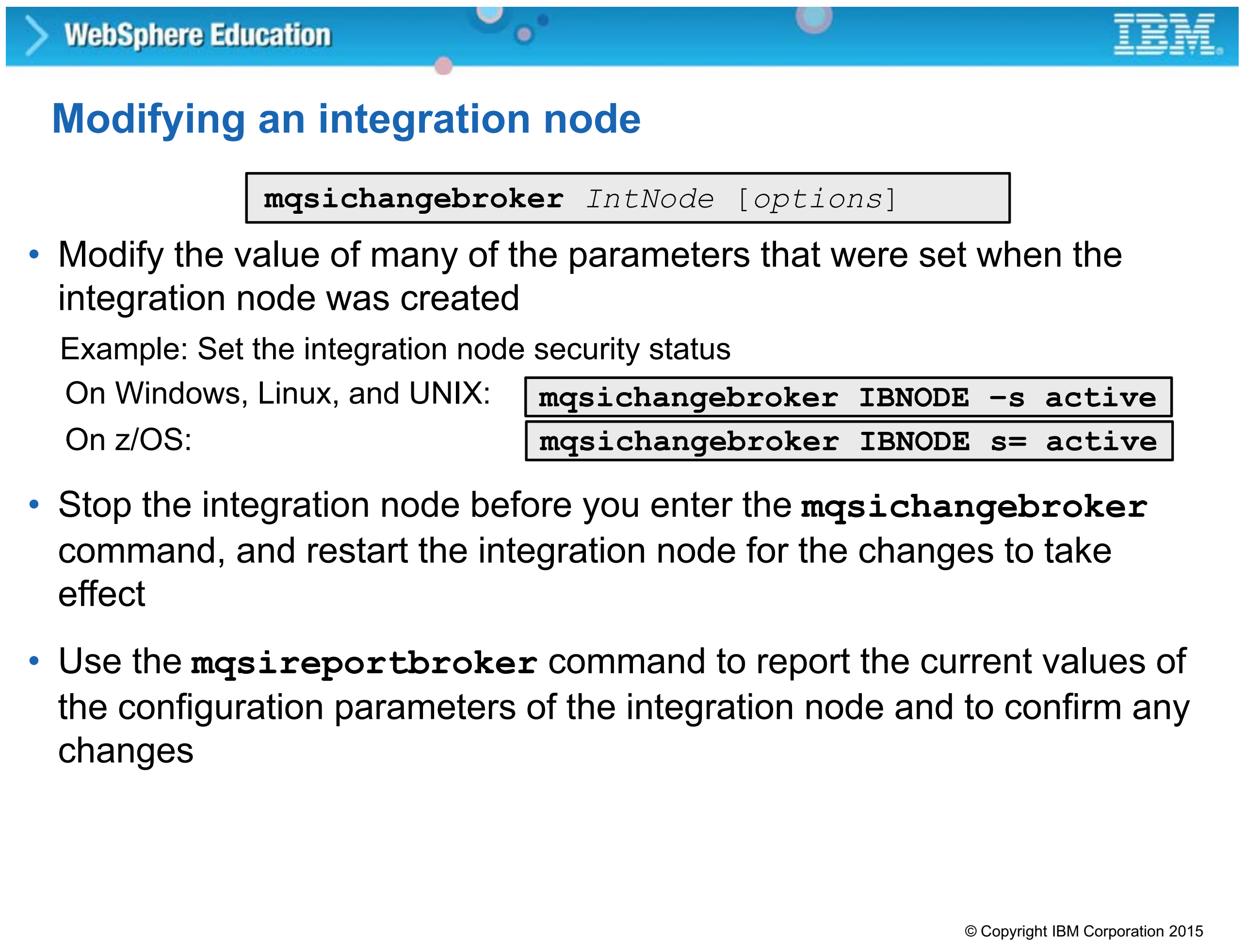
Notes:

When a command that requires a password is run without using one, you are prompted for the password.

In the example in the figure, a user ID for the database is provided (`-u DBUserID`), but a password is not provided. A prompt for the password is automatically displayed.

The response to the prompt is not displayed on the console when you type it.

Be careful when using commands that require a password in a script because the prompt cannot be responded to interactively.


WebSphere Education
IBM

Modifying an integration node

```
mqsichangebroker IntNode [options]
```

- Modify the value of many of the parameters that were set when the integration node was created

  Example: Set the integration node security status

  On Windows, Linux, and UNIX: `mqsichangebroker IBNODE -s active`

  On z/OS: `mqsichangebroker IBNODE s= active`

- Stop the integration node before you enter the **`mqsichangebroker`** command, and restart the integration node for the changes to take effect
- Use the **`mqsireportbroker`** command to report the current values of the configuration parameters of the integration node and to confirm any changes

© Copyright IBM Corporation 2015

Figure 5-5. Modifying an integration node WM6461.0

Notes:

Not all integration node properties can be changed with the `mqsichangebroker` command.

Here is a partial list of some commonly used options:

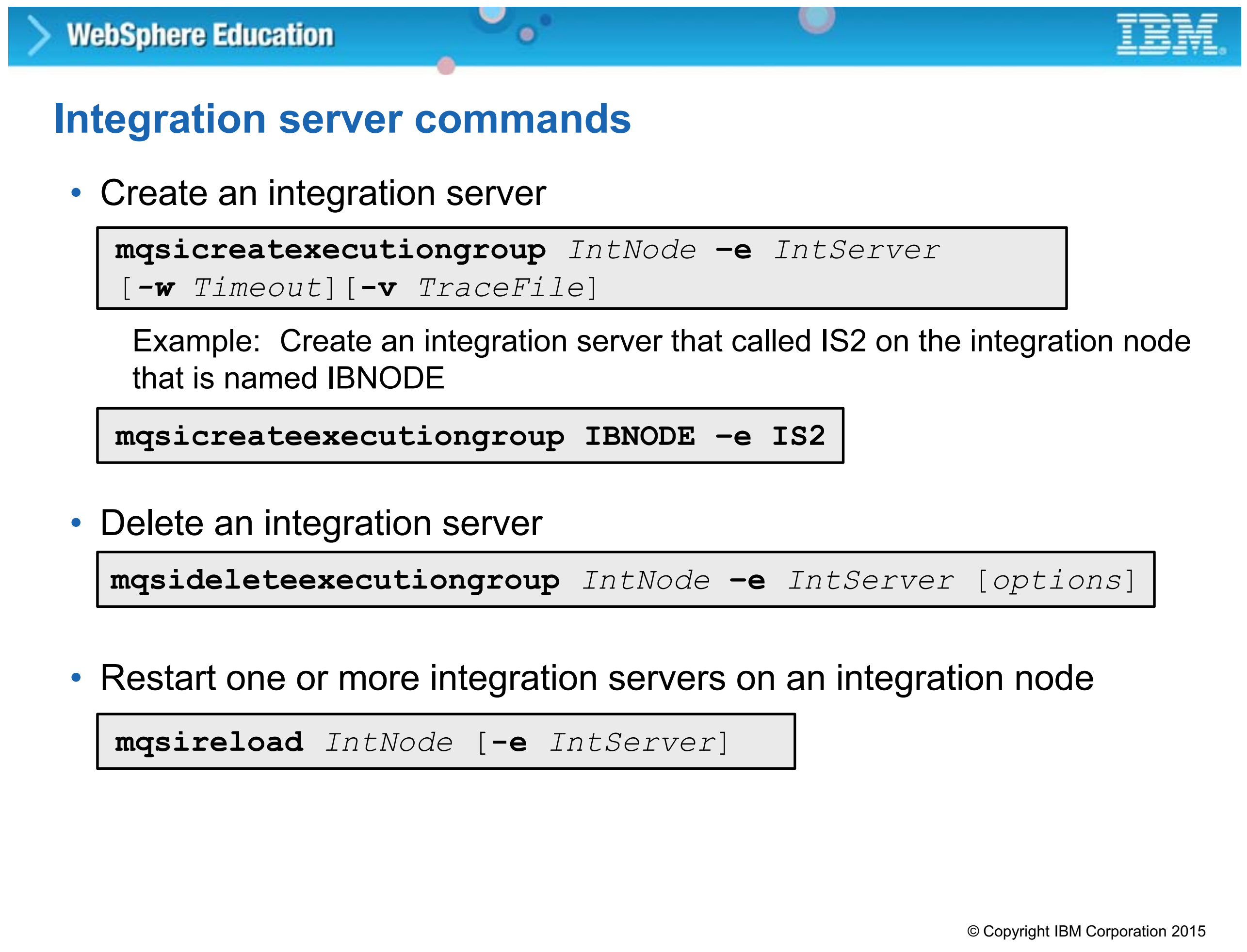
-g Maximum time to wait for a response from an integration server.

-k Maximum time to wait for a response from an integration node.

The `-g` and `-k` options are related. The total of the two values is the amount of time the integration node allows before sending a negative response. The default values are 300 seconds for `-g` and 60 seconds for `-k`.

You must stop the integration node before you enter the `mqsichangebroker` command and then restart the integration node for the changes to take effect.

- On Windows, Linux, and UNIX systems, use the **`mqsistop`** and **`mqsistart`** commands to stop and then restart the integration node.
- On z/OS, stop the integration node components by using the **`/F`** integration node PC option and restart the integration node components by using the **`/F`** integration node SC option.



WebSphere Education

Integration server commands

- Create an integration server

```
mqsicreatexecutiongroup IntNode -e IntServer
[-w Timeout][-v TraceFile]
```

Example: Create an integration server that called IS2 on the integration node that is named IBNODE

```
mqsicreateexecutiongroup IBNODE -e IS2
```

- Delete an integration server

```
mqsideleteexecutiongroup IntNode -e IntServer [options]
```

- Restart one or more integration servers on an integration node

```
mqsireload IntNode [-e IntServer]
```

Figure 5-6. Integration server commands WM6461.0

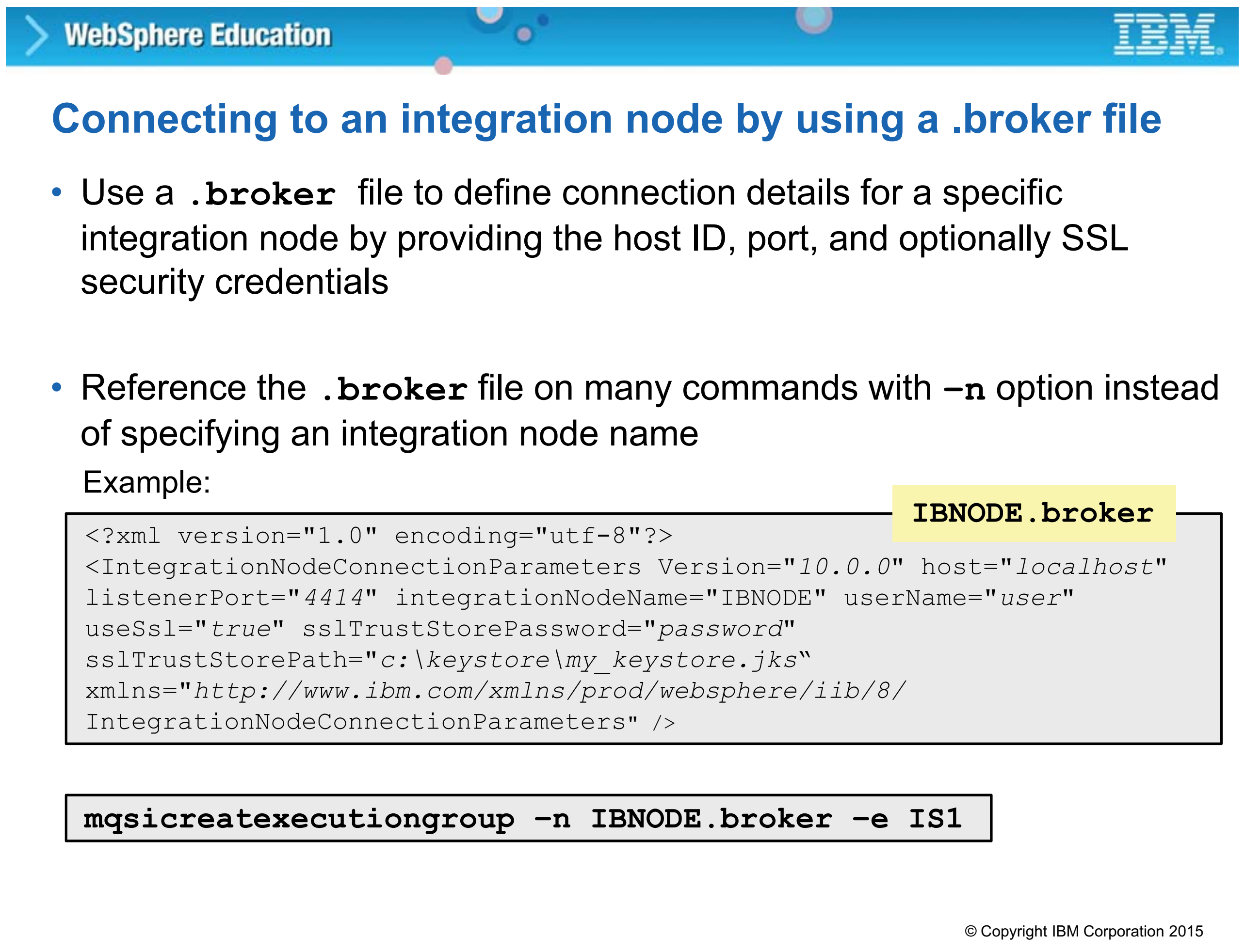
Notes:

An alternative to creating an integration server through the IBM Integration Toolkit or the IBM Integration web user interface is to use the command interface. Additionally, the required parameters and optional flags can be specified in a file, which is passed to the appropriate command by using the `-n` flag.

It is also possible to delete an integration server with the `mqsideleteexecutiongroup` command.

The `mqsireload` command sends a message to the integration node to stop and restart all its integration servers or a specific integration server, depending on the command form. You can use the `mqsireload` command to set the JVM debug port for the message flow debugger, for example, or to set content filtering on the integration node for publish/subscribe.

For a complete description of command options and syntax, see the IBM Knowledge Center for IBM Integration Bus.



Connecting to an integration node by using a .broker file

- Use a **.broker** file to define connection details for a specific integration node by providing the host ID, port, and optionally SSL security credentials
- Reference the **.broker** file on many commands with **–n** option instead of specifying an integration node name

Example:

IBNODE.broker

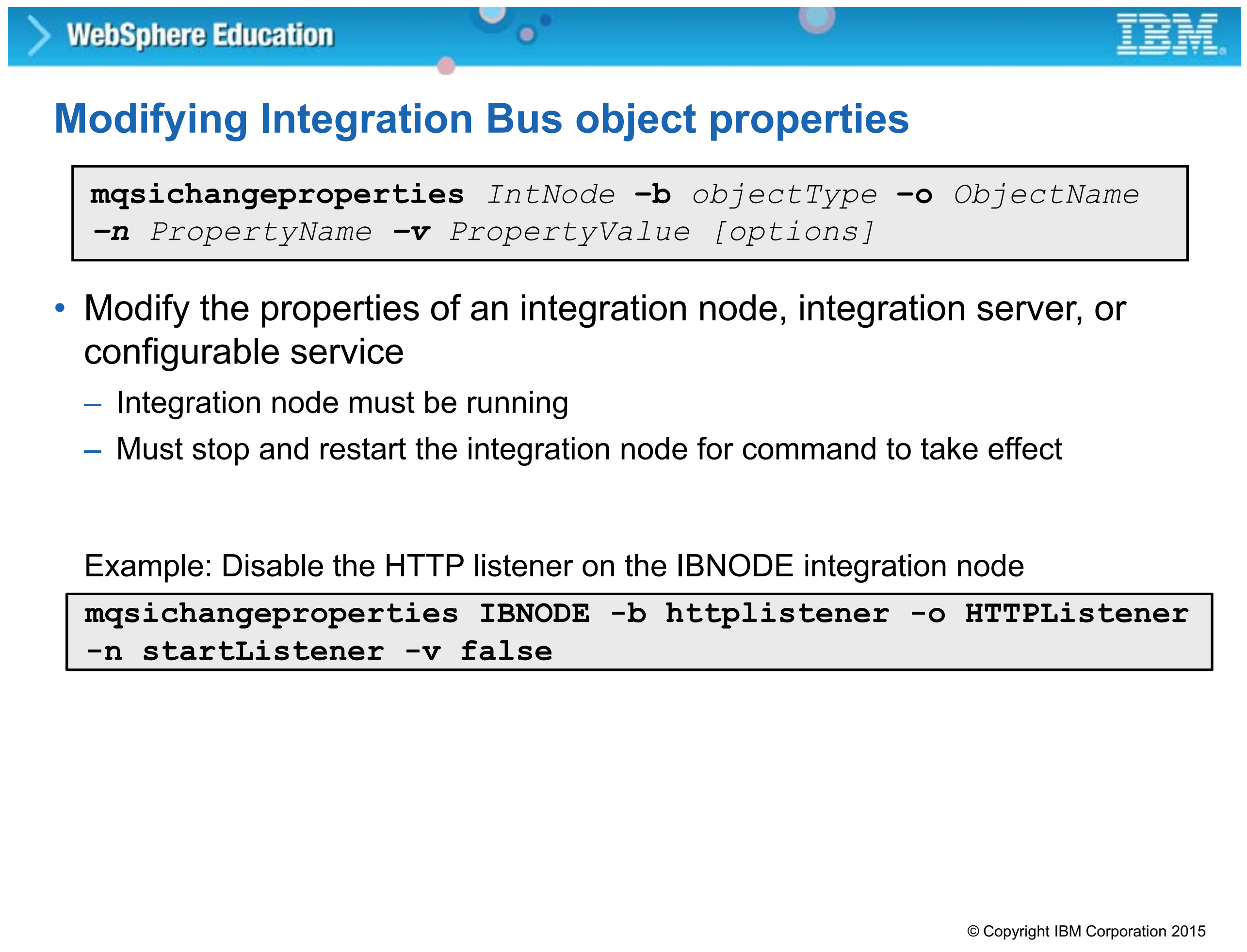
```
<?xml version="1.0" encoding="utf-8"?>
<IntegrationNodeConnectionParameters Version="10.0.0" host="localhost"
listenerPort="4414" integrationNodeName="IBNODE" userName="user"
useSsl="true" sslTrustStorePassword="password"
sslTrustStorePath="c:\keystore\my_keystore.jks"
xmlns="http://www.ibm.com/xmlns/prod/websphere/iib/8/
IntegrationNodeConnectionParameters" />
```

```
mqsicreateexecutiongroup –n IBNODE.broker –e IS1
```

Figure 5-7. Connecting to an integration node by using a .broker file WM6461.0

Notes:

You can use a `.broker` file to define connection details for a specific integration node on most commands by using the `-n` parameter.



WebSphere Education

Modifying Integration Bus object properties

```
mqsichangeproperties IntNode -b objectType -o ObjectName
-n PropertyName -v PropertyValue [options]
```

- Modify the properties of an integration node, integration server, or configurable service
  - Integration node must be running
  - Must stop and restart the integration node for command to take effect

Example: Disable the HTTP listener on the IBNODE integration node

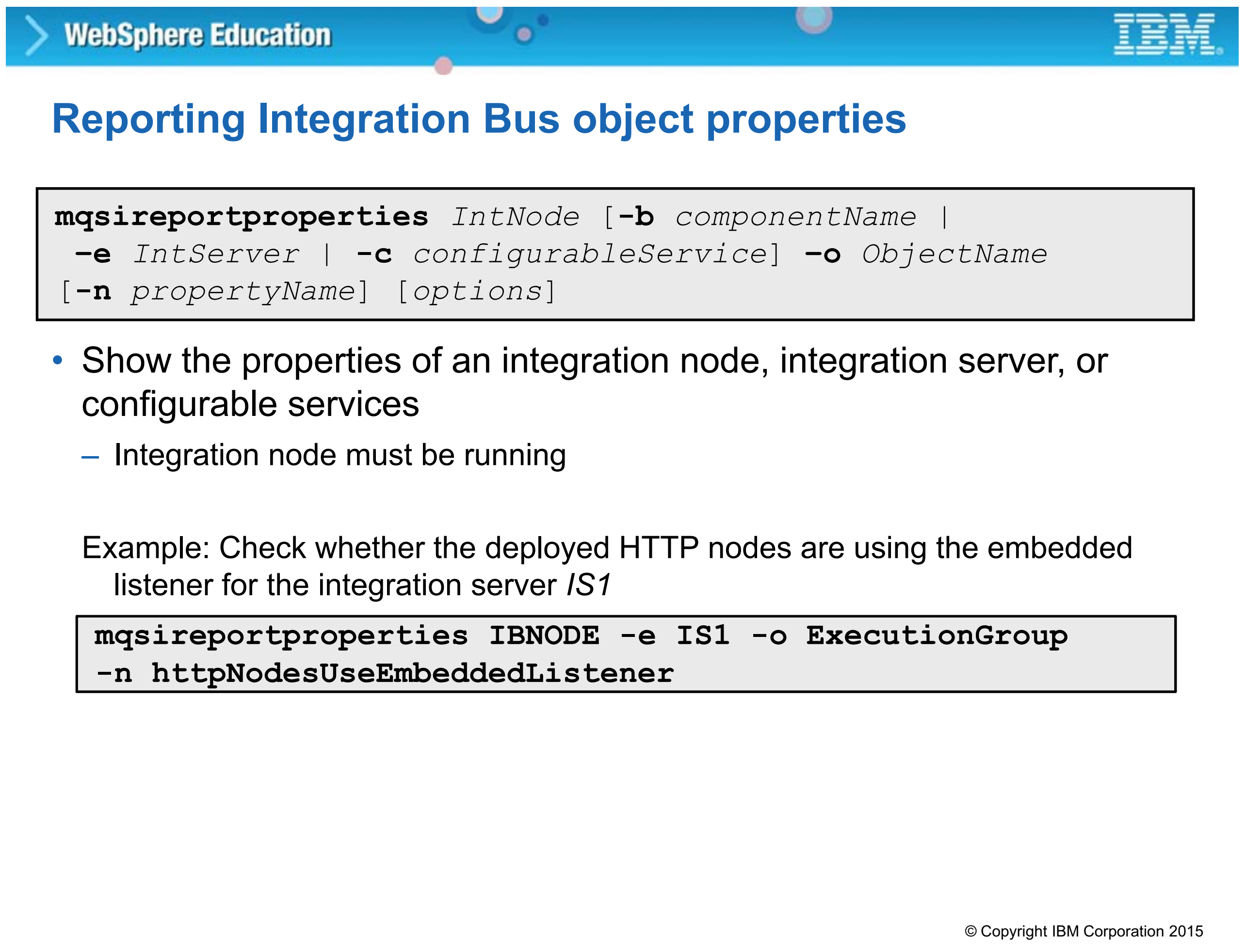
```
mqsichangeproperties IBNODE -b httplistener -o HTTPListener
-n startListener -v false
```

Figure 5-8. Modifying Integration Bus object properties WM6461.0

Notes:

You can use the `mqsichangeproperties` command to modify integration node properties and configurable properties of integration node resources.

Configurable services are typically runtime properties that are related to external services on which the integration node relies.



WebSphere Education

Reporting Integration Bus object properties

```
mqsireportproperties IntNode [-b componentName |
 -e IntServer | -c configurableService] -o ObjectName
[-n propertyName] [options]
```

- Show the properties of an integration node, integration server, or configurable services
  - Integration node must be running

Example: Check whether the deployed HTTP nodes are using the embedded listener for the integration server *IS1*

```
mqsireportproperties IBNODE -e IS1 -o ExecutionGroup
-n httpNodesUseEmbeddedListener
```

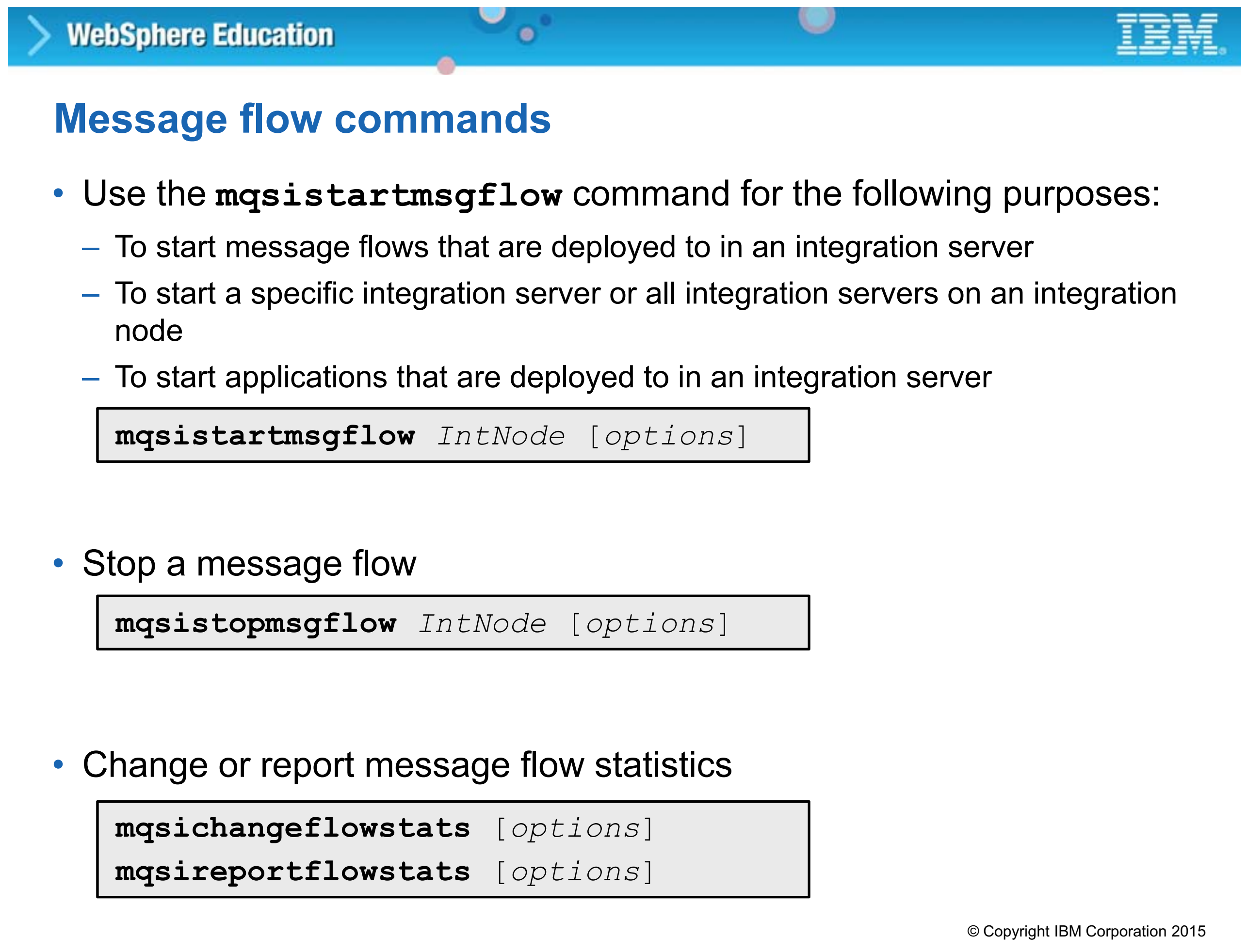
Figure 5-9. Reporting Integration Bus object properties WM6461.0

Notes:

The `mqsireportproperties` command displays integration node properties and configurable properties of integration node resources. The figure summarizes the command syntax.

> `-b` identifies the component. Valid values are `httplistener`, `securitycache`, and `servicefederation`.
>
> `-c` is the type of the configurable service. Specify a value of `AllTypes` to report on all configurable service types. For a list of supplied configurable services, and their properties and values, see “Configurable services properties” in the product documentation.
>
> `-e` identifies the integration server for which a report is required.
>
> `-o` identifies the object whose properties you want to read.

For a complete description of the command syntax, see the IBM Integration Bus product documentation.



WebSphere Education IBM

Message flow commands

- Use the **`mqsistartmsgflow`** command for the following purposes:
  - To start message flows that are deployed to in an integration server
  - To start a specific integration server or all integration servers on an integration node
  - To start applications that are deployed to in an integration server

```
mqsistartmsgflow IntNode [options]
```

- Stop a message flow

```
mqsistopmsgflow IntNode [options]
```

- Change or report message flow statistics

```
mqsichangeflowstats [options]
mqsireportflowstats [options]
```

© Copyright IBM Corporation 2015

Figure 5-10. Message flow commands WM6461.0

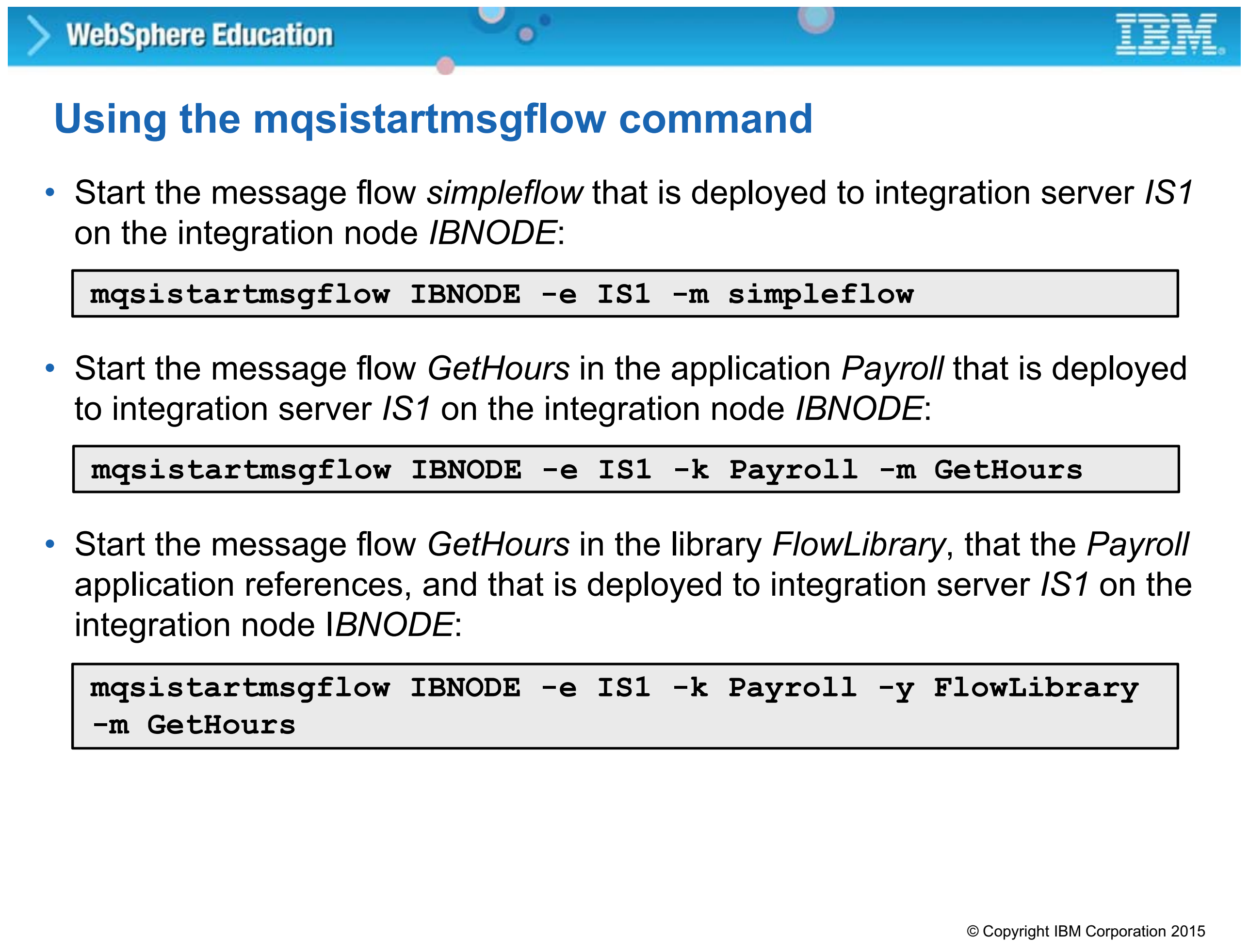
Notes:

After a message flow is deployed, it can be manually started with the `mqsistartmsgflow` command and stopped with the `mqsistopmsgflow` command.

The commands for message flows have many options for controlling how the message flows are started and stopped. For example, you can control whether message flows are started in a single integration server by using the `-e` option. You can use the `-m` option to specify message flow that is running on the integration server (which means that the `-e` option must also be present).

You can use the `mqsichangeflowstats` and `mqsireportflowstats` commands for controlling message flow statistics. These commands are described in more detail later in this course.

For a complete description of all commands and command syntax, see the product documentation.



Using the mqsistartmsgflow command

- Start the message flow *simpleflow* that is deployed to integration server *IS1* on the integration node *IBNODE*:

```
mqsistartmsgflow IBNODE -e IS1 -m simpleflow
```

- Start the message flow *GetHours* in the application *Payroll* that is deployed to integration server *IS1* on the integration node *IBNODE*:

```
mqsistartmsgflow IBNODE -e IS1 -k Payroll -m GetHours
```

- Start the message flow *GetHours* in the library *FlowLibrary*, that the *Payroll* application references, and that is deployed to integration server *IS1* on the integration node I*BNODE*:

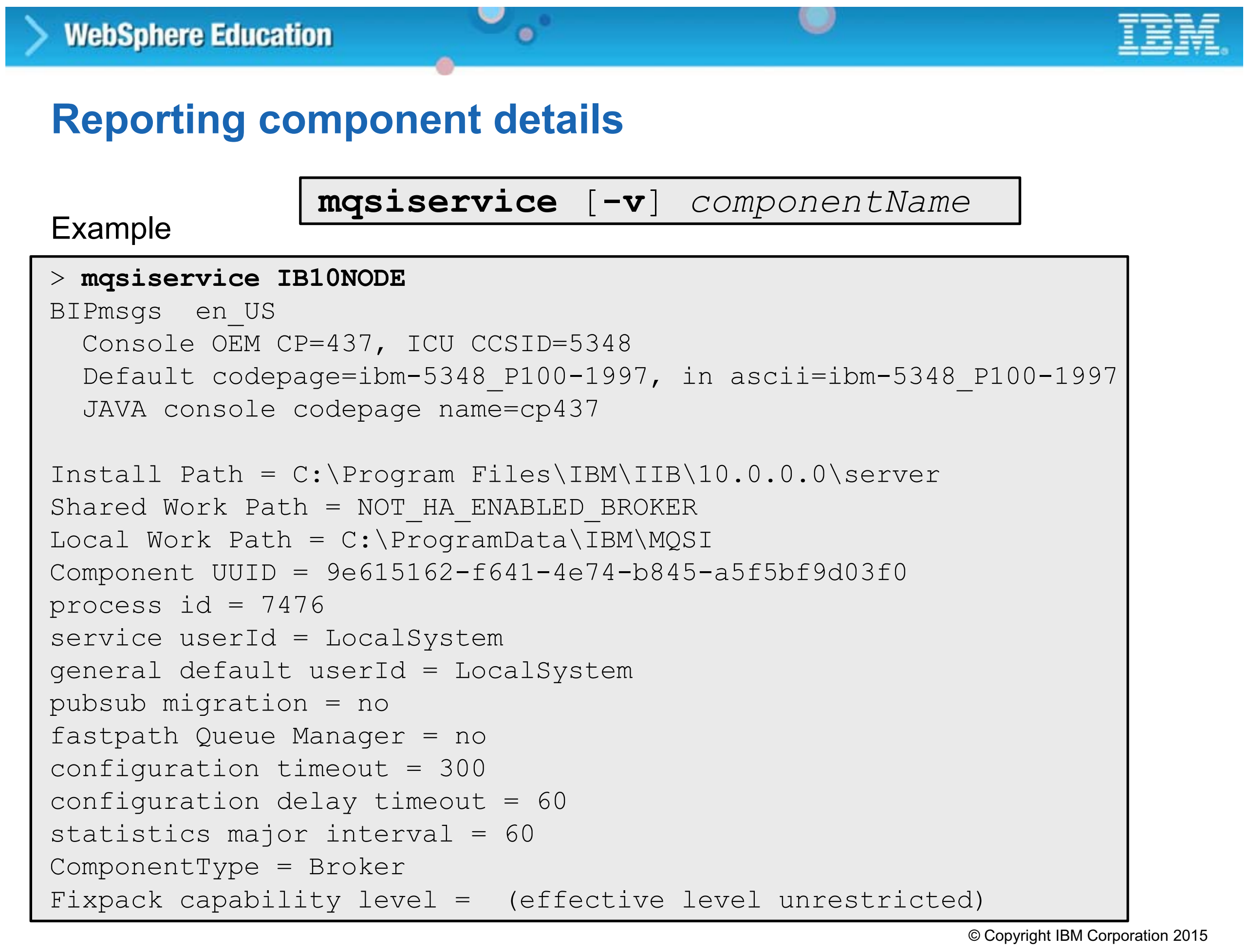
```
mqsistartmsgflow IBNODE -e IS1 -k Payroll -y FlowLibrary
-m GetHours
```

Figure 5-11. Using the mqsistartmsgflow command WM6461.0

Notes:

This figure provides some examples of the `mqsistartmsgflow` command.

Before you use this command, ensure that already deployed the message flows, if specified, to the integration node in a BAR file. You can start message flows only if the integration server to which the message flow is deployed is running.



WebSphere Education

Reporting component details

```
mqsiservice [-v] componentName
```

Example

```
> mqsiservice IB10NODE
BIPmsgs  en_US
  Console OEM CP=437, ICU CCSID=5348
  Default codepage=ibm-5348_P100-1997, in ascii=ibm-5348_P100-1997
  JAVA console codepage name=cp437

Install Path = C:\Program Files\IBM\IIB\10.0.0.0\server
Shared Work Path = NOT_HA_ENABLED_BROKER
Local Work Path = C:\ProgramData\IBM\MQSI
Component UUID = 9e615162-f641-4e74-b845-a5f5bf9d03f0
process id = 7476
service userId = LocalSystem
general default userId = LocalSystem
pubsub migration = no
fastpath Queue Manager = no
configuration timeout = 300
configuration delay timeout = 60
statistics major interval = 60
ComponentType = Broker
Fixpack capability level =  (effective level unrestricted)
```

Figure 5-12. Reporting component details WM6461.0

Notes:

Use the mqsiservice command to report on the configuration of one or more integration nodes. The mqsiservice command syntax is:

```
mqsiservice [-v] [componentName [-r label=value]] [-m messageNumber
[-c messageCatalog]] [-t]
```

The command includes the following options:

-v is the product version information.

componentName is the name of the component to query for its registry data.

-r label=value sets the registry key 'label' to 'value'. Use with caution.

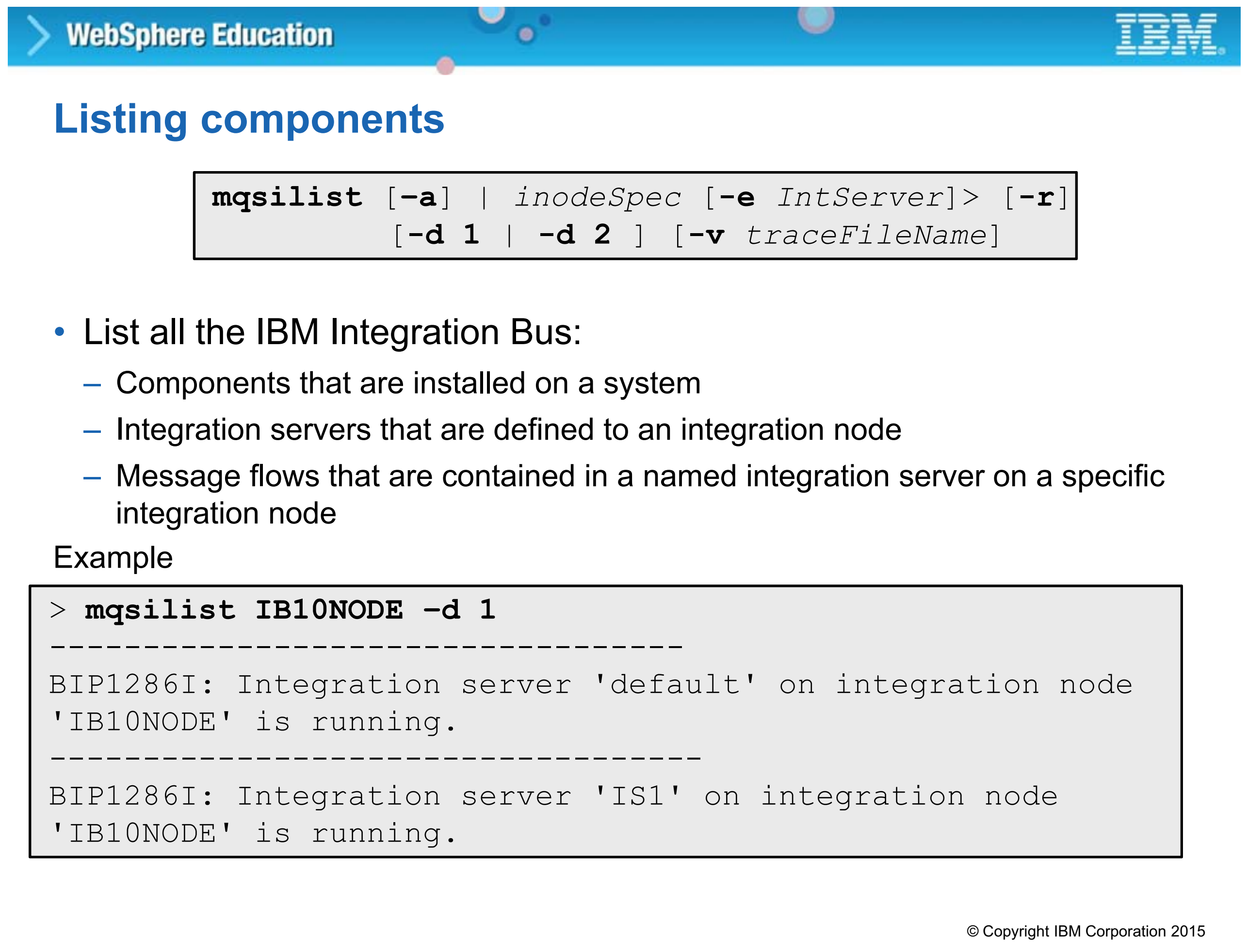
-r label= sets the registry key 'label' to null. Use with caution.

-r label deletes the registry key 'label'. Use with caution.

-m messageNumber is a message number to output.

-c messageCatalog is the name of the message catalog to use.

-t outputs information about current time and time zone.



WebSphere Education

Listing components

```
mqsilist [-a] | inodeSpec [-e IntServer]> [-r]
          [-d 1 | -d 2 ] [-v traceFileName]
```

- List all the IBM Integration Bus:
  - Components that are installed on a system
  - Integration servers that are defined to an integration node
  - Message flows that are contained in a named integration server on a specific integration node

Example

```
> mqsilist IB10NODE -d 1
----------------------------------
BIP1286I: Integration server 'default' on integration node
'IB10NODE' is running.
-----------------------------------
BIP1286I: Integration server 'IS1' on integration node
'IB10NODE' is running.
```

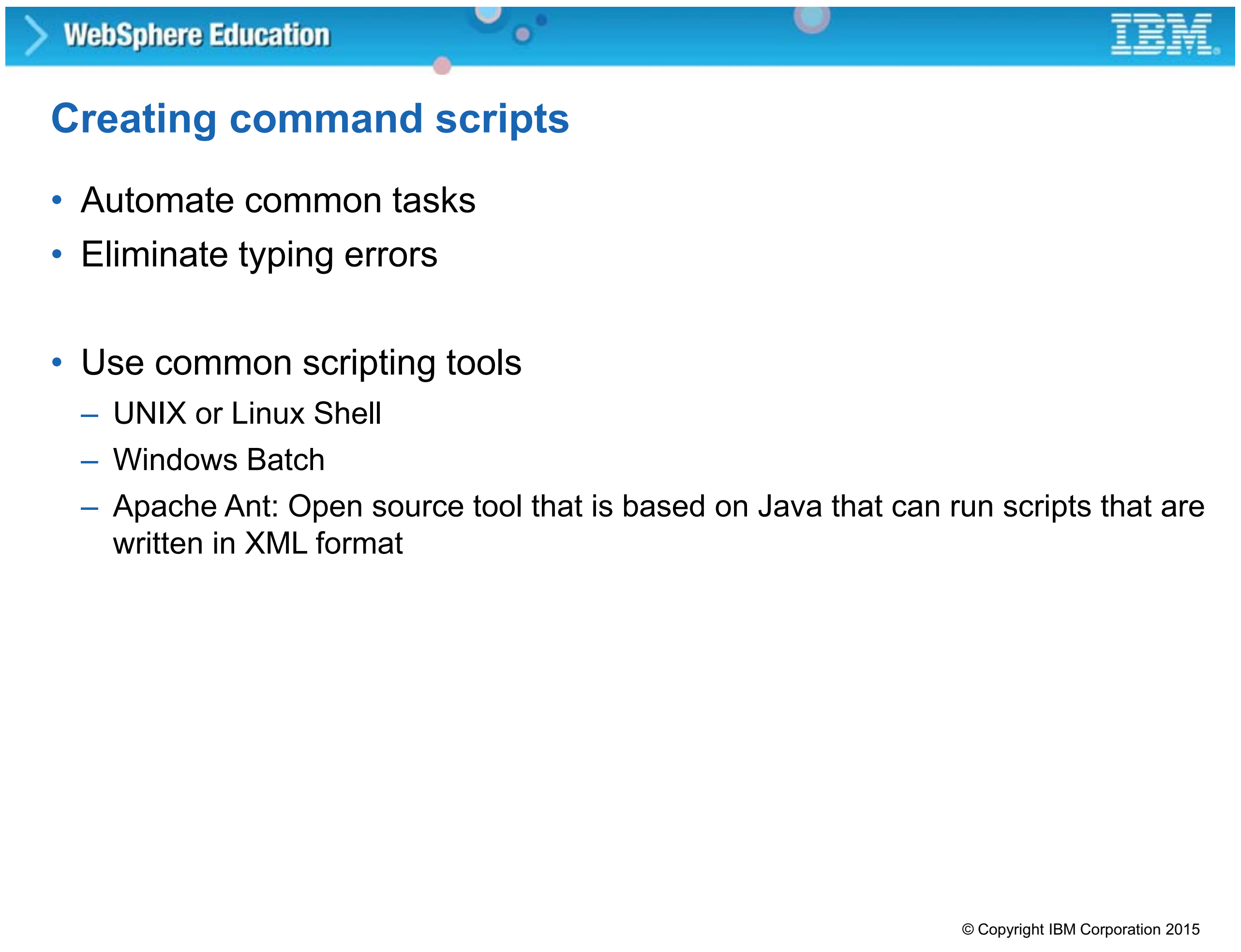
© Copyright IBM Corporation 2015

Figure 5-13. Listing components WM6461.0

Notes:

Use the `mqsilist` command to list the IBM Integration Bus components that are installed on a computer.

Use the `mqsilist` command with no options to list all of the components that are installed on the system.



WebSphere Education

Creating command scripts

- Automate common tasks
- Eliminate typing errors

- Use common scripting tools
  - UNIX or Linux Shell
  - Windows Batch
  - Apache Ant: Open source tool that is based on Java that can run scripts that are written in XML format

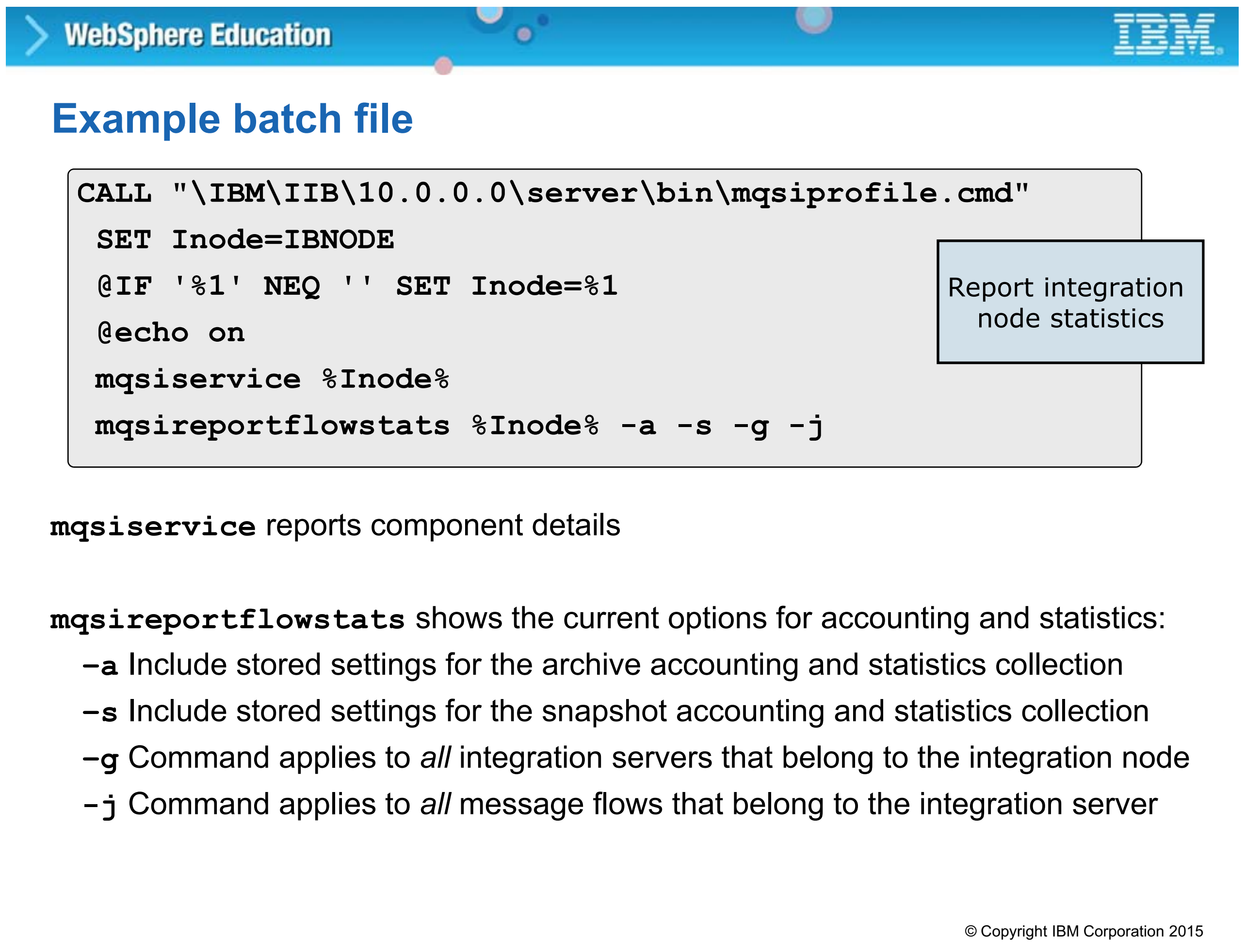
© Copyright IBM Corporation 2015

Figure 5-14. Creating command scripts WM6461.0

Notes:

IBM Integration Bus commands can be included in batch files and command scripts to automate common tasks such as modifying configurable properties in a BAR file.

Any scripting tool can be used, but the user who runs the script must have the correct permissions. One of the newer scripting tools is Apache Ant, which is an open source build tool that is based on Java.



WebSphere Education

Example batch file

```
CALL "\IBM\IIB\10.0.0.0\server\bin\mqsiprofile.cmd"
 SET Inode=IBNODE
 @IF '%1' NEQ '' SET Inode=%1
 @echo on
 mqsiservice %Inode%
 mqsireportflowstats %Inode% -a -s -g -j
```

Report integration node statistics

`mqsiservice` reports component details

`mqsireportflowstats` shows the current options for accounting and statistics:

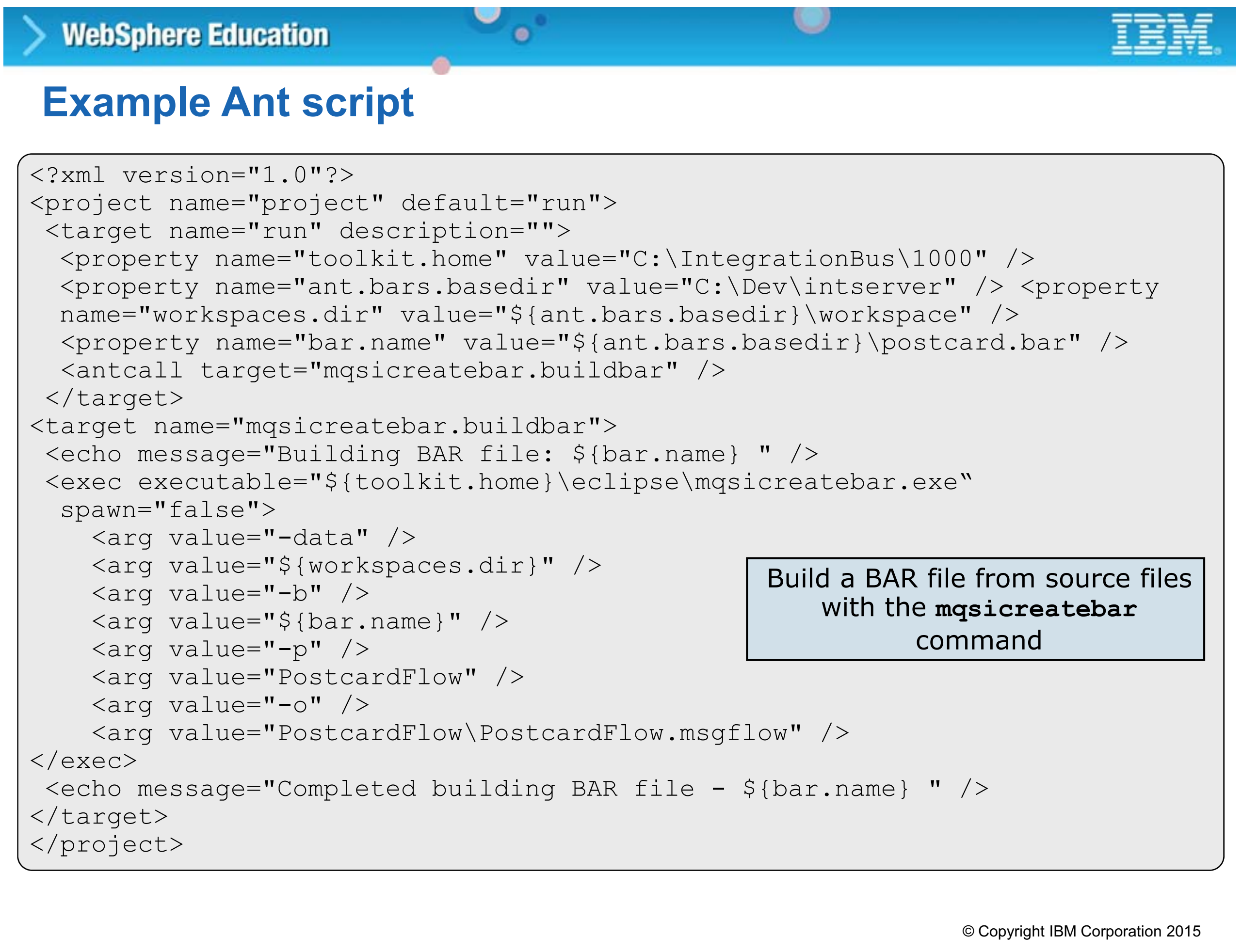
- **-a** Include stored settings for the archive accounting and statistics collection
- **-s** Include stored settings for the snapshot accounting and statistics collection
- **-g** Command applies to *all* integration servers that belong to the integration node
- **-j** Command applies to *all* message flows that belong to the integration server

Figure 5-15. Example batch file WM6461.0

Notes:

The example batch file in the figure reports integration node statistics for all integration servers and all message flows.

The command has an optional argument for specifying the integration node. If an integration node is not specified, the command is run against the integration node that is named `IBNODE`.



WebSphere Education

Example Ant script

```
<?xml version="1.0"?>
<project name="project" default="run">
 <target name="run" description="">
  <property name="toolkit.home" value="C:\IntegrationBus\1000" />
  <property name="ant.bars.basedir" value="C:\Dev\intserver" /> <property
  name="workspaces.dir" value="${ant.bars.basedir}\workspace" />
  <property name="bar.name" value="${ant.bars.basedir}\postcard.bar" />
  <antcall target="mqsicreatebar.buildbar" />
 </target>
<target name="mqsicreatebar.buildbar">
 <echo message="Building BAR file: ${bar.name} " />
 <exec executable="${toolkit.home}\eclipse\mqsicreatebar.exe"
  spawn="false">
    <arg value="-data" />
    <arg value="${workspaces.dir}" />
    <arg value="-b" />
    <arg value="${bar.name}" />
    <arg value="-p" />
    <arg value="PostcardFlow" />
    <arg value="-o" />
    <arg value="PostcardFlow\PostcardFlow.msgflow" />
</exec>
 <echo message="Completed building BAR file - ${bar.name} " />
</target>
</project>
```

Build a BAR file from source files with the **`mqsicreatebar`** command

© Copyright IBM Corporation 2015

Figure 5-16. Example Ant script WM6461.0

Notes:

The figure shows an example of an Ant script that builds a BAR file from source files that were written with the `mqsicreatebar` command. Ant is like a Java version of “make”. Ant scripts are written in XML, and as with “make”, Ant targets can depend on other targets.

The script builds and runs the `mqsicreatebar` command with the following options:

`-data` identifies the workspace directory that contains the message flow application projects.

`-b` identifies the BAR file name.

`-p` specifies the projects to include in the BAR file.

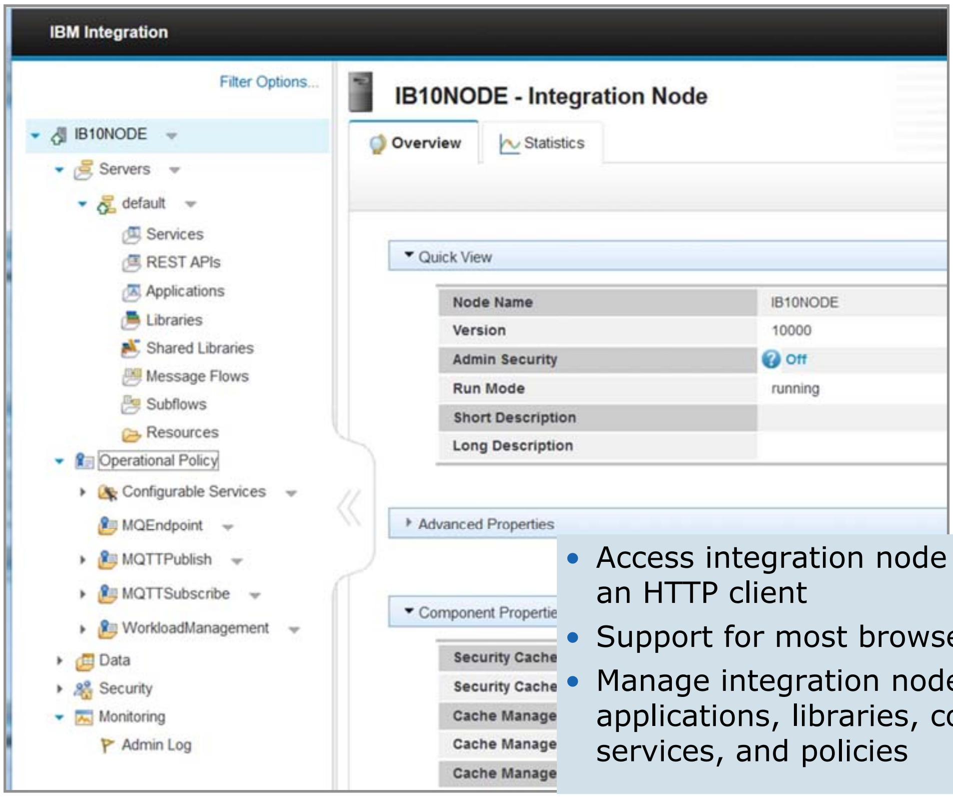
`-o` identifies the workspace relative path (including the project) of a message flow (`.msgflow`) file to add to the BAR file.

For more information, see the IBM developerWorks topic “IBM Integration Bus deployment scripting by using Ant”.

5.2. IBM Integration web user interface

WebSphere Education

IBM Integration web user interface



Node Name	IB10NODE
Version	10000
Admin Security	Off
Run Mode	running
Short Description	
Long Description	

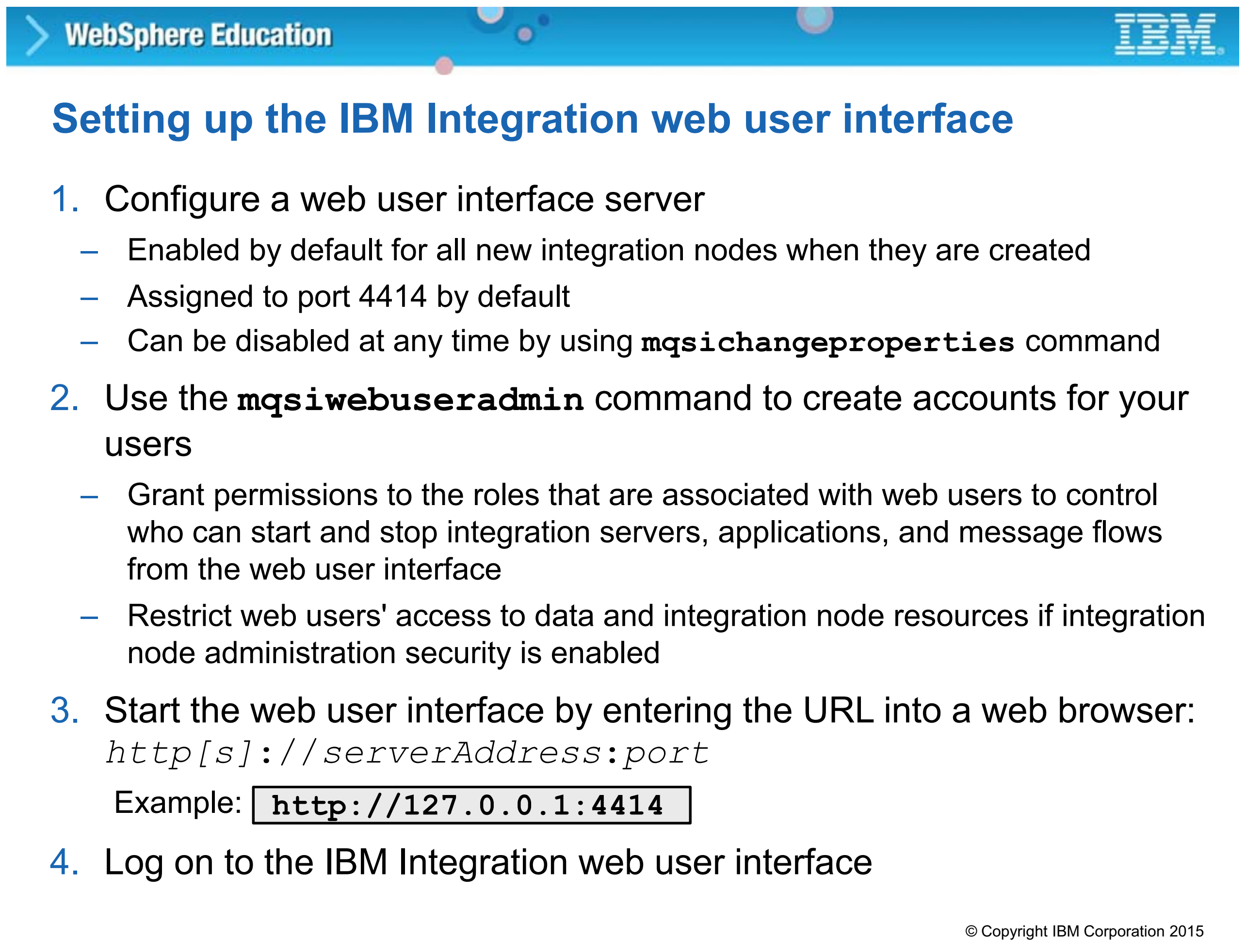
- Access integration node resources through an HTTP client
- Support for most browsers
- Manage integration nodes, message flow applications, libraries, configurable services, and policies

Figure 5-17. IBM Integration web user interface WM6461.0

Notes:

The IBM Integration Bus web user interface enables web users to access integration node resources through an HTTP client. It provides Integration Bus administrators with an alternative to using commands for administering integration node resources.

Support is provided for most major browsers. For more information about supported browsers, see the product documentation.



WebSphere Education IBM

Setting up the IBM Integration web user interface

1. Configure a web user interface server
   - Enabled by default for all new integration nodes when they are created
   - Assigned to port 4414 by default
   - Can be disabled at any time by using **`mqsichangeproperties`** command
2. Use the **`mqsiwebuseradmin`** command to create accounts for your users
   - Grant permissions to the roles that are associated with web users to control who can start and stop integration servers, applications, and message flows from the web user interface
   - Restrict web users' access to data and integration node resources if integration node administration security is enabled
3. Start the web user interface by entering the URL into a web browser: *`http[s]://serverAddress:port`*

   Example: **`http://127.0.0.1:4414`**
4. Log on to the IBM Integration web user interface

© Copyright IBM Corporation 2015

Figure 5-18. Setting up the IBM Integration web user interface WM6461.0

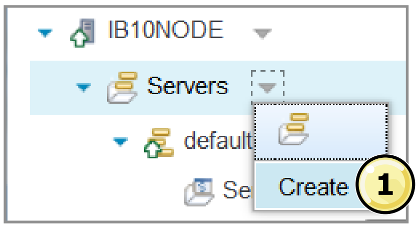
Notes:

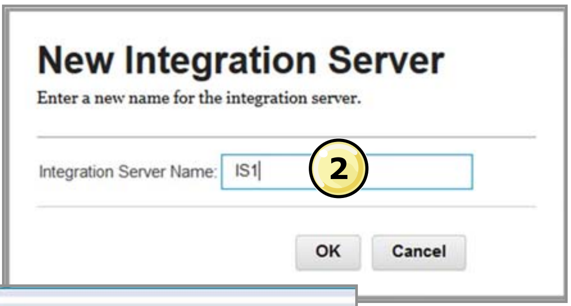
The web user interface is enabled by default for all new integration nodes when they are created. You can disable and enable the web user interface at any time.

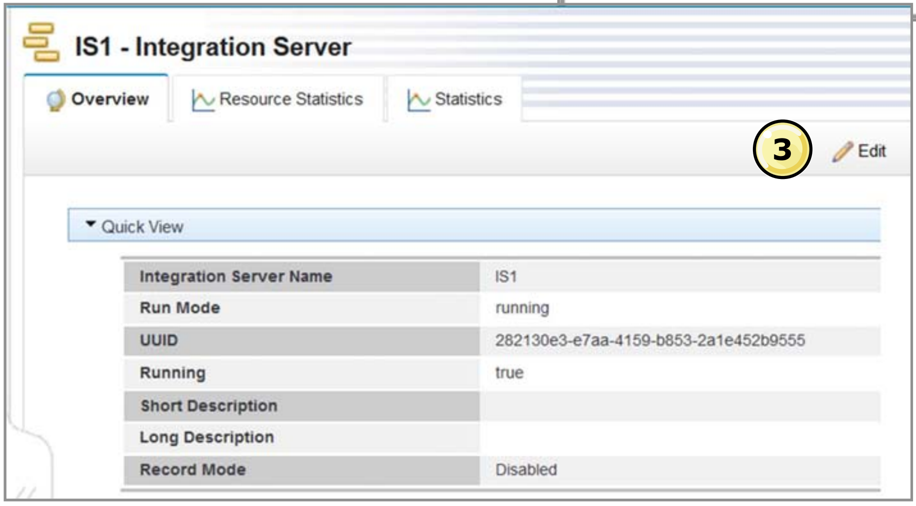
Complete the steps in the figure to configure the web user interface, create the required web user accounts, and log on to the web user interface.

Optionally, you can also customize the web user interface according to individual preferences.

Creating integration servers in the IBM Integration web user interface







Integration Server Name	IS1
Run Mode	running
UUID	282130e3-e7aa-4159-b853-2a1e452b9555
Running	true
Short Description	
Long Description	
Record Mode	Disabled

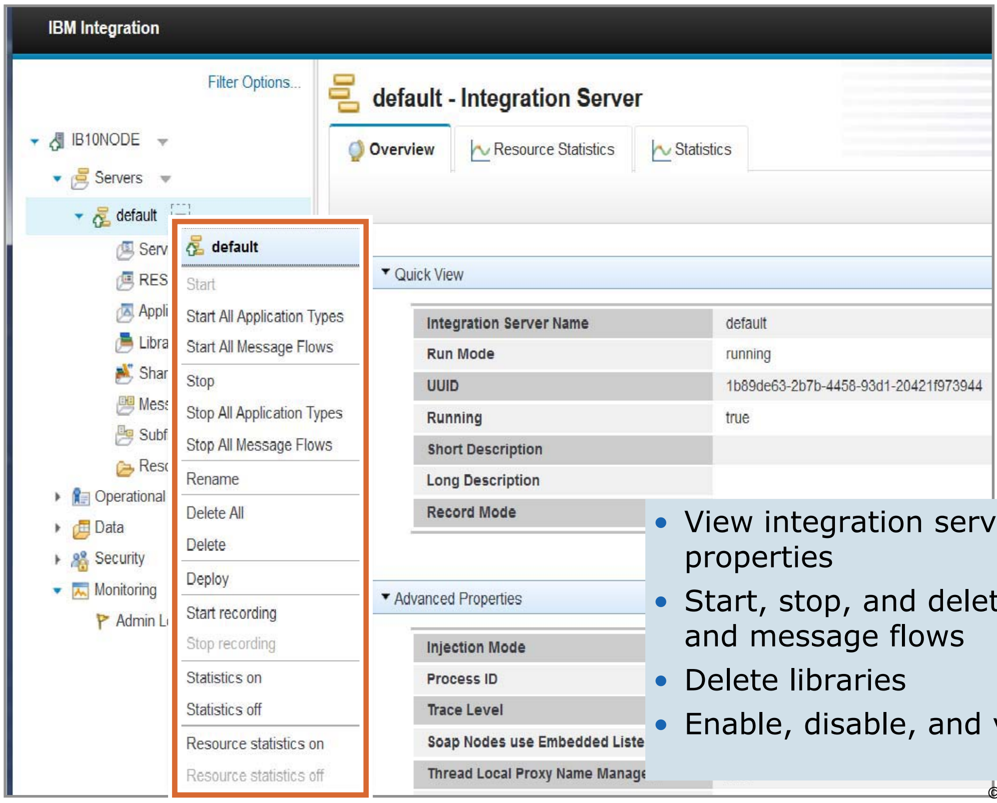
1. Click **Servers** arrow and then click **Create**.
2. Type integration server name and then click **OK**.
3. (Optional) Click **Edit** to modify integration server properties.

Figure 5-19. Creating integration servers in the IBM Integration web user interface WM6461.0

Notes:

This figure shows the steps for creating an integration server in the IBM Integration web user interface.

Managing integration servers in the IBM Integration web user interface



Integration Server Name	default
Run Mode	running
UUID	1b89de63-2b7b-4458-93d1-20421f973944
Running	true
Short Description	
Long Description	
Record Mode	

Injection Mode
Process ID
Trace Level
Soap Nodes use Embedded Liste
Thread Local Proxy Name Manage

- View integration servers and server properties
- Start, stop, and delete applications and message flows
- Delete libraries
- Enable, disable, and view statistics

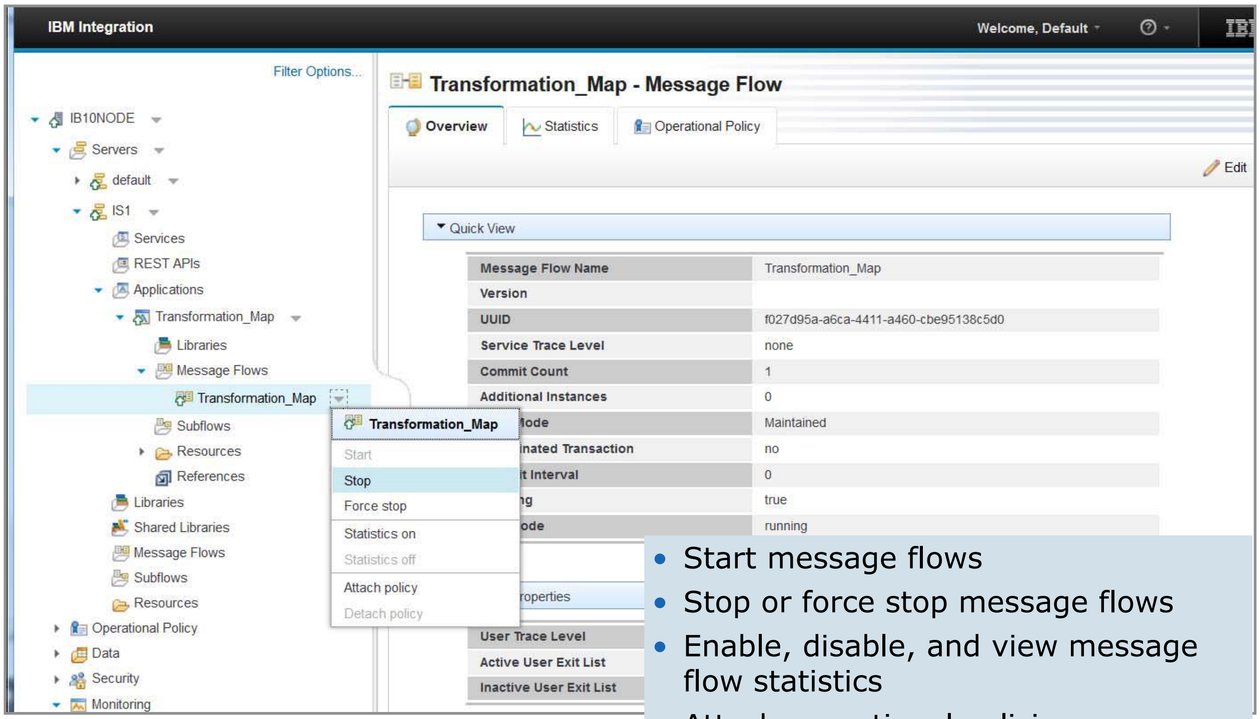
Figure 5-20. Managing integration servers in the IBM Integration web user interface WM6461.0

Notes:

You control integration servers and view integration server properties in the IBM Integration web user interface. You can also control message flow applications, libraries, and statistics.

WebSphere Education

Managing message flows in the IBM Integration web user interface



Message Flow Name	Transformation_Map
Version	
UUID	f027d95a-a6ca-4411-a460-cbe95138c5d0
Service Trace Level	none
Commit Count	1
Additional Instances	0
...lode	Maintained
...inated Transaction	no
...t Interval	0
...ng	true
...ode	running
User Trace Level	
Active User Exit List	
Inactive User Exit List	

- Start message flows
- Stop or force stop message flows
- Enable, disable, and view message flow statistics
- Attach operational policies

Figure 5-21. Managing message flows in the IBM Integration web user interface WM6461.0

Notes:

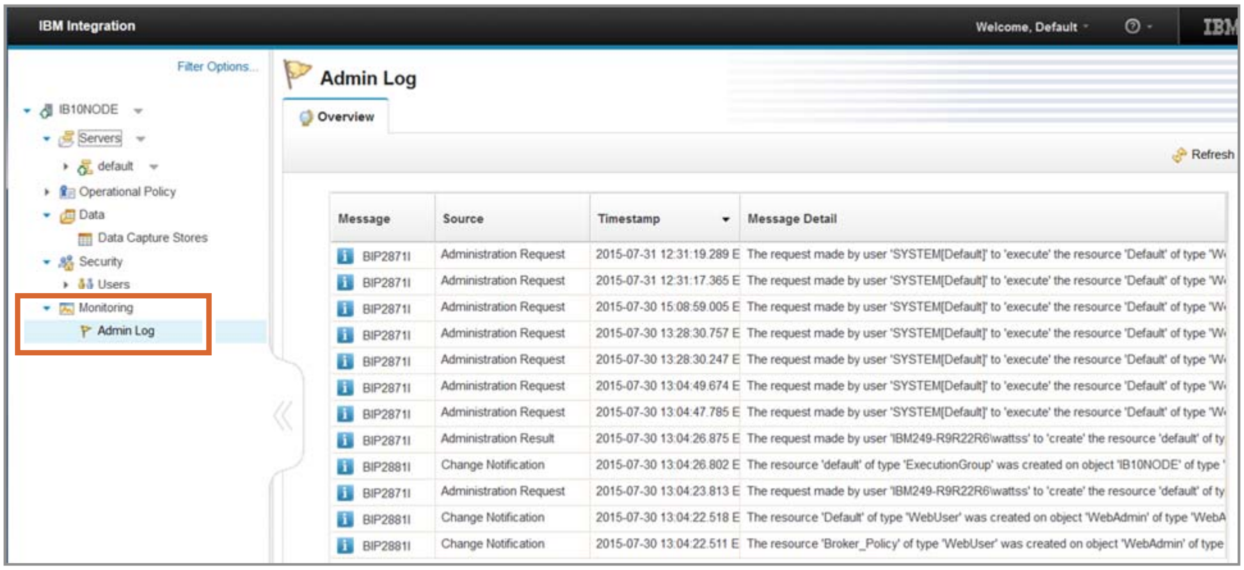
You can use the web user interface to start and stop message flow applications.

You can also use the web user interface to work with statistics and accounting data for your message flows. You can start and stop the collection of snapshot statistics and accounting data, and then display the data in a format that helps you to analyze and tune the performance of your message flows and applications.

You can also use the web user interface to create workload management policies and attach them to message flows.

WebSphere Education

IBM Integration web user interface Administration Log



Admin Log

Message	Source	Timestamp	Message Detail
BIP2871I	Administration Request	2015-07-31 12:31:19.289 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-31 12:31:17.365 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-30 15:08:59.005 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-30 13:28:30.757 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-30 13:28:30.247 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-30 13:04:49.674 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Request	2015-07-30 13:04:47.785 E	The request made by user 'SYSTEM[Default]' to 'execute' the resource 'Default' of type 'W
BIP2871I	Administration Result	2015-07-30 13:04:26.875 E	The request made by user 'IBM249-R9R22R6\wattss' to 'create' the resource 'default' of ty
BIP2881I	Change Notification	2015-07-30 13:04:26.802 E	The resource 'default' of type 'ExecutionGroup' was created on object 'IB10NODE' of type '
BIP2871I	Administration Request	2015-07-30 13:04:23.813 E	The request made by user 'IBM249-R9R22R6\wattss' to 'create' the resource 'default' of ty
BIP2881I	Change Notification	2015-07-30 13:04:22.518 E	The resource 'Default' of type 'WebUser' was created on object 'WebAdmin' of type 'WebA
BIP2881I	Change Notification	2015-07-30 13:04:22.511 E	The resource 'Broker_Policy' of type 'WebUser' was created on object 'WebAdmin' of type

- Managed by the integration node
- Displays messages about events that occur within the integration node such as results of a deployment or a change to integration node properties
- Messages can be information, errors, or warnings

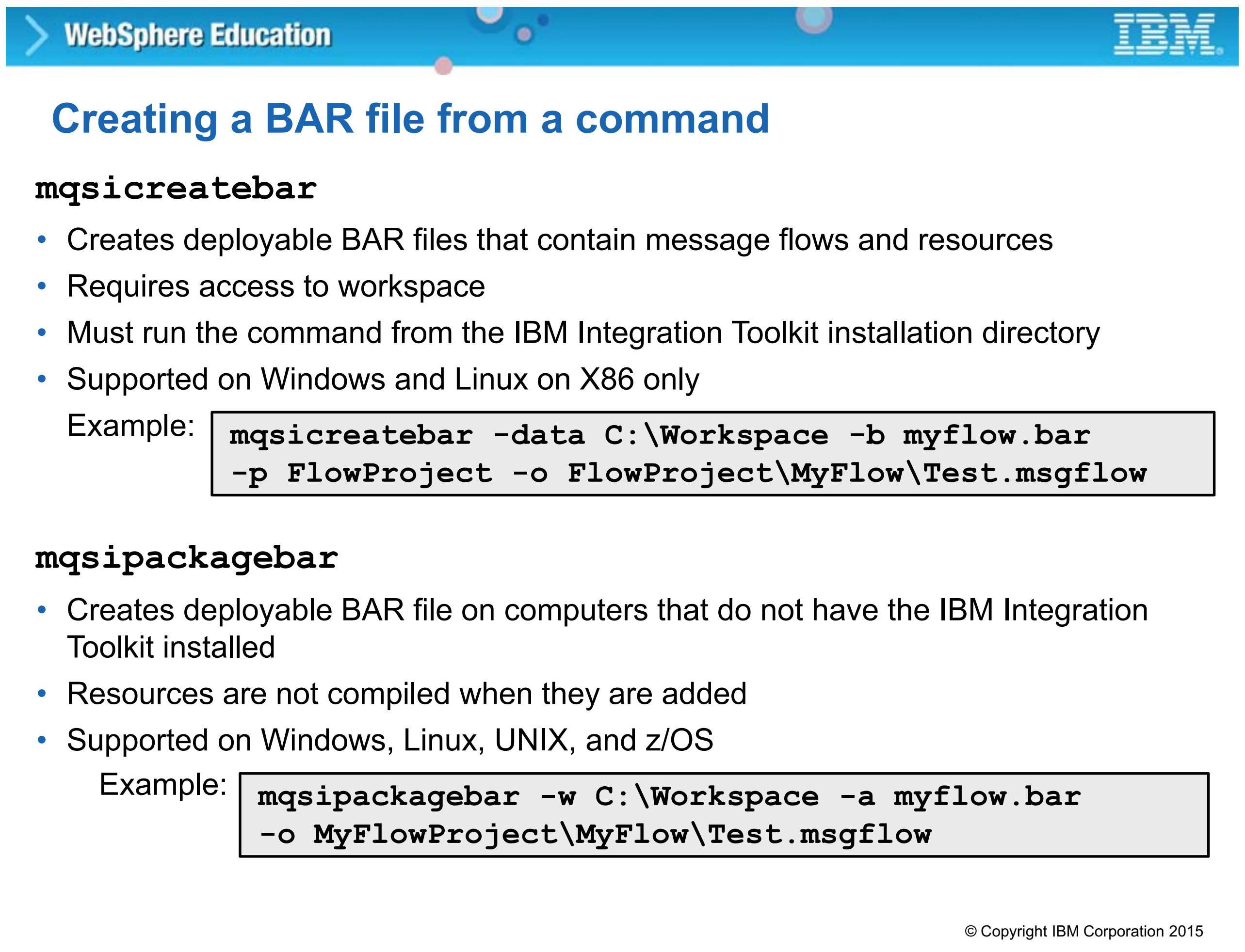
Figure 5-22. IBM Integration web user interface Administration Log WM6461.0

Notes:

The IBM Integration web user interface Administration (Admin) Log view shows messages about events that occur within the integration node.

5.3. Deploying message flow applications

This topic describes how to use IBM Integration Bus commands and the web user interface to deploy message flow applications.



WebSphere Education

Creating a BAR file from a command

mqsicreatebar

- Creates deployable BAR files that contain message flows and resources
- Requires access to workspace
- Must run the command from the IBM Integration Toolkit installation directory
- Supported on Windows and Linux on X86 only

Example:

```
mqsicreatebar -data C:\Workspace -b myflow.bar
-p FlowProject -o FlowProject\MyFlow\Test.msgflow
```

mqsipackagebar

- Creates deployable BAR file on computers that do not have the IBM Integration Toolkit installed
- Resources are not compiled when they are added
- Supported on Windows, Linux, UNIX, and z/OS

Example:

```
mqsipackagebar -w C:\Workspace -a myflow.bar
-o MyFlowProject\MyFlow\Test.msgflow
```

Figure 5-23. Creating a BAR file from a command WM6461.0

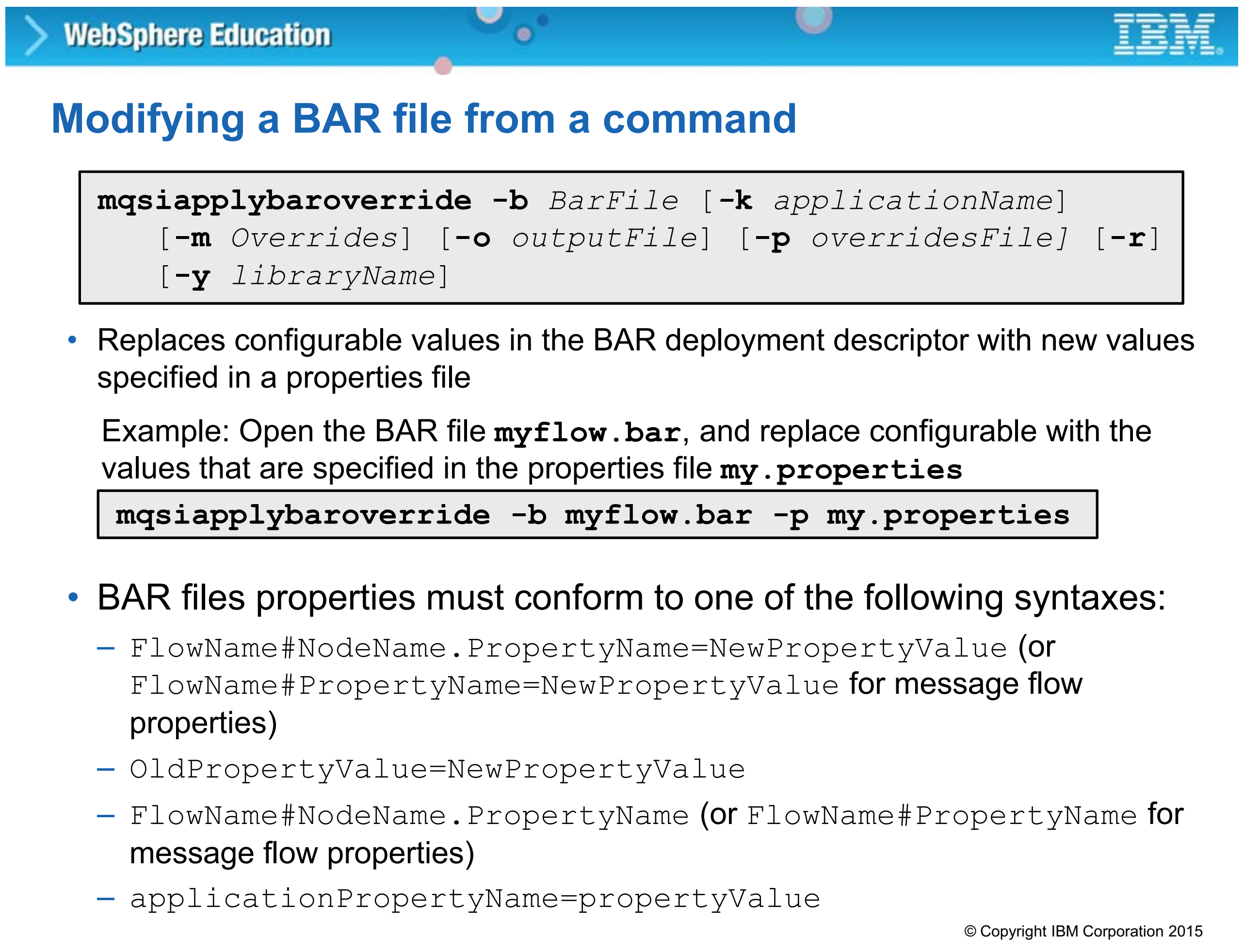
Notes:

The `mqsicreatebar` command creates deployable BAR files that contain compiled message flows and models. The `mqsicreatebar` command compiles the message flows and requires IBM Integration Toolkit.

As an option, you can use the `mqsipackagebar` command to create deployable BAR files. You can use this command to create BAR files on computers that do not have the IBM Integration Toolkit installed. Resources that are added to a BAR file by using the `mqsipackagebar` command are not compiled when they are added. A BAR file that is created by using this command, must contain the deployable resources.

If you use a repository to store your message flows and dictionaries, you can write scripts that use the `mqsipackagebar` or the `mqsicreatebar` commands with the repository command tools to deploy the message flow applications.

The fully qualified path is required when specifying the BAR file name in the commands.



WebSphere Education

Modifying a BAR file from a command

```
mqsiapplybaroverride -b BarFile [-k applicationName]
   [-m Overrides] [-o outputFile] [-p overridesFile] [-r]
   [-y libraryName]
```

- Replaces configurable values in the BAR deployment descriptor with new values specified in a properties file

  Example: Open the BAR file **`myflow.bar`**, and replace configurable with the values that are specified in the properties file **`my.properties`**

  ```
  mqsiapplybaroverride -b myflow.bar -p my.properties
  ```

- BAR files properties must conform to one of the following syntaxes:
  - `FlowName#NodeName.PropertyName=NewPropertyValue` (or `FlowName#PropertyName=NewPropertyValue` for message flow properties)
  - `OldPropertyValue=NewPropertyValue`
  - `FlowName#NodeName.PropertyName` (or `FlowName#PropertyName` for message flow properties)
  - `applicationPropertyName=propertyValue`

© Copyright IBM Corporation 2015

Figure 5-24. Modifying a BAR file from a command WM6461.0

Notes:

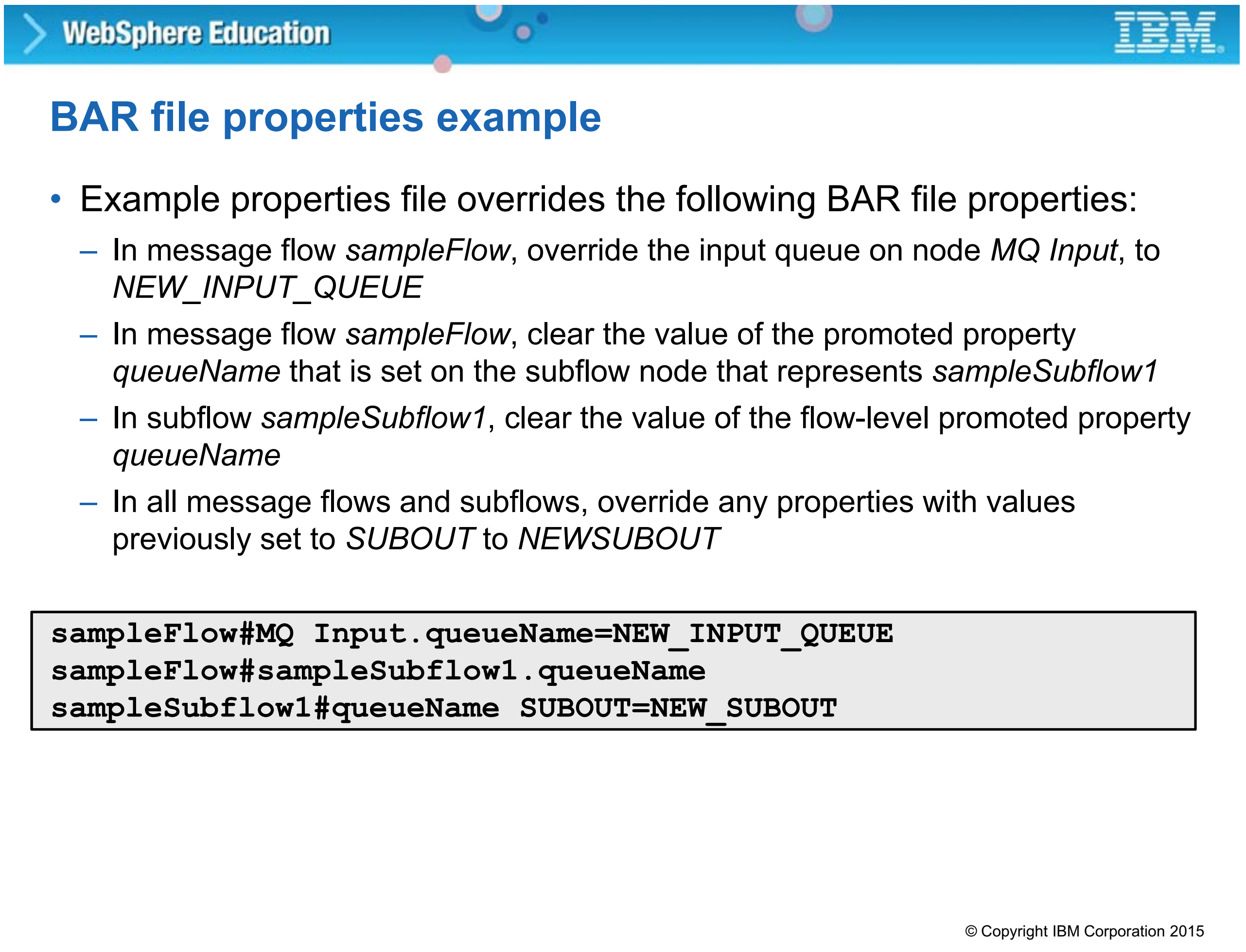
You can use the `mqsiapplybaroverride` command to modify a BAR file.

You can also use the `mqsiapplybaroverride` command to replace configurable values in the BAR deployment descriptor, `META-INF/broker.xml` with new values that you specify in a properties file. Configurable properties allow an administrator to update target-dependent properties, such as queue names, queue manager names, and database connections.

By changing configurable properties, you can customize a BAR file for a new environment, for example a test system, without editing and rebuilding the message flows, message maps, or ESQL transformation program.

Follow these steps to edit properties with the `mqsiapplybaroverride` command:

1. Open an Integration Bus command window that is configured for your environment.
2. Create a text file (with a `.properties` file extension).
3. Type the command, on a single line, specifying the location of your BAR deployment descriptor (typically `broker.xml`) and the file that contains the properties to be changed. A file with a `.bar` extension is created.



BAR file properties example

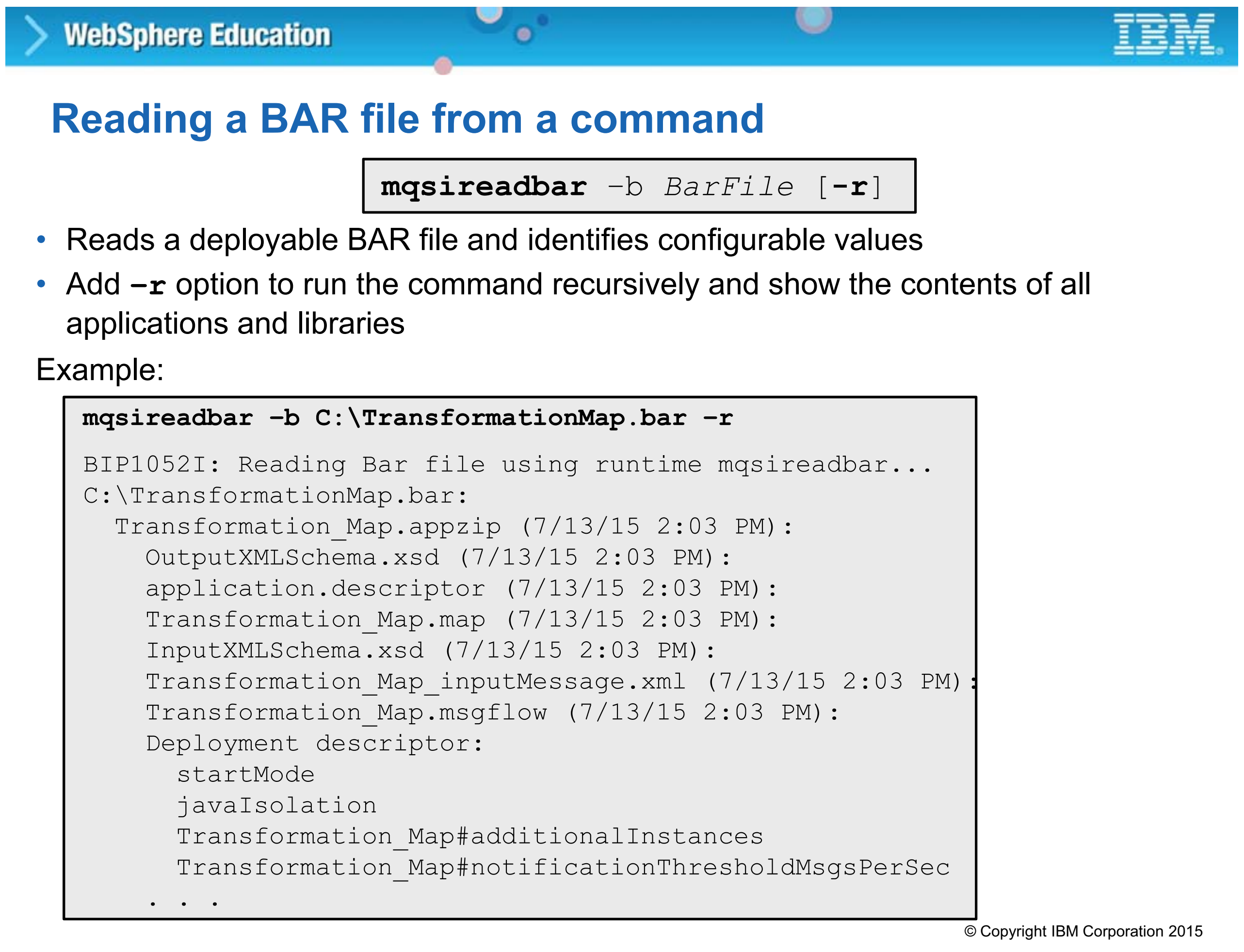
- Example properties file overrides the following BAR file properties:
  - In message flow *sampleFlow*, override the input queue on node *MQ Input*, to *NEW_INPUT_QUEUE*
  - In message flow *sampleFlow*, clear the value of the promoted property *queueName* that is set on the subflow node that represents *sampleSubflow1*
  - In subflow *sampleSubflow1*, clear the value of the flow-level promoted property *queueName*
  - In all message flows and subflows, override any properties with values previously set to *SUBOUT* to *NEWSUBOUT*

```
sampleFlow#MQ Input.queueName=NEW_INPUT_QUEUE
sampleFlow#sampleSubflow1.queueName
sampleSubflow1#queueName SUBOUT=NEW_SUBOUT
```

Figure 5-25. BAR file properties example WM6461.0

Notes:

This figure shows an example of a BAR file properties file.



WebSphere Education

Reading a BAR file from a command

```
mqsireadbar -b BarFile [-r]
```

- Reads a deployable BAR file and identifies configurable values
- Add **-r** option to run the command recursively and show the contents of all applications and libraries

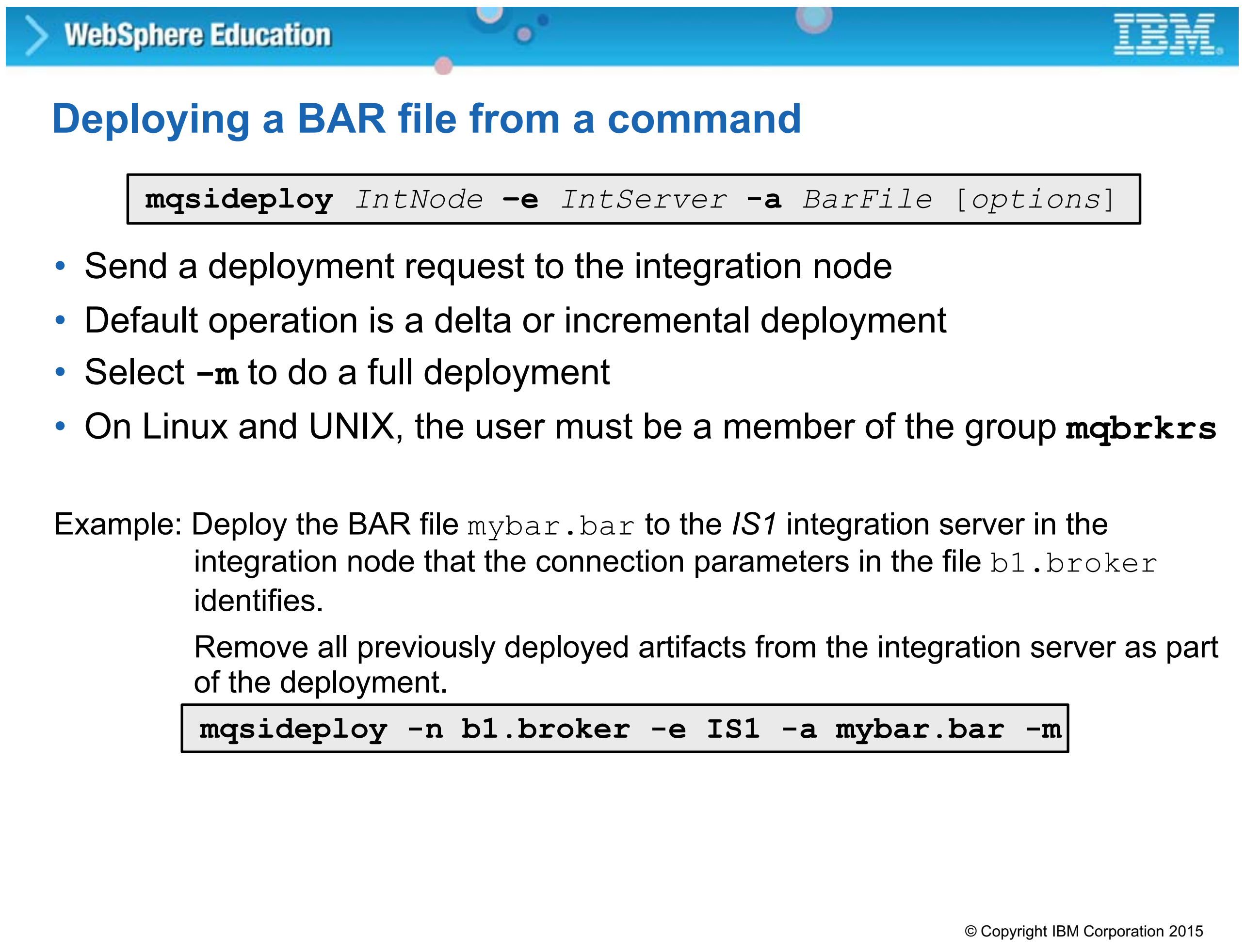
Example:

```
mqsireadbar -b C:\TransformationMap.bar -r

BIP1052I: Reading Bar file using runtime mqsireadbar...
C:\TransformationMap.bar:
  Transformation_Map.appzip (7/13/15 2:03 PM):
    OutputXMLSchema.xsd (7/13/15 2:03 PM):
    application.descriptor (7/13/15 2:03 PM):
    Transformation_Map.map (7/13/15 2:03 PM):
    InputXMLSchema.xsd (7/13/15 2:03 PM):
    Transformation_Map_inputMessage.xml (7/13/15 2:03 PM):
    Transformation_Map.msgflow (7/13/15 2:03 PM):
    Deployment descriptor:
      startMode
      javaIsolation
      Transformation_Map#additionalInstances
      Transformation_Map#notificationThresholdMsgsPerSec
    . . .
```

Figure 5-26. Reading a BAR file from a command WM6461.0

Notes:

The `mqsireadbar` command lists the keywords that are defined for each deployable file within a deployable BAR file. It also displays deployment descriptors and the list of any properties in the BAR file that can be overridden, together with the new values of any overrides that are currently applied.



WebSphere Education

Deploying a BAR file from a command

```
mqsideploy IntNode -e IntServer -a BarFile [options]
```

- Send a deployment request to the integration node
- Default operation is a delta or incremental deployment
- Select **`-m`** to do a full deployment
- On Linux and UNIX, the user must be a member of the group **`mqbrkrs`**

Example: Deploy the BAR file `mybar.bar` to the *IS1* integration server in the integration node that the connection parameters in the file `b1.broker` identifies.

Remove all previously deployed artifacts from the integration server as part of the deployment.

```
mqsideploy -n b1.broker -e IS1 -a mybar.bar -m
```

Figure 5-27. Deploying a BAR file from a command WM6461.0

Notes:

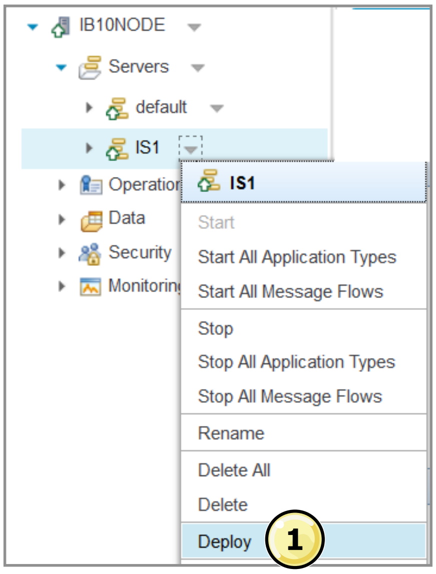
The `mqsideploy` command sends a deployment request. Typically this command is used to deploy a BAR file to an integration server. The `mqsideploy` command is used frequently in scripts.

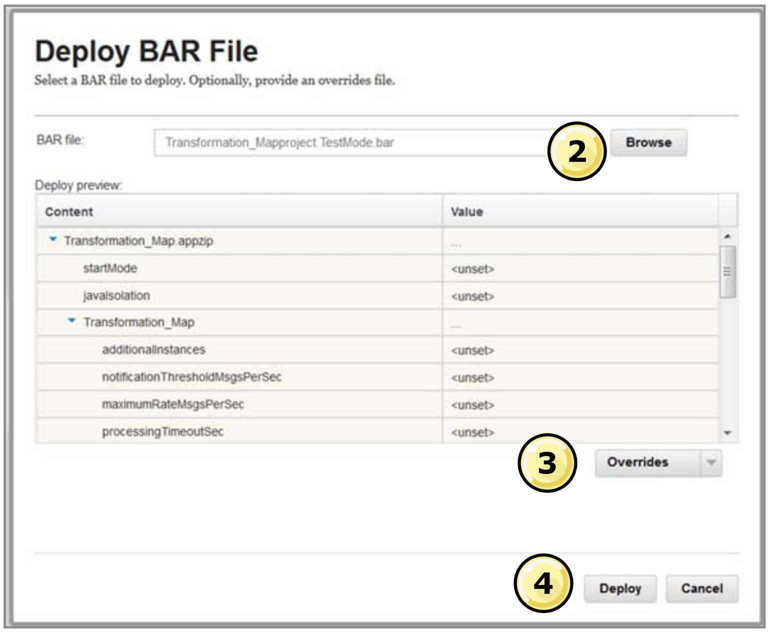
It is also possible to use the `mqsideploy` command to remove objects from an integration server. Use the `-d` flag followed by one of more objects, which are separated by a colon "`:`". For example, this command removes the two message flows that are called `Admin1` and `Admin2` from integration server that is named `IS1`.

```
mqsideploy -n IBNODE.node -e IS1 -d Admin1.msgflow:Admin2.msgflow
```

WebSphere Education

Deploying a BAR file from the Integration Bus web user interface




Deploy BAR File

Select a BAR file to deploy. Optionally, provide an overrides file.

BAR file: Transformation_Mapproject.TestMode.bar

Deploy preview:

Content	Value
Transformation_Map.appzip	...
startMode	<unset>
javaIsolation	<unset>
Transformation_Map	...
additionalInstances	<unset>
notificationThresholdMsgsPerSec	<unset>
maximumRateMsgsPerSec	<unset>
processingTimeoutSec	<unset>

1. Click the arrow by the integration and then click **Deploy**.
2. Click **Browse**, select to the BAR file, and then click **Open**.
3. (Optional) Select an overrides file.
4. Click **Deploy**.

Figure 5-28. Deploying a BAR file from the Integration Bus web user interface WM6461.0

Notes:

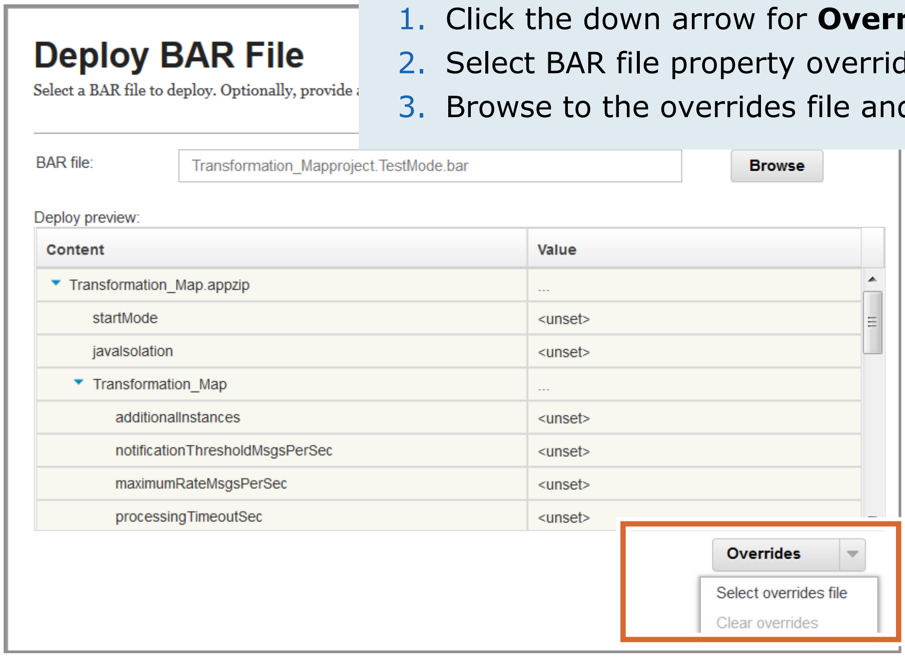
As shown in the figure, you can deploy a BAR file by using the Integration Bus web user interface.

You can also override options in the BAR file by using a BAR file properties override file.

WebSphere Education

Modifying a BAR file in the Integration Bus web user interface

- Uses **mqsiapplybaroverride** BAR file property overrides file
- On the **Deploy Bar File** window:
  1. Click the down arrow for **Overrides**
  2. Select BAR file property overrides file
  3. Browse to the overrides file and then click **Open**



Deploy BAR File

Select a BAR file to deploy. Optionally, provide

BAR file: Transformation_Mapproject.TestMode.bar

Deploy preview:

Content	Value
Transformation_Map.appzip	...
startMode	<unset>
javaIsolation	<unset>
Transformation_Map	...
additionalInstances	<unset>
notificationThresholdMsgsPerSec	<unset>
maximumRateMsgsPerSec	<unset>
processingTimeoutSec	<unset>

Figure 5-29. Modifying a BAR file in the Integration Bus web user interface WM6461.0

Notes:

You can override the properties in the BAR file by creating an overrides file and then selecting the file in the web user interface.

5.4. IBM Integration API overview

This topic introduces the IBM Integration API and the sample IBM Integration API Exerciser.

WebSphere Education IBM

IBM Integration Java API overview

- Java programming interface that applications can use to control integration nodes and resources
- Also known as the Configuration Manager Proxy (CMP) API
- Uses service-oriented type architecture
- JAR file **`IntegrationAPI.jar`** supplies all the classes
- Java classes sit logically between the user application and the integration node, inside the Java virtual machine (JVM) of the user application

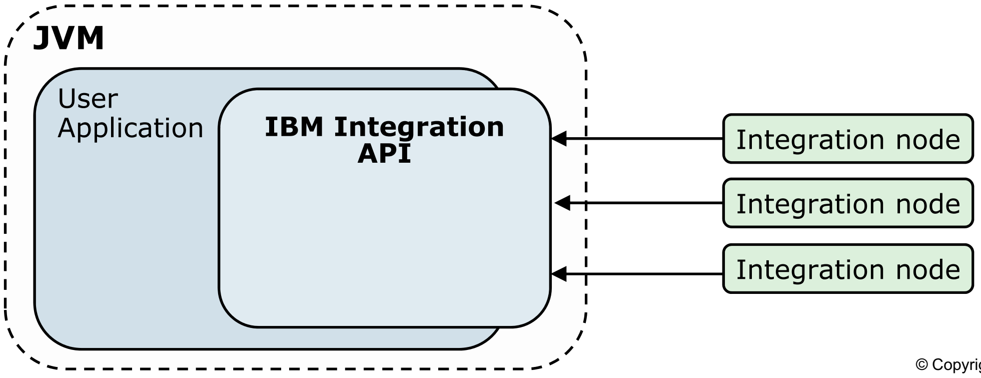
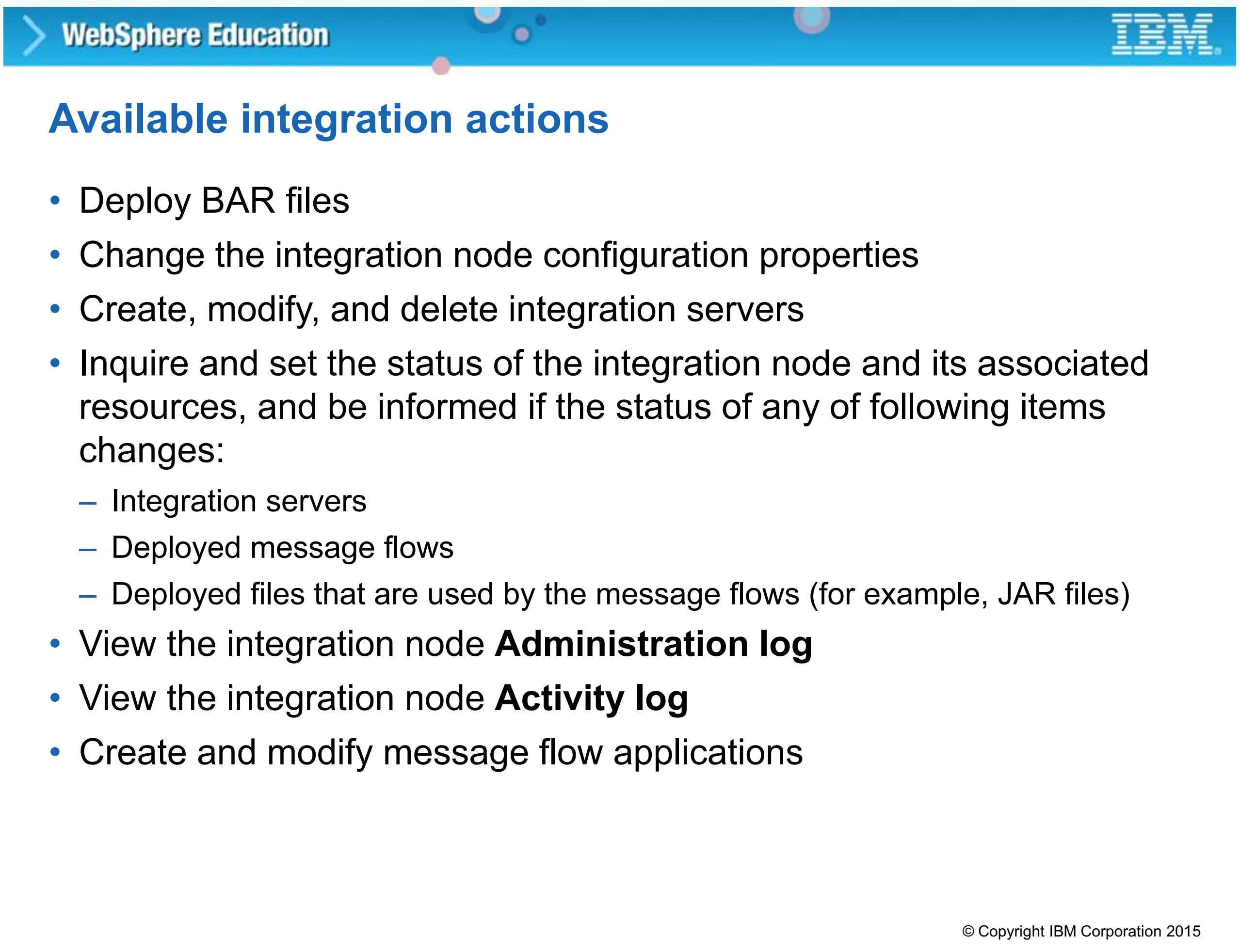


Figure 5-30. IBM Integration Java API overview WM6461.0

Notes:

The IBM Integration API is a programming interface that your applications can use to control integration nodes and their resources through a remote interface.

As shown in the figure, the API Java classes sit logically between the user application and the integration node, inside the Java virtual machine (JVM) of the user application.



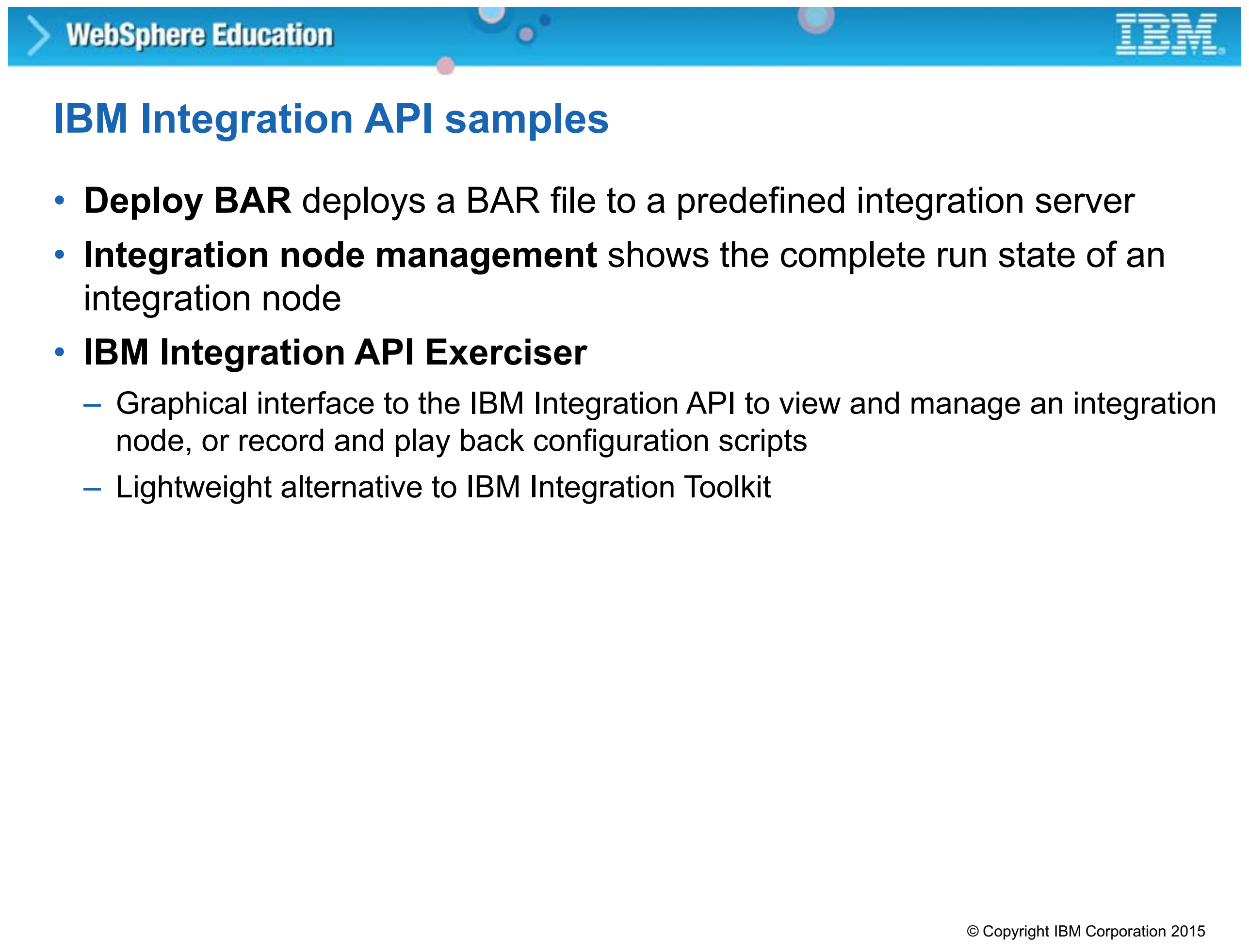
Available integration actions

- Deploy BAR files
- Change the integration node configuration properties
- Create, modify, and delete integration servers
- Inquire and set the status of the integration node and its associated resources, and be informed if the status of any of following items changes:
  - Integration servers
  - Deployed message flows
  - Deployed files that are used by the message flows (for example, JAR files)
- View the integration node **Administration log**
- View the integration node **Activity log**
- Create and modify message flow applications

Figure 5-31. Available integration actions WM6461.0

Notes:

This figure lists the actions that available in the IBM Integration API.



WebSphere Education

IBM Integration API samples

- **Deploy BAR** deploys a BAR file to a predefined integration server
- **Integration node management** shows the complete run state of an integration node
- **IBM Integration API Exerciser**
  - Graphical interface to the IBM Integration API to view and manage an integration node, or record and play back configuration scripts
  - Lightweight alternative to IBM Integration Toolkit

Figure 5-32. IBM Integration API samples WM6461.0

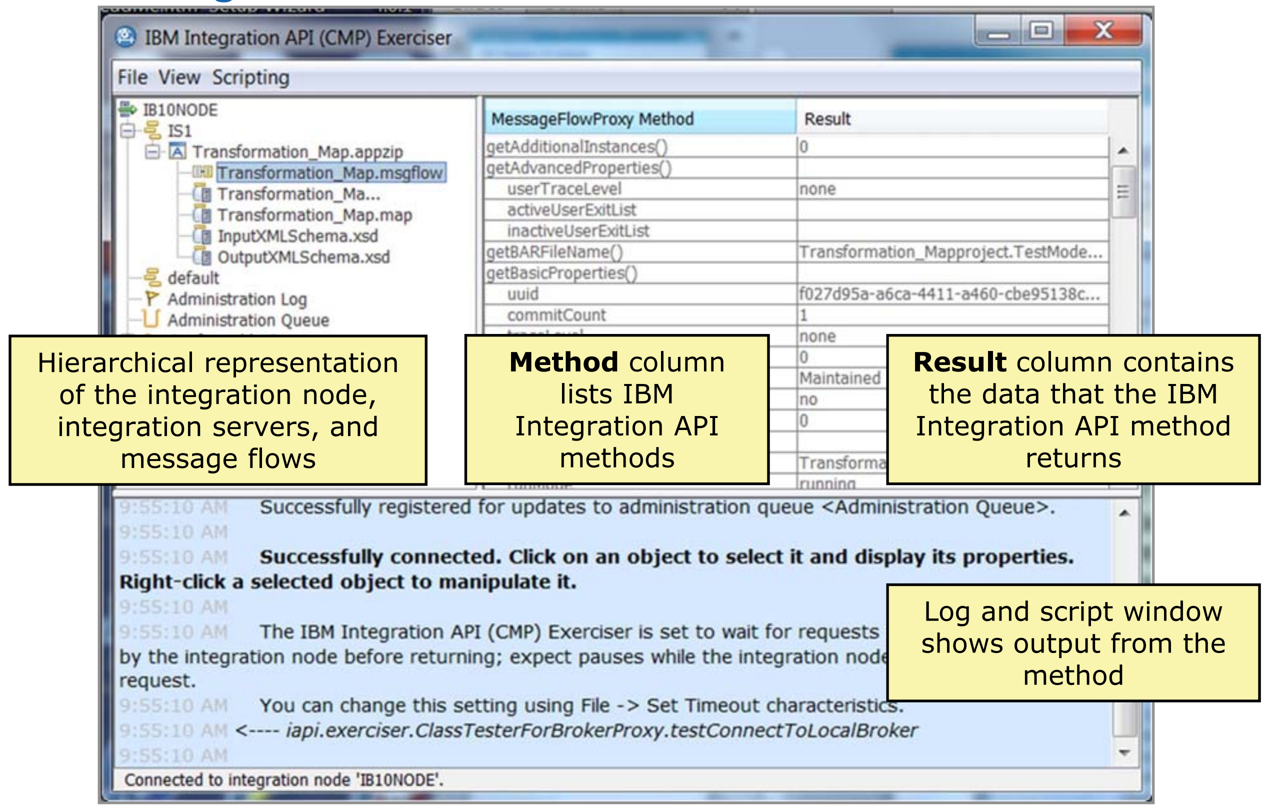
Notes:

You can explore the IBM Integration Bus samples to learn the basic features that the Integration API provides.

The listed samples are available for use with the product and are documented in the IBM Knowledge Center for IBM Integration Bus. They can also serve as a basis for applications that you might want to develop to automate more of your administration work.

WebSphere Education

IBM Integration API Exerciser


IBM Integration API (CMP) Exerciser
File View Scripting
IB10NODE
IS1
Transformation_Map.appzip
Transformation_Map.msgflow
Transformation_Ma...
Transformation_Map.map
InputXMLSchema.xsd
OutputXMLSchema.xsd
default
Administration Log
Administration Queue
Hierarchical representation of the integration node, integration servers, and message flows
Method column lists IBM Integration API methods
Result column contains the data that the IBM Integration API method returns
Log and script window shows output from the method

MessageFlowProxy Method	Result
getAdditionalInstances()	0
getAdvancedProperties()	
userTraceLevel	none
activeUserExitList	
inactiveUserExitList	
getBARFileName()	Transformation_Mapproject.TestMode...
getBasicProperties()	
uuid	f027d95a-a6ca-4411-a460-cbe95138c...
commitCount	1
	none
	0
	Maintained
	no
	0
	Transforma
	running

```
9:55:10 AM   Successfully registered for updates to administration queue <Administration Queue>.
9:55:10 AM
9:55:10 AM   Successfully connected. Click on an object to select it and display its properties.
Right-click a selected object to manipulate it.
9:55:10 AM
9:55:10 AM   The IBM Integration API (CMP) Exerciser is set to wait for requests
by the integration node before returning; expect pauses while the integration node
request.
9:55:10 AM   You can change this setting using File -> Set Timeout characteristics.
9:55:10 AM <---- iapi.exerciser.ClassTesterForBrokerProxy.testConnectToLocalBroker
9:55:10 AM
Connected to integration node 'IB10NODE'.
```

© Copyright IBM Corporation 2015

Figure 5-33. IBM Integration API Exerciser WM6461.0

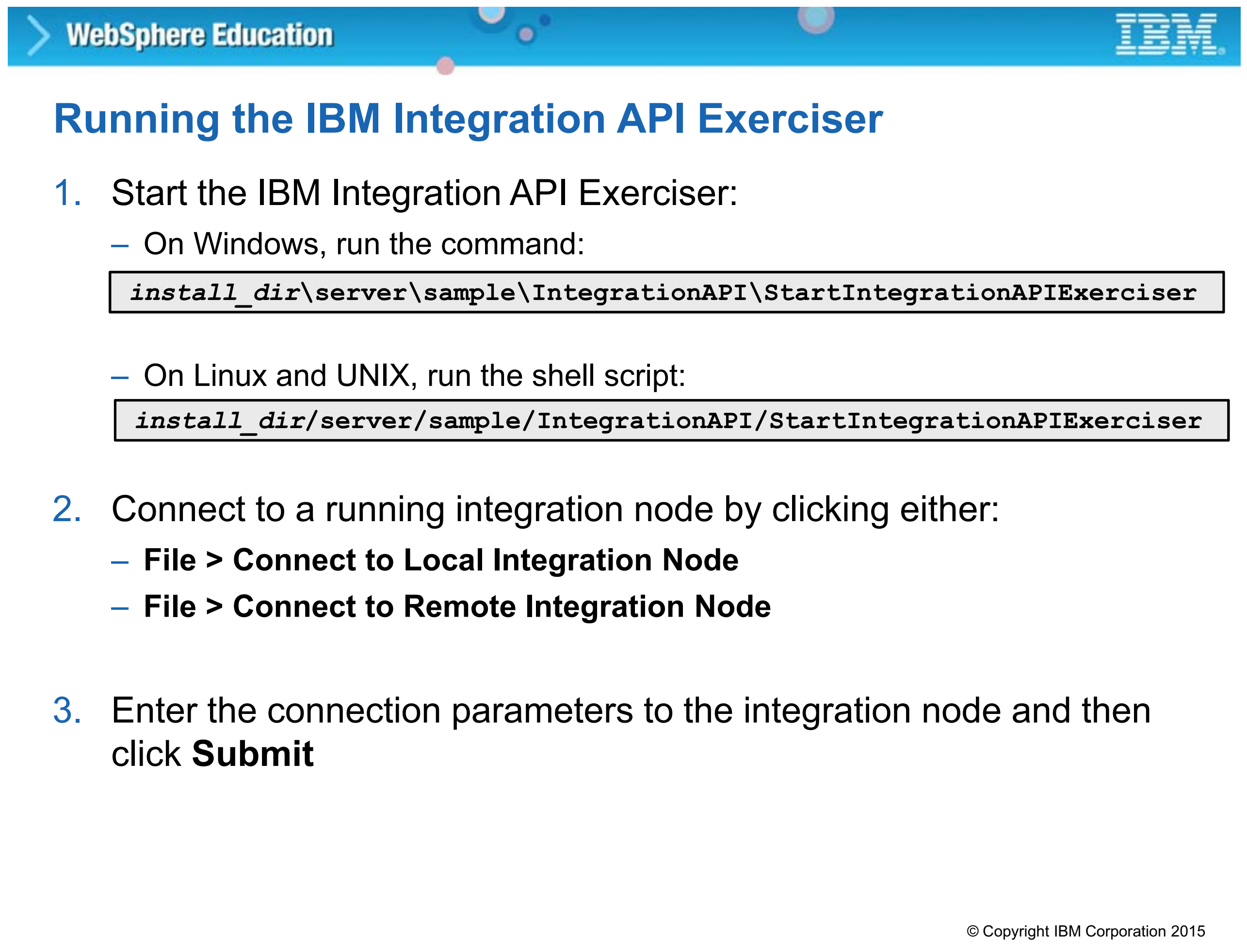
Notes:

You can use the Integration API Exerciser to view and manage an integration node, or to record and test configuration scripts.

The upper-left portion of the Integration API Exerciser contains a hierarchical representation of the integration node to which you are connected. Selecting objects in the tree causes the table on the right to change, reflecting the attributes of the object that you select.

The **Method** column lists Application API methods that you can call in your own Java applications.

The **Result** column indicates the data that is returned by calling the Application API method on the selected object.



WebSphere Education

Running the IBM Integration API Exerciser

1. Start the IBM Integration API Exerciser:
   – On Windows, run the command:

```
install_dir\server\sample\IntegrationAPI\StartIntegrationAPIExerciser
```

   – On Linux and UNIX, run the shell script:

```
install_dir/server/sample/IntegrationAPI/StartIntegrationAPIExerciser
```

2. Connect to a running integration node by clicking either:
   – **File > Connect to Local Integration Node**
   – **File > Connect to Remote Integration Node**

3. Enter the connection parameters to the integration node and then click **Submit**

© Copyright IBM Corporation 2015

Figure 5-34. Running the IBM Integration API Exerciser WM6461.0

Notes:

The figure lists the steps for starting the Integration API Exerciser on Windows, Linux, and UNIX.

In the command, `install_dir` is the Integration Bus installation directory.

On Linux and UNIX, ensure that the user ID has write permission to the current directory. The IBM Integration API Exerciser stores its configuration settings in a file in this directory.

5.5. Backing up and restoring resources

This topic describes the procedures for backing up and restoring IBM Integration Bus resources.

Back up and restore

- Back up and restore commands for disaster recovery
- Operational state is not saved
- Backs up configuration state from the file system
  - Internally managed integration node configuration store
  - Artifacts that are deployed from BAR files
  - Shared configuration
  - Registry
- Online backups are supported

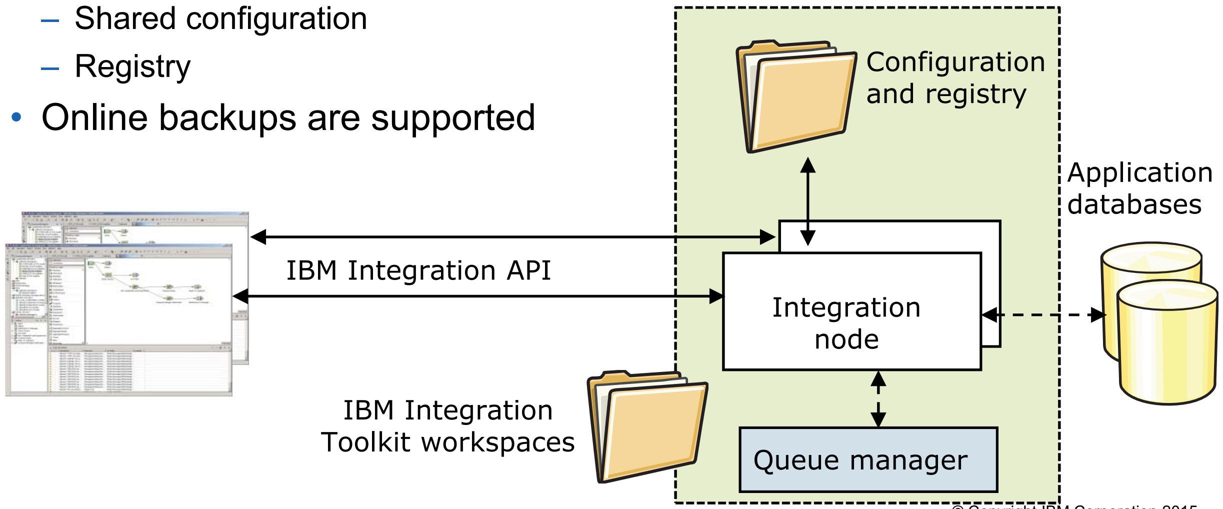


Figure 5-35. Back up and restore WM6461.0

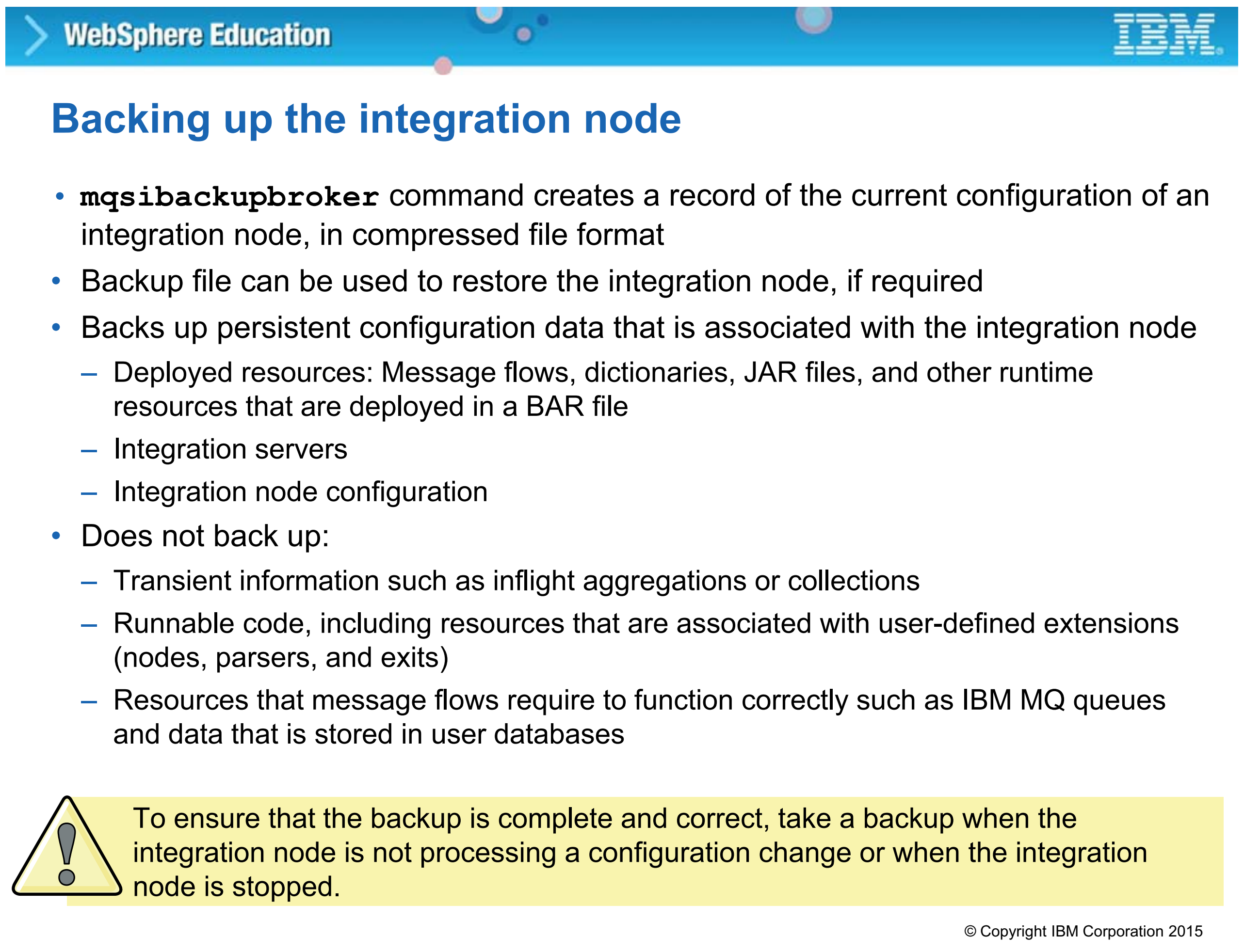
Notes:

Planning a backup and recovery strategy is an important administrative task.

IBM Integration Bus contains two commands for disaster recovery: `mqsibackupbroker` and `mqsirestorebroker`

Each command requires the integration node name and a location on disk for the archive file.

If it is necessary to restore, you can restore directly from the backup file if the integration node name and operating system match the original operating system and integration node name.



WebSphere Education

Backing up the integration node

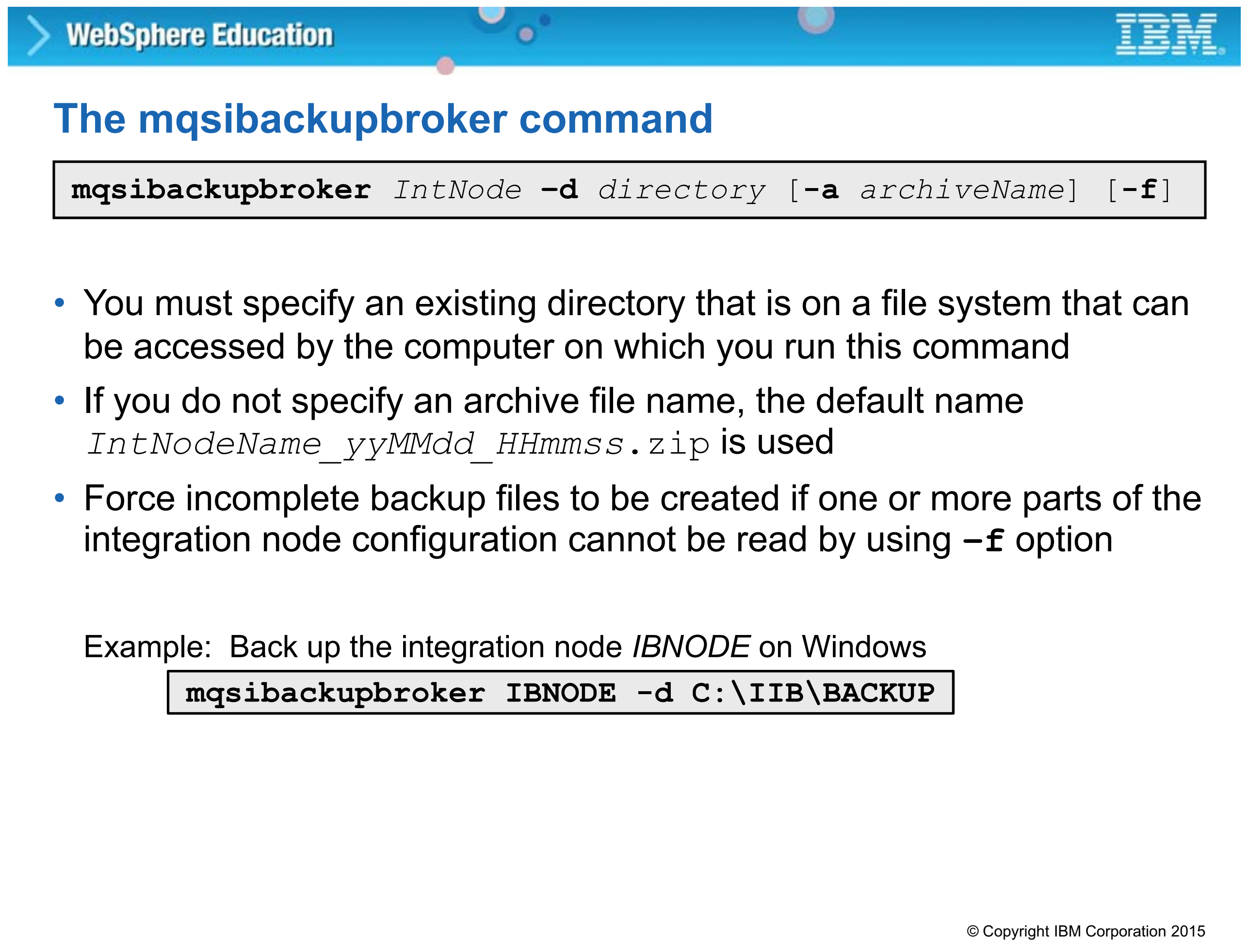
- **`mqsibackupbroker`** command creates a record of the current configuration of an integration node, in compressed file format
- Backup file can be used to restore the integration node, if required
- Backs up persistent configuration data that is associated with the integration node
  - Deployed resources: Message flows, dictionaries, JAR files, and other runtime resources that are deployed in a BAR file
  - Integration servers
  - Integration node configuration
- Does not back up:
  - Transient information such as inflight aggregations or collections
  - Runnable code, including resources that are associated with user-defined extensions (nodes, parsers, and exits)
  - Resources that message flows require to function correctly such as IBM MQ queues and data that is stored in user databases

To ensure that the backup is complete and correct, take a backup when the integration node is not processing a configuration change or when the integration node is stopped.

Figure 5-36. Backing up the integration node WM6461.0

Notes:

The backup command creates a compressed file that contains the internal configuration store, registry, and artifacts from the BAR file such as stylesheets. Backup can run while the integration node is running.



WebSphere Education IBM

The mqsibackupbroker command

```
mqsibackupbroker IntNode -d directory [-a archiveName] [-f]
```

- You must specify an existing directory that is on a file system that can be accessed by the computer on which you run this command
- If you do not specify an archive file name, the default name `IntNodeName_yyMMdd_HHmmss.zip` is used
- Force incomplete backup files to be created if one or more parts of the integration node configuration cannot be read by using **`-f`** option

Example: Back up the integration node *IBNODE* on Windows

```
mqsibackupbroker IBNODE -d C:\IIB\BACKUP
```

Figure 5-37. The mqsibackupbroker command WM6461.0

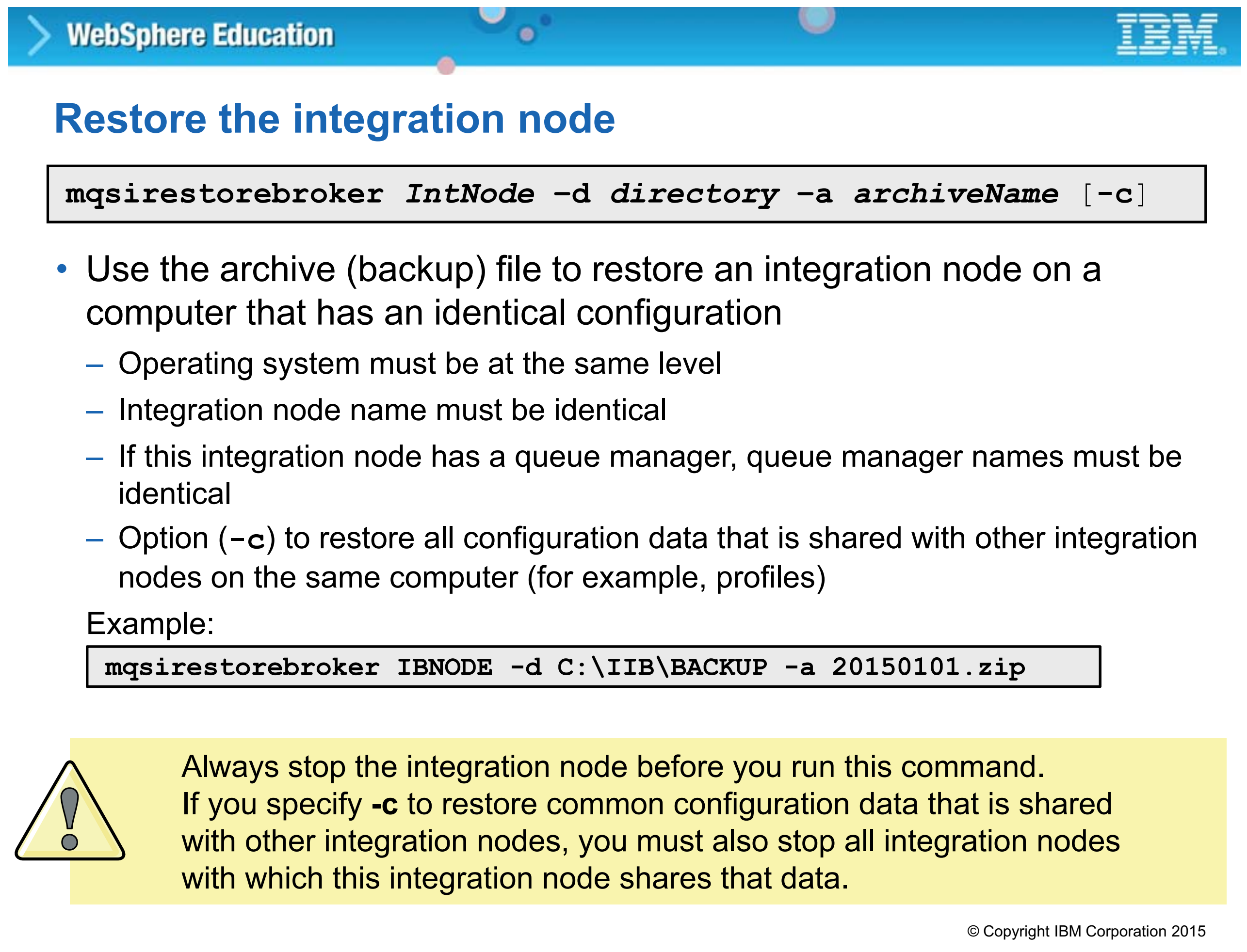
Notes:

The backup command creates a compressed file that contains the internal configuration store, registry, and artifacts from the BAR file such as stylesheets. Backup can run while the integration node is running.

The `mqsibackupbroker` command includes the following parameters:

`-`d is the required directory in which the backup file is created. You must specify a directory that is on a file system that is accessible by the computer on which you run this command.

`-`a is the optional backup (archive) file name. The file is created in the compressed file format. If you do not specify this parameter, the default name *`IntNodeName_yyMMdd_HHmmss`*`.zip` is used.



WebSphere Education IBM

Restore the integration node

```
mqsirestorebroker IntNode -d directory -a archiveName [-c]
```

- Use the archive (backup) file to restore an integration node on a computer that has an identical configuration
  - Operating system must be at the same level
  - Integration node name must be identical
  - If this integration node has a queue manager, queue manager names must be identical
  - Option (`-c`) to restore all configuration data that is shared with other integration nodes on the same computer (for example, profiles)

  Example:

```
mqsirestorebroker IBNODE -d C:\IIB\BACKUP -a 20150101.zip
```

Always stop the integration node before you run this command.
If you specify **-c** to restore common configuration data that is shared with other integration nodes, you must also stop all integration nodes with which this integration node shares that data.

© Copyright IBM Corporation 2015

Figure 5-38. Restore the integration node WM6461.0

Notes:

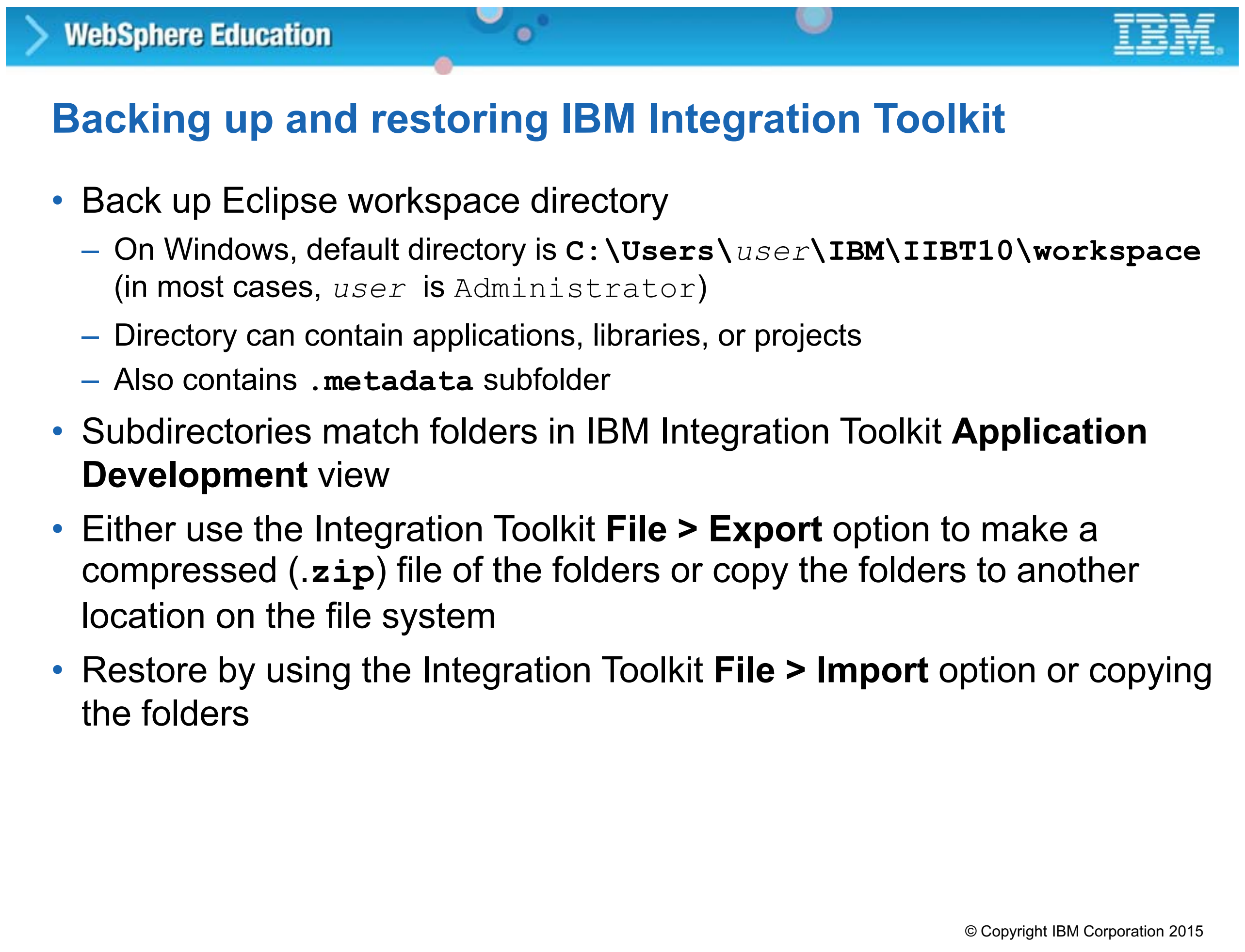
If it is necessary to restore, you can restore directly from the backup file if the integration node name and operating system match the original integration node and operating system.

The `mqsirestorebroker` command includes the following parameters:

> `-d` is the directory in which the backup file is stored. The file must be stored in a file system that is accessible from the computer on which you run this command.
>
> `-a` is the name of the backup (archive) file.
>
> `-c` restores all configuration data that is shared with other integration nodes on the same computer; for example, profiles.

Always stop the integration node before you run this command. If you specify `-c` to restore common configuration data that is shared with other integration nodes, you must also stop all integration nodes with which this integration node shares that data.

If you run this command when an integration node is active, the results are unpredictable.



WebSphere Education

Backing up and restoring IBM Integration Toolkit

- Back up Eclipse workspace directory
  - On Windows, default directory is **`C:\Users\`***`user`***`\IBM\IIBT10\workspace`** (in most cases, *`user`* is `Administrator`)
  - Directory can contain applications, libraries, or projects
  - Also contains **`.metadata`** subfolder
- Subdirectories match folders in IBM Integration Toolkit **Application Development** view
- Either use the Integration Toolkit **File > Export** option to make a compressed (**`.zip`**) file of the folders or copy the folders to another location on the file system
- Restore by using the Integration Toolkit **File > Import** option or copying the folders

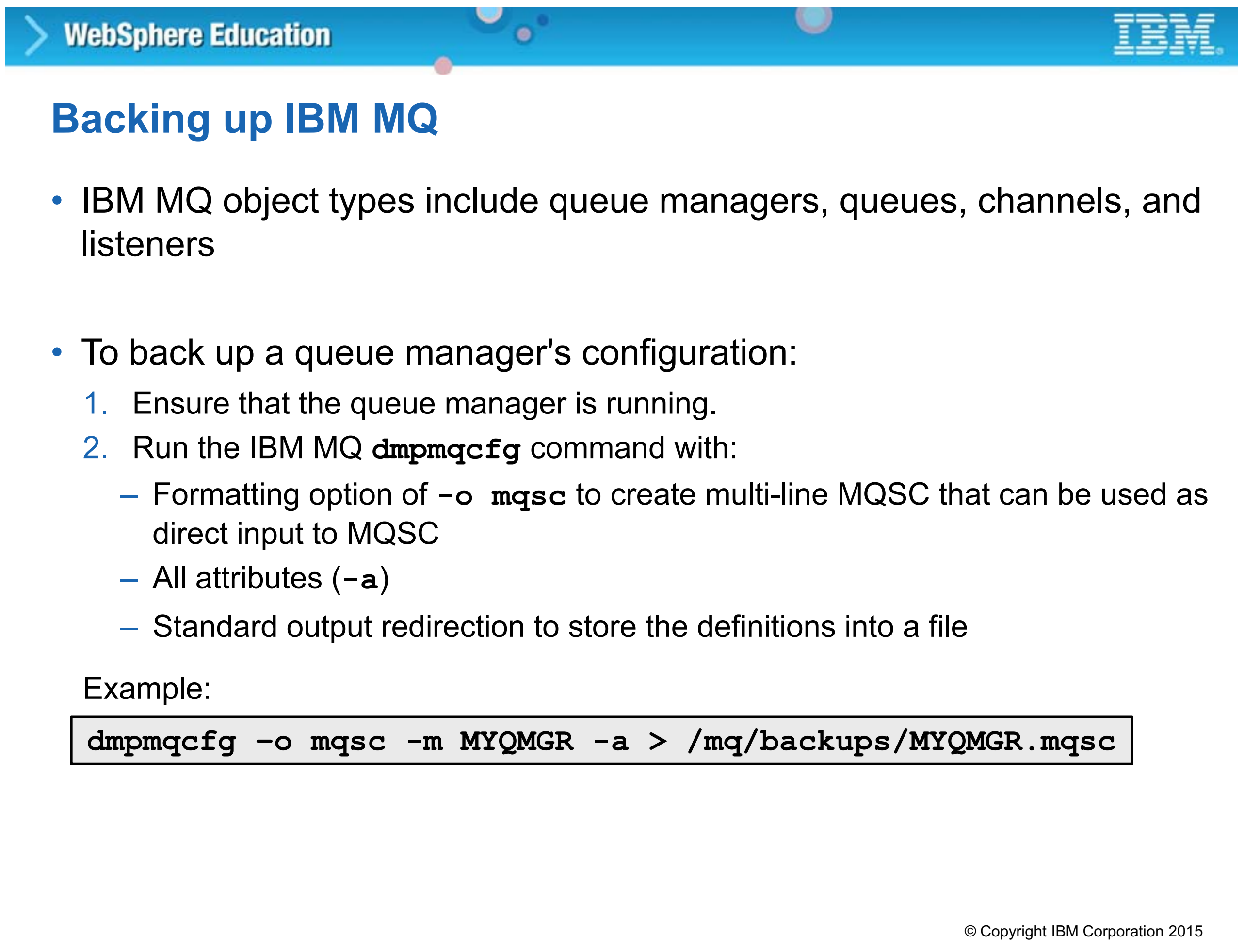
© Copyright IBM Corporation 2015

Figure 5-39. Backing up and restoring IBM Integration Toolkit WM6461.0

Notes:

You must plan to back up your entire workspace directory (or directories) because the IBM Integration Toolkit is made up of many files. It is possible to export projects by using the **File > Export** option or you can copy the files. The **Export** option generates a compressed file, which is easier to manage.

The workspace contains a `.metadata` subfolder, which is essential because it contains all the settings, plug-ins, references to projects, and other objects. However, projects are not necessarily subfolders in the workspace directory. They can exist anywhere in the (local or shared) file system.



WebSphere Education

Backing up IBM MQ

- IBM MQ object types include queue managers, queues, channels, and listeners

- To back up a queue manager's configuration:
  1. Ensure that the queue manager is running.
  2. Run the IBM MQ **`dmpmqcfg`** command with:
     - Formatting option of **`-o mqsc`** to create multi-line MQSC that can be used as direct input to MQSC
     - All attributes (**`-a`**)
     - Standard output redirection to store the definitions into a file

  Example:

```
dmpmqcfg -o mqsc -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

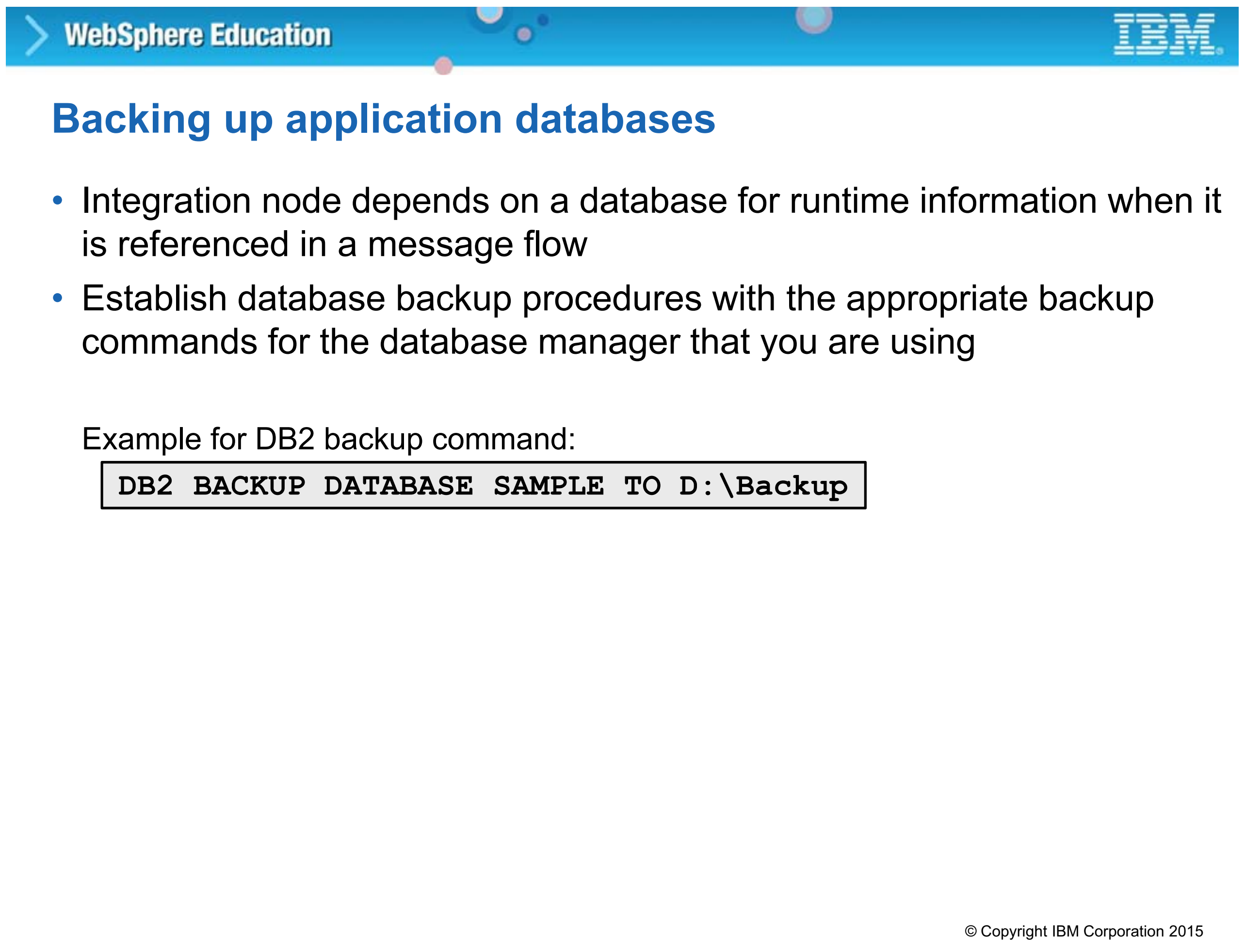
© Copyright IBM Corporation 2015

Figure 5-40. Backing up IBM MQ WM6461.0

Notes:

If the integration node is associated with an IBM MQ queue manager, back up the queue manager, queues, channels, and listeners.

For more information about backing up IBM MQ resources, see the IBM MQ product documentation.



WebSphere Education

Backing up application databases

- Integration node depends on a database for runtime information when it is referenced in a message flow
- Establish database backup procedures with the appropriate backup commands for the database manager that you are using

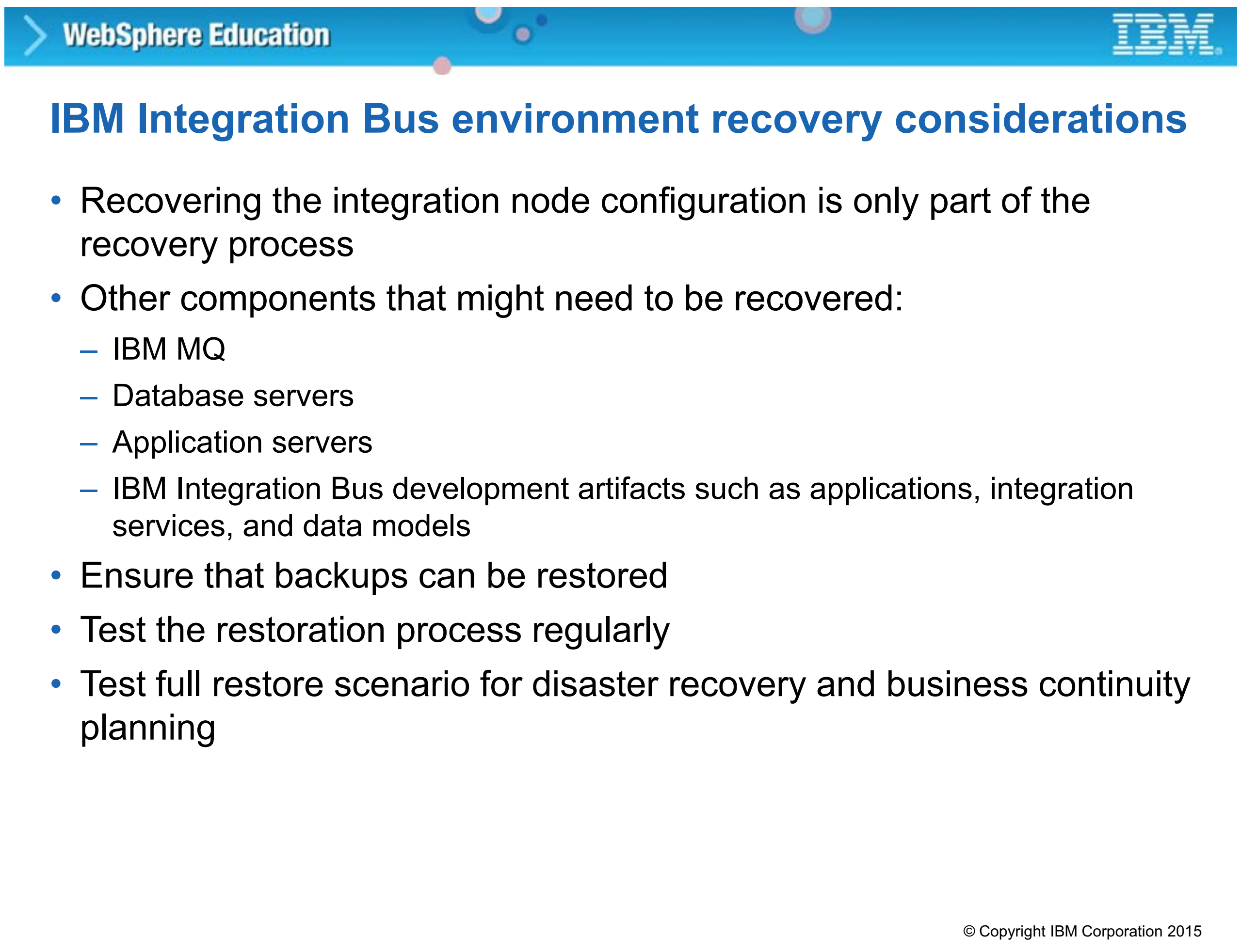
Example for DB2 backup command:

```
DB2 BACKUP DATABASE SAMPLE TO D:\Backup
```

Figure 5-41. Backing up application databases WM6461.0

Notes:

If your message flows access user databases through an ODBC or JDBC connection, back up the files that you use for these connections. Take a copy of these files and store them safely in a different location.



IBM Integration Bus environment recovery considerations

- Recovering the integration node configuration is only part of the recovery process
- Other components that might need to be recovered:
  - IBM MQ
  - Database servers
  - Application servers
  - IBM Integration Bus development artifacts such as applications, integration services, and data models
- Ensure that backups can be restored
- Test the restoration process regularly
- Test full restore scenario for disaster recovery and business continuity planning

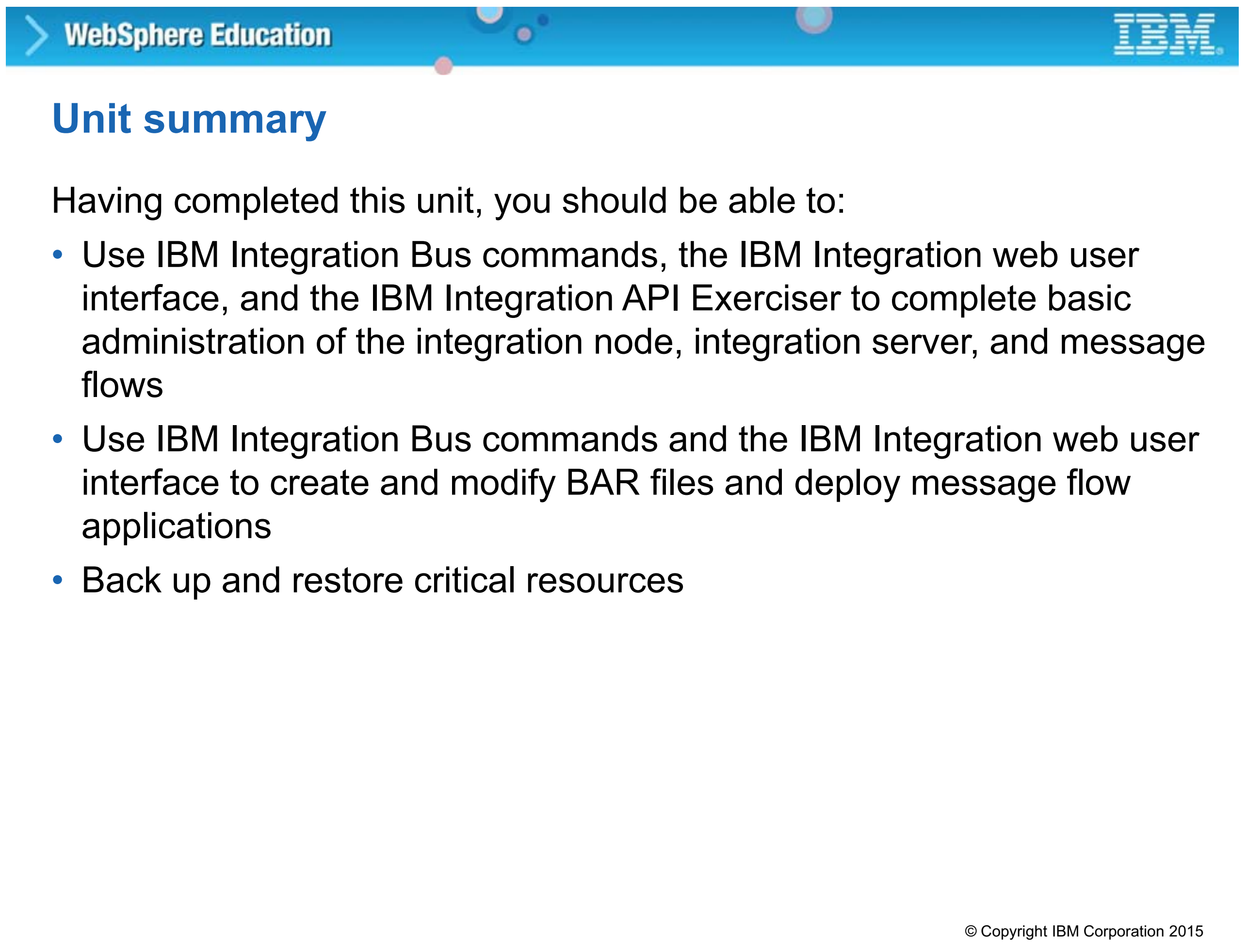
Figure 5-42. IBM Integration Bus environment recovery considerations WM6461.0

Notes:

If a critical component in an Integration Bus environment fails, consider what you must do to restore the environment. You know how to back up and restore an integration node configuration, but a failure typically affects more than just the integration node; it affects every application that accesses that integration node. For example:

- An application server might need to be recovered or restarted, or the server workload might need to be reset or resynchronized with processing that already completed.
- Application databases that the message flow applications use must be available. Again, consider what happens to in-flight transactions at the time of a failure.
- Although you can run the integration node environment without the supporting source artifacts (message flows, message definitions, and other objects), it might be necessary to restore those components as part of the recovery process. The source code repository must be as current as the integration node environment.

Ensure that the configuration backups that you create with the `mqsibackupbroker` command are available (not on the same system where the integration node is located). You must also ensure that the backups can be restored successfully.



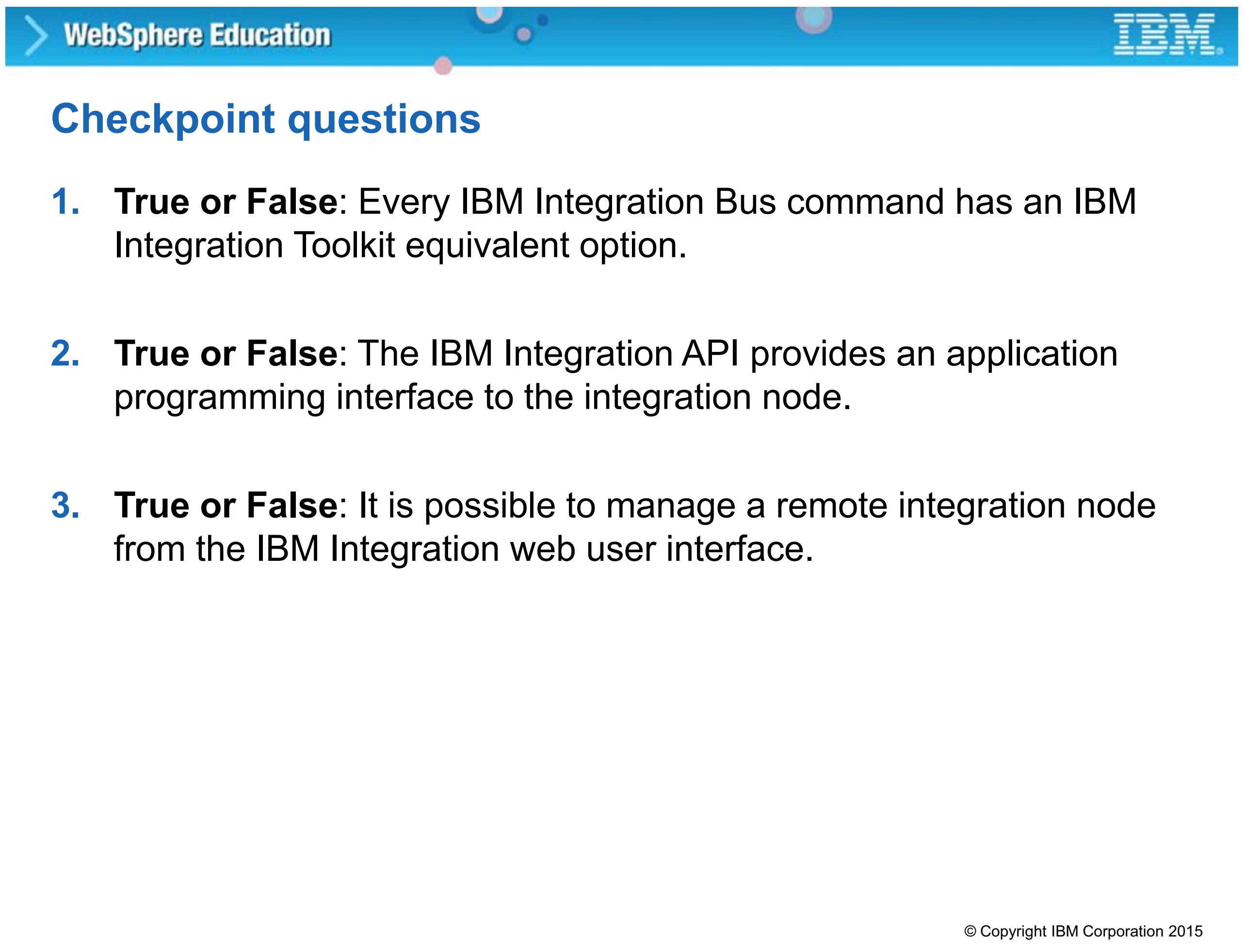
Unit summary

Having completed this unit, you should be able to:

- Use IBM Integration Bus commands, the IBM Integration web user interface, and the IBM Integration API Exerciser to complete basic administration of the integration node, integration server, and message flows
- Use IBM Integration Bus commands and the IBM Integration web user interface to create and modify BAR files and deploy message flow applications
- Back up and restore critical resources

Figure 5-43. Unit summary WM6461.0

Notes:



WebSphere Education IBM.

Checkpoint questions

1. **True or False**: Every IBM Integration Bus command has an IBM Integration Toolkit equivalent option.

2. **True or False**: The IBM Integration API provides an application programming interface to the integration node.

3. **True or False**: It is possible to manage a remote integration node from the IBM Integration web user interface.

© Copyright IBM Corporation 2015

Figure 5-44. Checkpoint questions WM6461.0

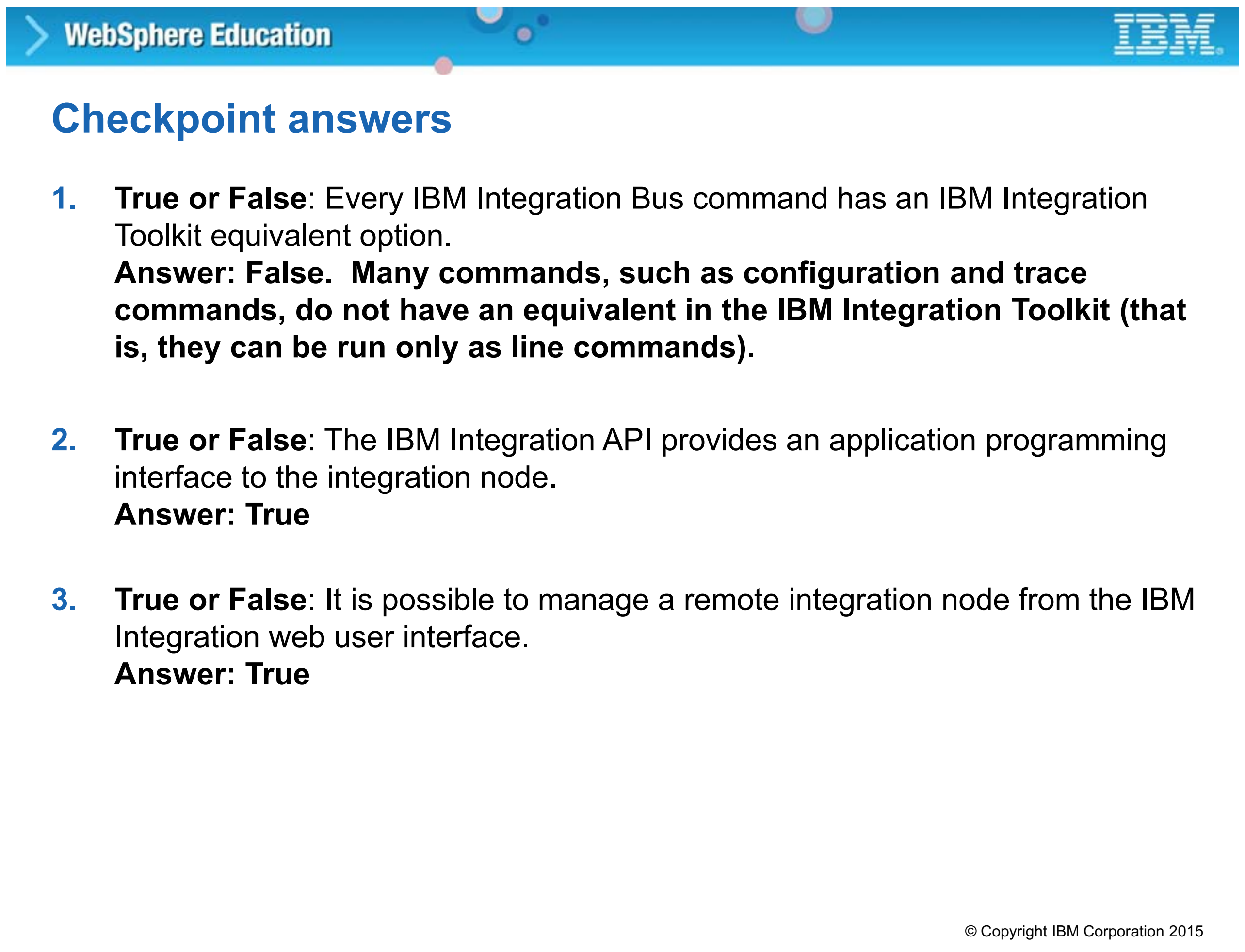
Notes:

Write your answers here:

1.

2.

3.



WebSphere Education

Checkpoint answers

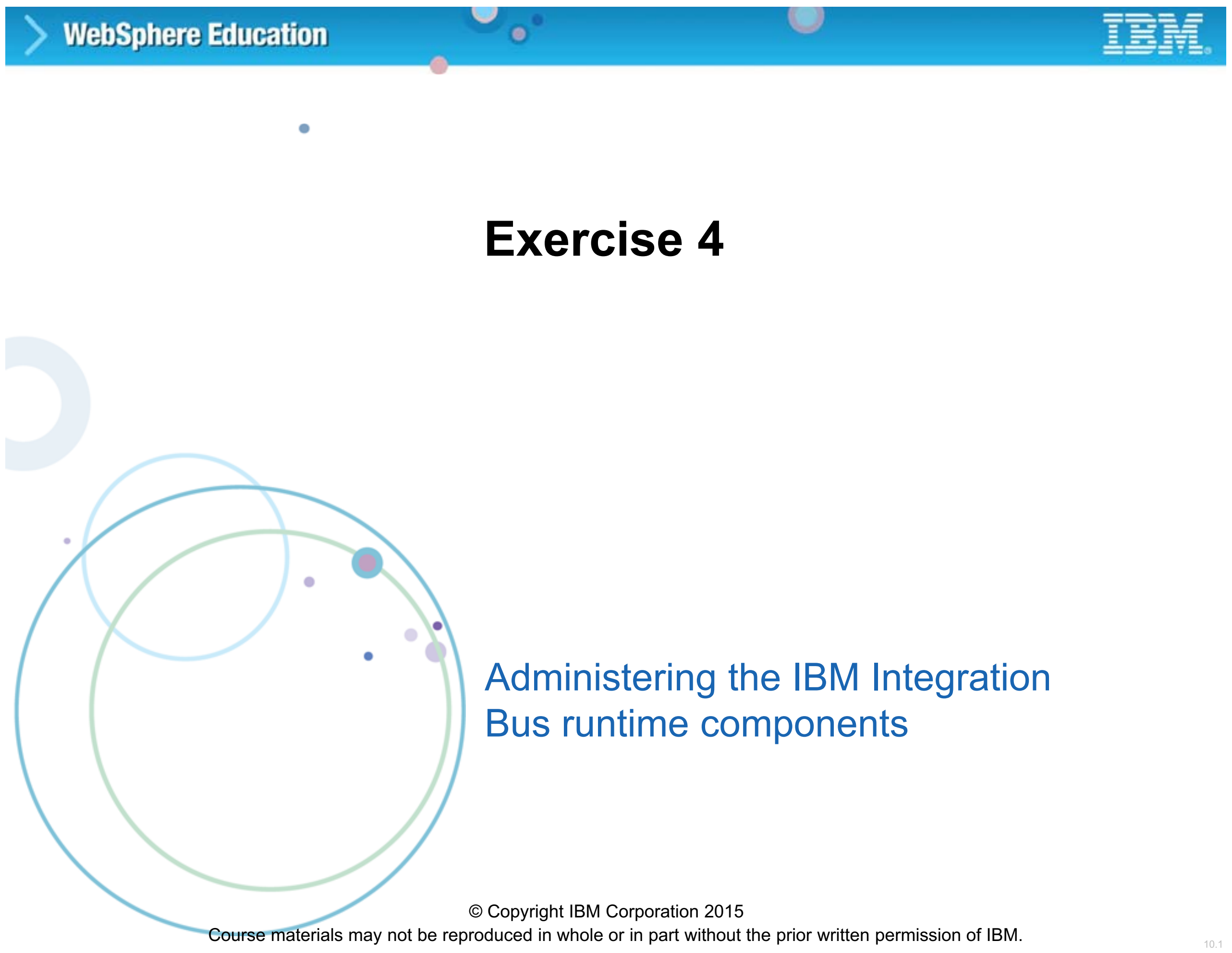
1. **True or False**: Every IBM Integration Bus command has an IBM Integration Toolkit equivalent option.
   Answer: False. Many commands, such as configuration and trace commands, do not have an equivalent in the IBM Integration Toolkit (that is, they can be run only as line commands).

2. **True or False**: The IBM Integration API provides an application programming interface to the integration node.
   Answer: True

3. **True or False**: It is possible to manage a remote integration node from the IBM Integration web user interface.
   Answer: True

Figure 5-45. Checkpoint answers WM6461.0

Notes:



WebSphere Education

IBM

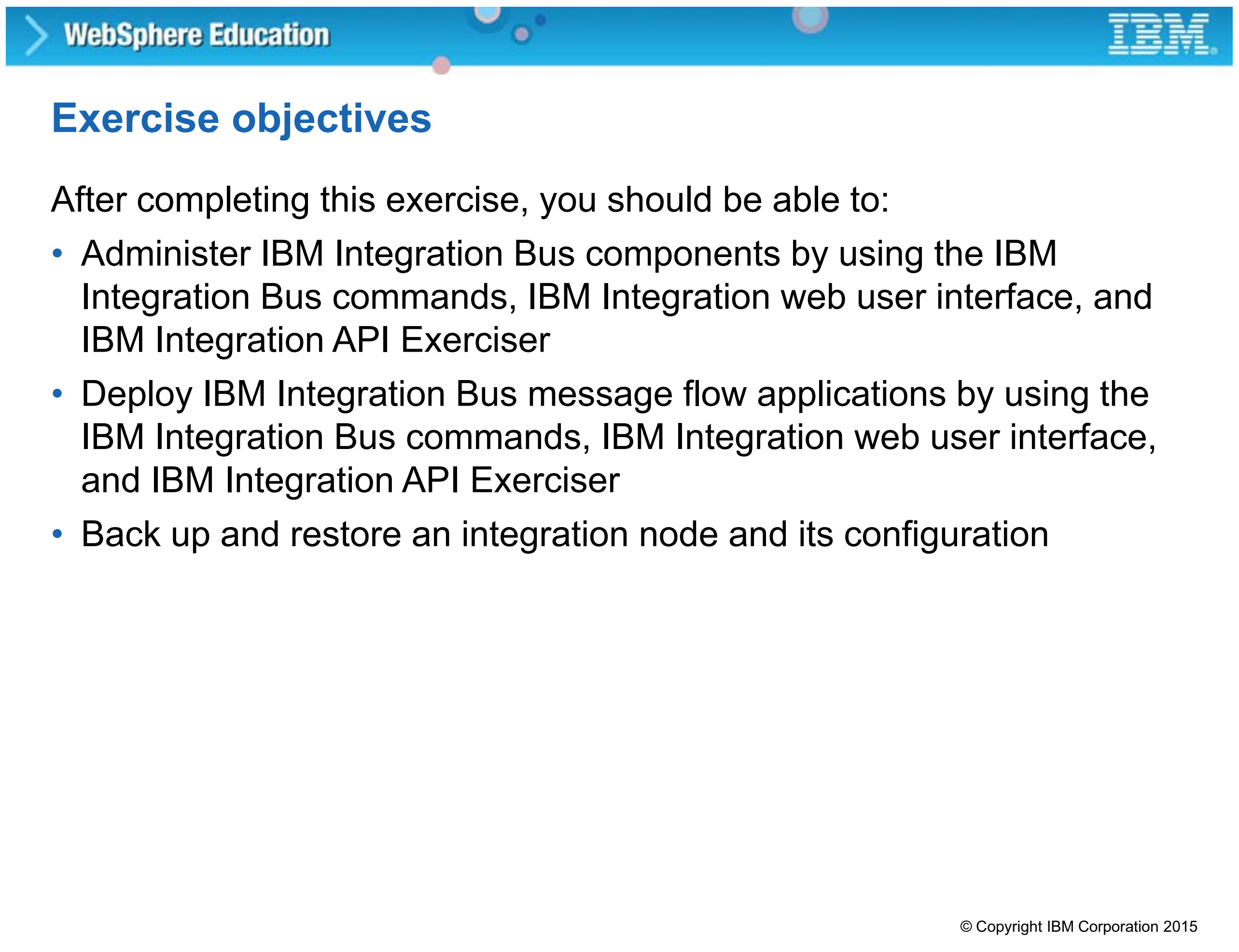
Exercise 4

Administering the IBM Integration Bus runtime components

Figure 5-46. Exercise 4 WM6461.0

Notes:

In this exercise, you use the IBM Integration Bus commands and IBM Integration web user interface to manage IBM Integration Bus runtime components. You also back up and restore the integration node configuration data.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Administer IBM Integration Bus components by using the IBM Integration Bus commands, IBM Integration web user interface, and IBM Integration API Exerciser
- Deploy IBM Integration Bus message flow applications by using the IBM Integration Bus commands, IBM Integration web user interface, and IBM Integration API Exerciser
- Back up and restore an integration node and its configuration

© Copyright IBM Corporation 2015

Figure 5-47. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 6. Implementing IBM Integration Bus administration security

What this unit is about

This unit describes how to implement integration node component security and integration node administration security. It also describes how to configure access authority for the IBM Integration web user interface.

What you should be able to do

After completing this unit, you should be able to:

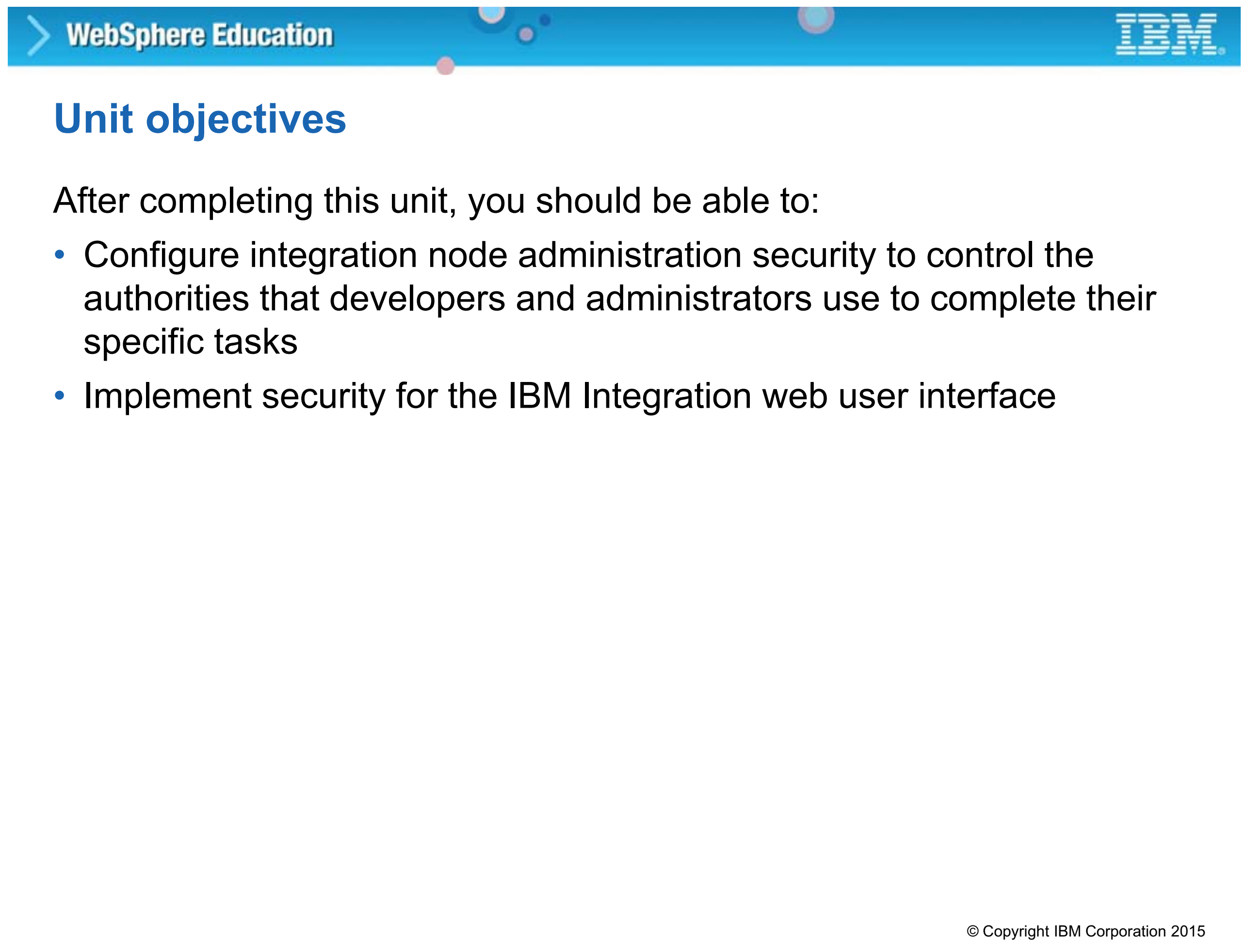
- Configure integration node administration security to control the authorities that developers and administrators use to complete their specific tasks
- Implement security for the IBM Integration web user interface

How you will check your progress

- Checkpoints
- Exercise: Using file-based security to control administration access
- Exercise: Using queue-based security to control administration access

References

IBM Knowledge Center



WebSphere Education

Unit objectives

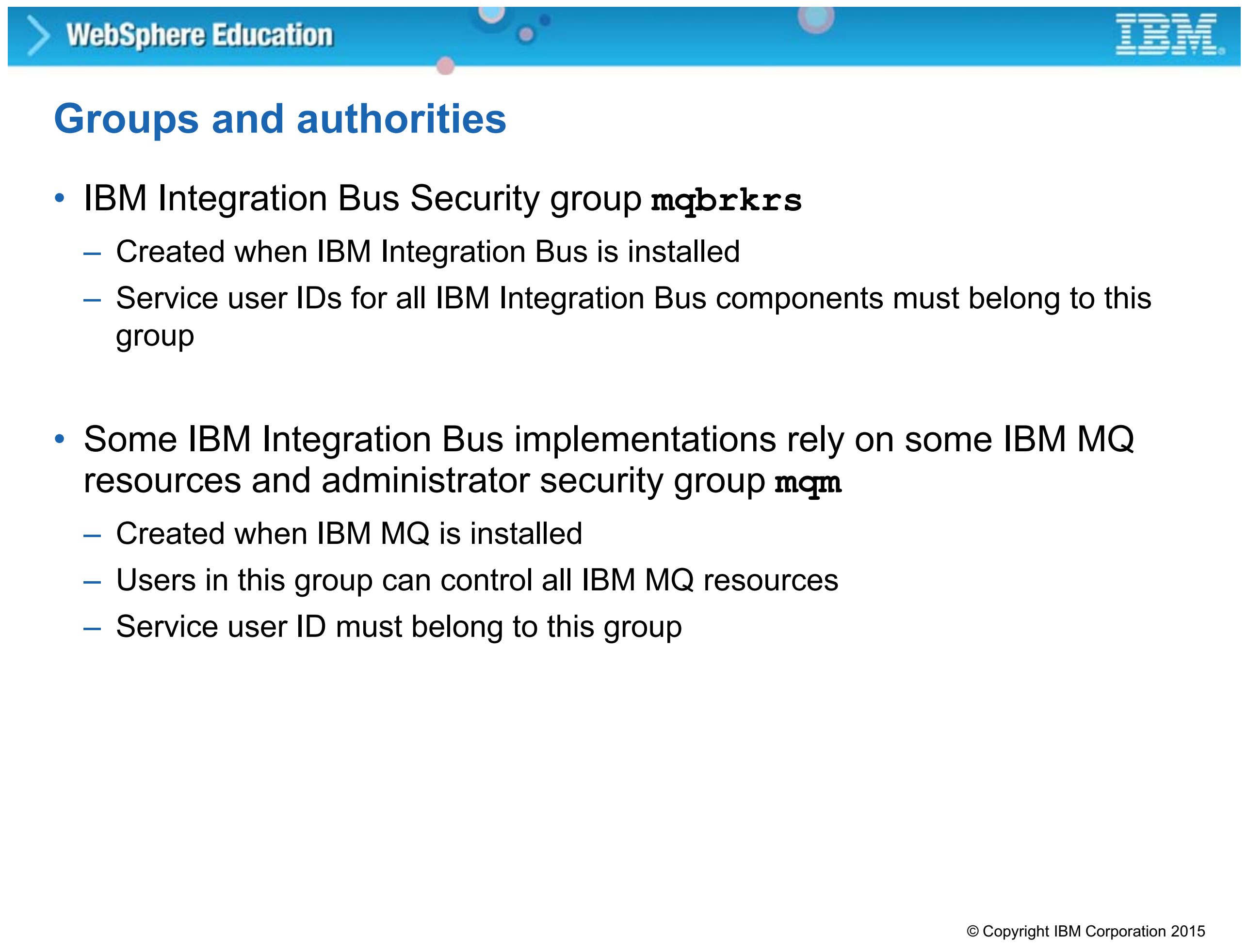
After completing this unit, you should be able to:

- Configure integration node administration security to control the authorities that developers and administrators use to complete their specific tasks
- Implement security for the IBM Integration web user interface

Figure 6-1. Unit objectives WM6461.0

Notes:

6.1. Integration node security overview



WebSphere Education

Groups and authorities

- IBM Integration Bus Security group **`mqbrkrs`**
  - Created when IBM Integration Bus is installed
  - Service user IDs for all IBM Integration Bus components must belong to this group
- Some IBM Integration Bus implementations rely on some IBM MQ resources and administrator security group **`mqm`**
  - Created when IBM MQ is installed
  - Users in this group can control all IBM MQ resources
  - Service user ID must belong to this group

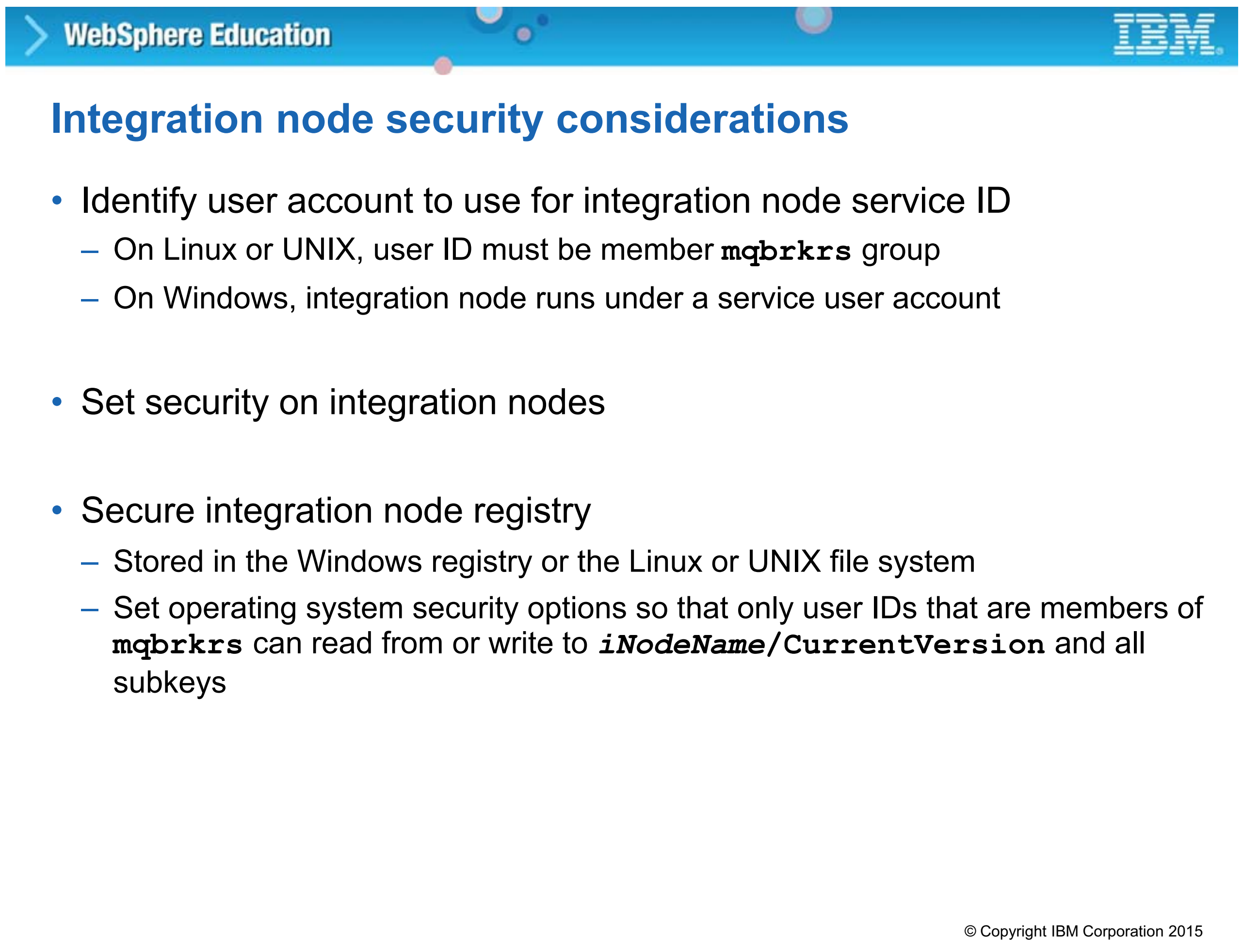
© Copyright IBM Corporation 2015

Figure 6-2. Groups and authorities WM6461.0

Notes:

The Integration Bus security group is `mqbrkrs`. This group is created when Integration Bus is installed. Members of the `mqbrkrs` group can modify an integration node, back up and restore an integration node, and delete an integration node. On Windows, the integration node service ID must belong to this group.

For implementations where an integration node is associated with a queue manager, IBM MQ is the security policy decision point for Integration Bus. The IBM MQ `mqm` administrator security group is created when IBM MQ is installed. It identifies users that can control all IBM MQ resources, including the integration node queue manager. Any new users that must be able to control IBM MQ resources must be added to this group. On Windows, the integration node service ID must also belong to this group.



WebSphere Education

Integration node security considerations

- Identify user account to use for integration node service ID
  - On Linux or UNIX, user ID must be member **`mqbrkrs`** group
  - On Windows, integration node runs under a service user account
- Set security on integration nodes
- Secure integration node registry
  - Stored in the Windows registry or the Linux or UNIX file system
  - Set operating system security options so that only user IDs that are members of **`mqbrkrs`** can read from or write to ***`iNodeName`*****`/CurrentVersion`** and all subkeys

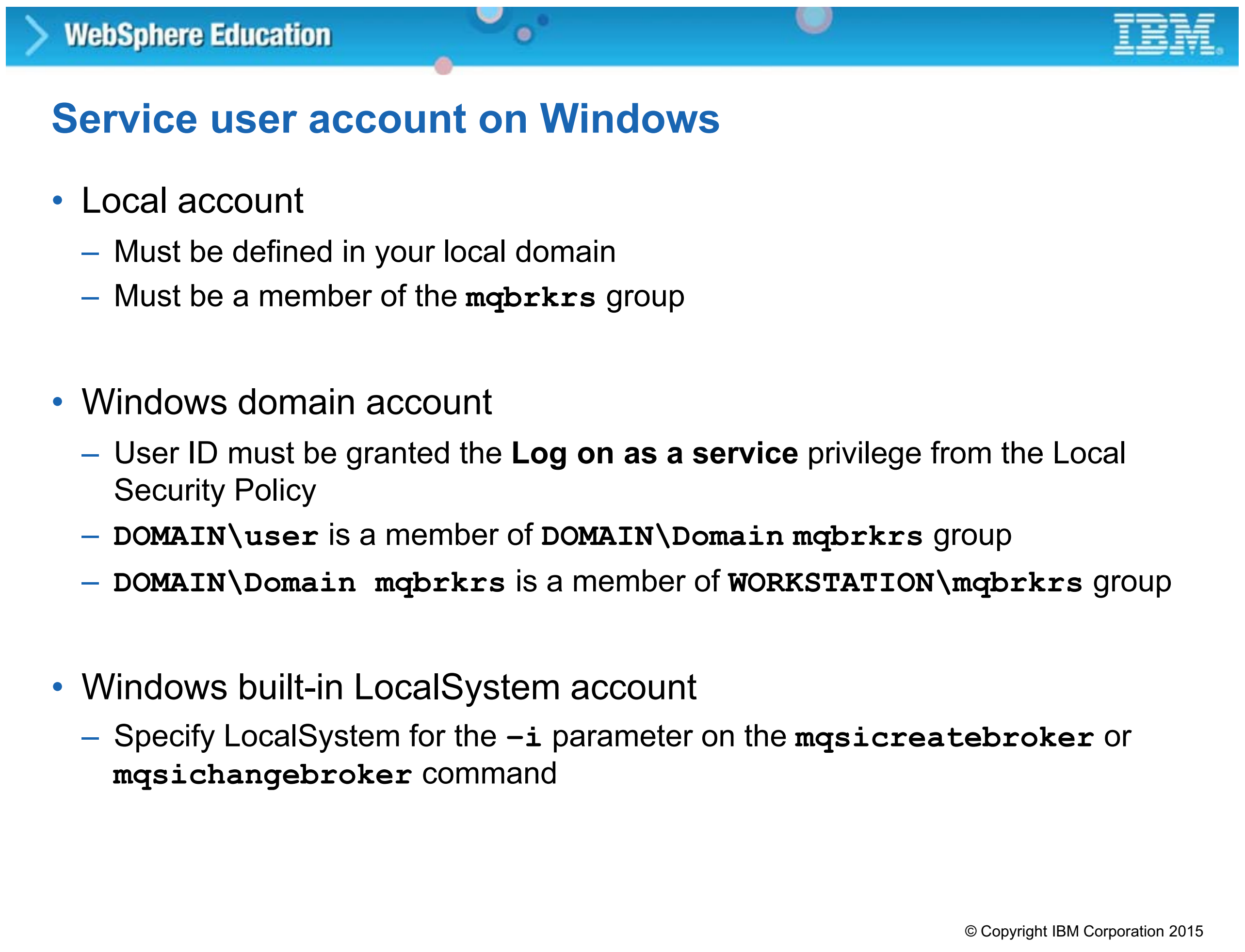
Figure 6-3. Integration node security considerations WM6461.0

Notes:

Consider several factors when you are deciding which users can run integration node commands, and which users can control security for other integration node resources. Although most security for the integration node and integration node resources is optional, you might find it appropriate to restrict the tasks that some user IDs can do. You can then apply greater control to monitor changes.

On Windows, the integration node runs under a service user account. On Linux or UNIX, the user ID where you run the `mqsistart` command is where the integration node component process runs.

Integration node operation depends on the information in the integration node registry, which must be secured to guard against accidental corruption. The integration node registry is stored in the Windows registry or in file system on Linux or UNIX.



WebSphere Education IBM

Service user account on Windows

- Local account
  - Must be defined in your local domain
  - Must be a member of the **`mqbrkrs`** group
- Windows domain account
  - User ID must be granted the **Log on as a service** privilege from the Local Security Policy
  - **`DOMAIN\user`** is a member of **`DOMAIN\Domain mqbrkrs`** group
  - **`DOMAIN\Domain mqbrkrs`** is a member of **`WORKSTATION\mqbrkrs`** group
- Windows built-in LocalSystem account
  - Specify LocalSystem for the **`-i`** parameter on the **`mqsicreatebroker`** or **`mqsichangebroker`** command

© Copyright IBM Corporation 2015

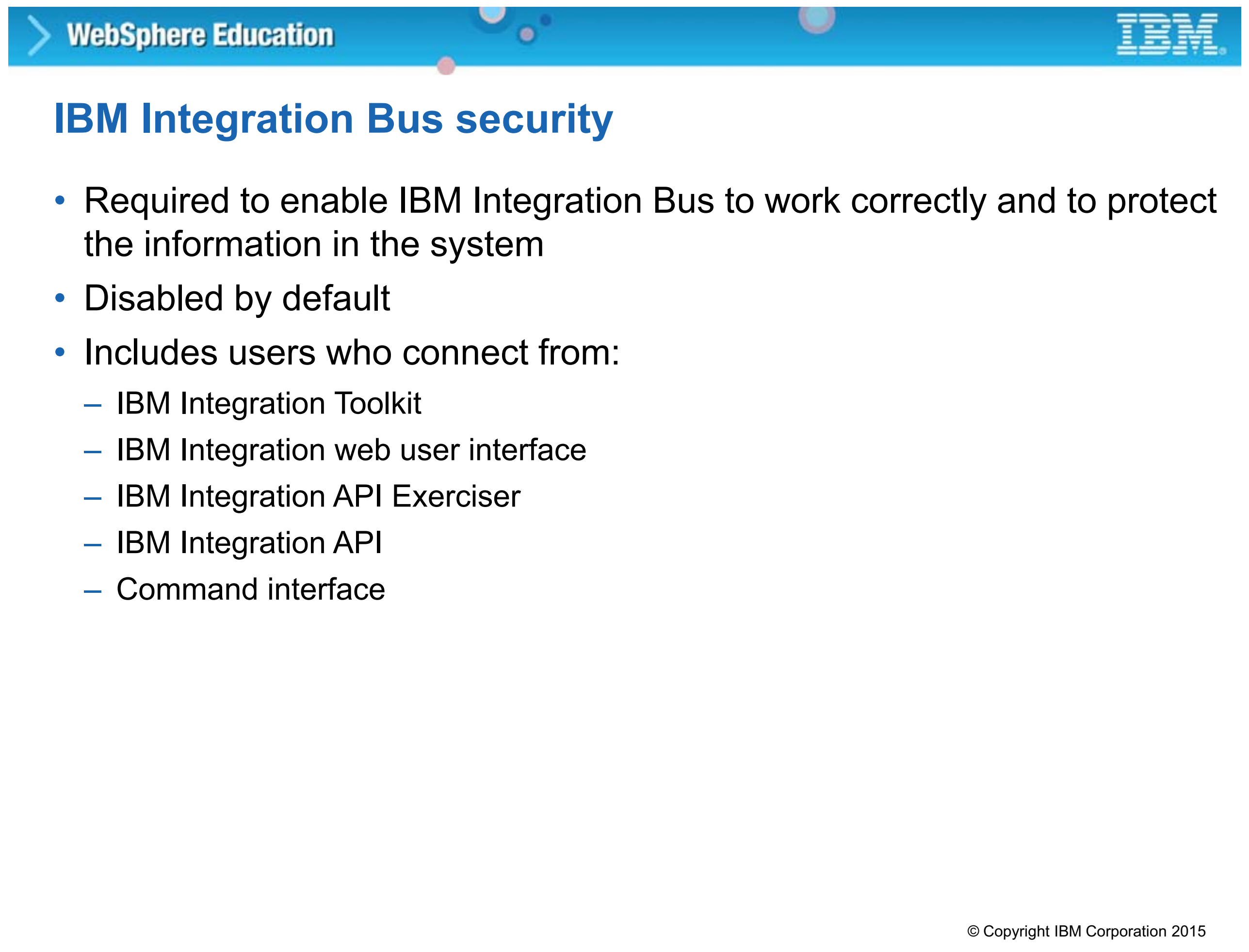
Figure 6-4. Service user account on Windows WM6461.0

Notes:

On Windows, the integration node runs under a service user account. The service user account can be a Windows local account, a Windows domain account, or the built-in LocalSystem account. The figure lists the requirements for each of the options.

For a local account or Windows domain account (cases 1 and 2), the user ID chosen must be granted the *Logon as a service* privilege. The `mqsichangebroker` or the `mqsichangeproperties` commands normally grant this privilege when a service user ID is specified that does not have it.

If you want to grant the privilege manually before running these commands, use the Local Security Policy tool in Windows (click **Control Panel > Administrative Tools > Local Security Policy**).



WebSphere Education

IBM Integration Bus security

- Required to enable IBM Integration Bus to work correctly and to protect the information in the system
- Disabled by default
- Includes users who connect from:
  - IBM Integration Toolkit
  - IBM Integration web user interface
  - IBM Integration API Exerciser
  - IBM Integration API
  - Command interface

© Copyright IBM Corporation 2015

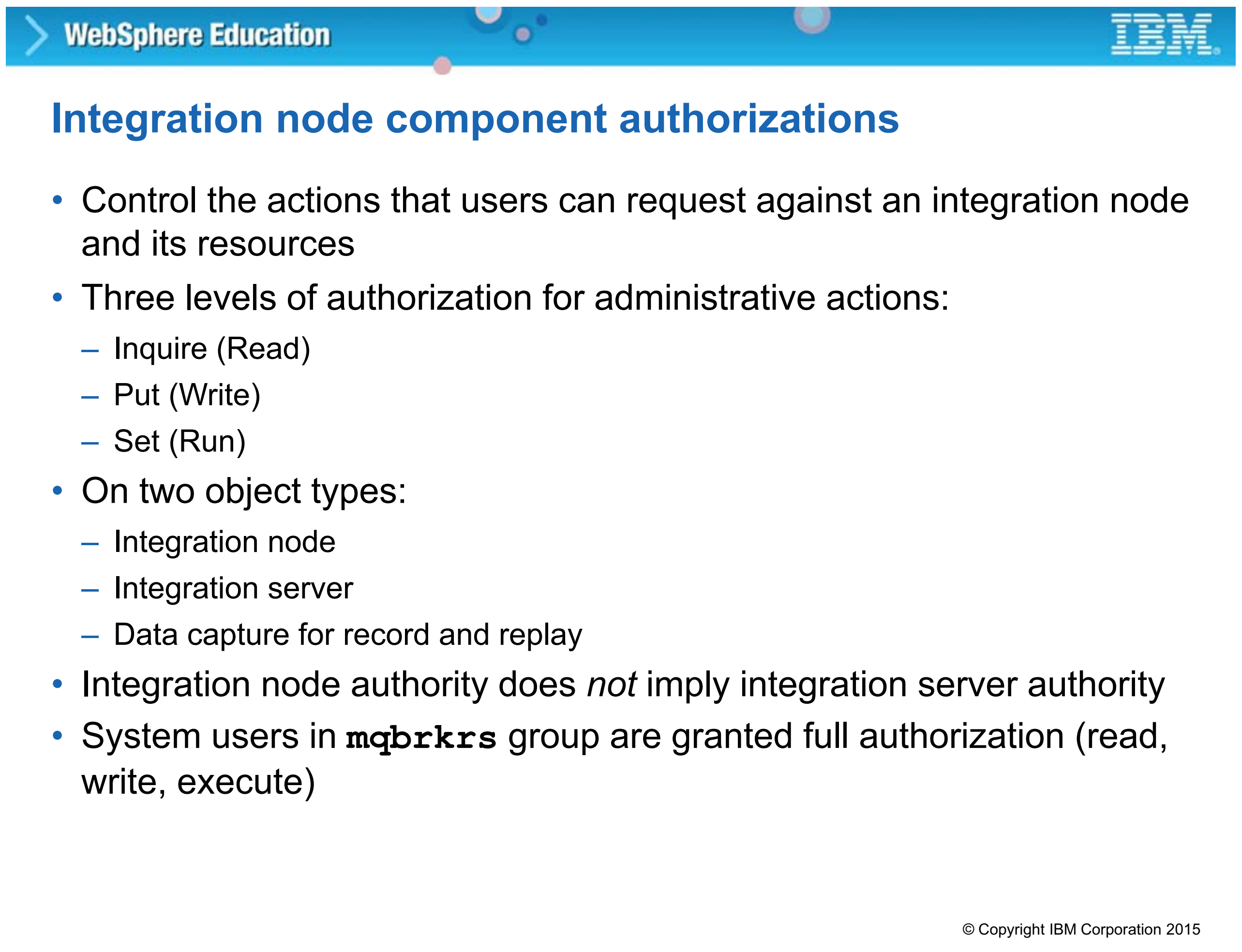
Figure 6-5. IBM Integration Bus security WM6461.0

Notes:

IBM Integration Bus security is based on permissions that are assigned to users and groups. Security is not enabled by default. It can be enabled by using commands. Security can also be specified when you create an integration node.

In IBM Integration Bus, the integration node is the policy enforcement point, and, in many implementations the policy decision point.

In implementations where an integration node is linked to a queue manager, IBM MQ is the policy decision point. This architecture creates a single place to define security and builds on existing skills by taking advantage of security support in IBM MQ.



WebSphere Education

Integration node component authorizations

- Control the actions that users can request against an integration node and its resources
- Three levels of authorization for administrative actions:
  - Inquire (Read)
  - Put (Write)
  - Set (Run)
- On two object types:
  - Integration node
  - Integration server
  - Data capture for record and replay
- Integration node authority does *not* imply integration server authority
- System users in **`mqbrkrs`** group are granted full authorization (read, write, execute)

Figure 6-6. Integration node component authorizations WM6461.0

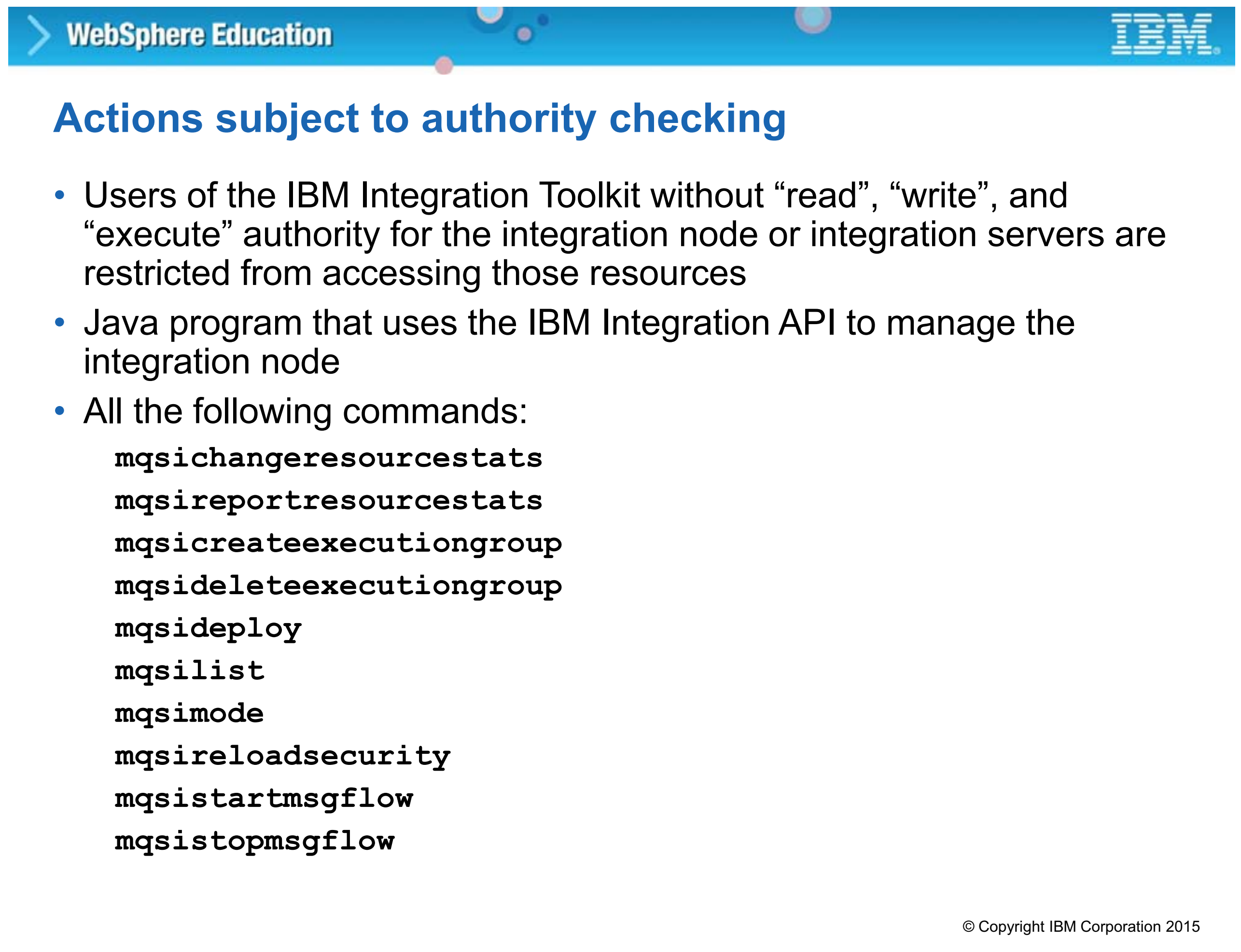
Notes:

Integration Bus component authorizations control the actions that users can request against the integration node and its resources. Integration Bus has three levels of authorization: inquire (read), put (write) and set (run).

Permissions are configured on only two types of objects: integration node and integration server.

Integration node and integration server authority are separate.

As mentioned previously, members of the `mqbrkrs` group can modify an integration node, back up and restore an integration node, and delete an integration node.



WebSphere Education

Actions subject to authority checking

- Users of the IBM Integration Toolkit without “read”, “write”, and “execute” authority for the integration node or integration servers are restricted from accessing those resources
- Java program that uses the IBM Integration API to manage the integration node
- All the following commands:

```
mqsichangeresourcestats
mqsireportresourcestats
mqsicreateexecutiongroup
mqsideleteexecutiongroup
mqsideploy
mqsilist
mqsimode
mqsireloadsecurity
mqsistartmsgflow
mqsistopmsgflow
```

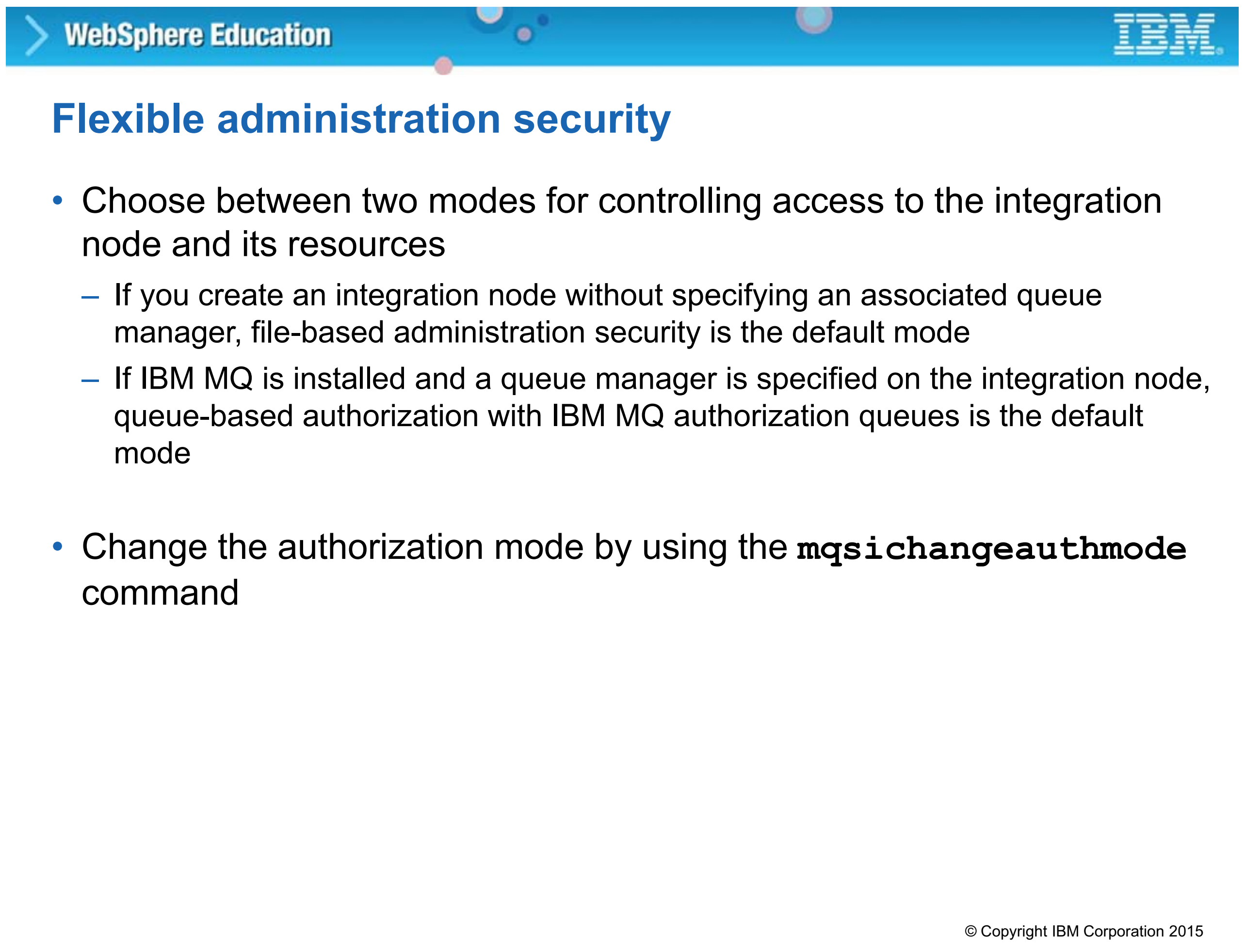
Figure 6-7. Actions subject to authority checking WM6461.0

Notes:

When integration node authorization security is activated, certain actions are restricted based on the access that is configured for each user or group.

For example, if developers are working with existing integration servers, and pre-production resources such as BAR files, you must give them inquire, put, and set permissions on the integration server authorization queues.

This figure lists the actions that subject to authority checking.



WebSphere Education

Flexible administration security

- Choose between two modes for controlling access to the integration node and its resources
  - If you create an integration node without specifying an associated queue manager, file-based administration security is the default mode
  - If IBM MQ is installed and a queue manager is specified on the integration node, queue-based authorization with IBM MQ authorization queues is the default mode
- Change the authorization mode by using the **`mqsichangeauthmode`** command

© Copyright IBM Corporation 2015

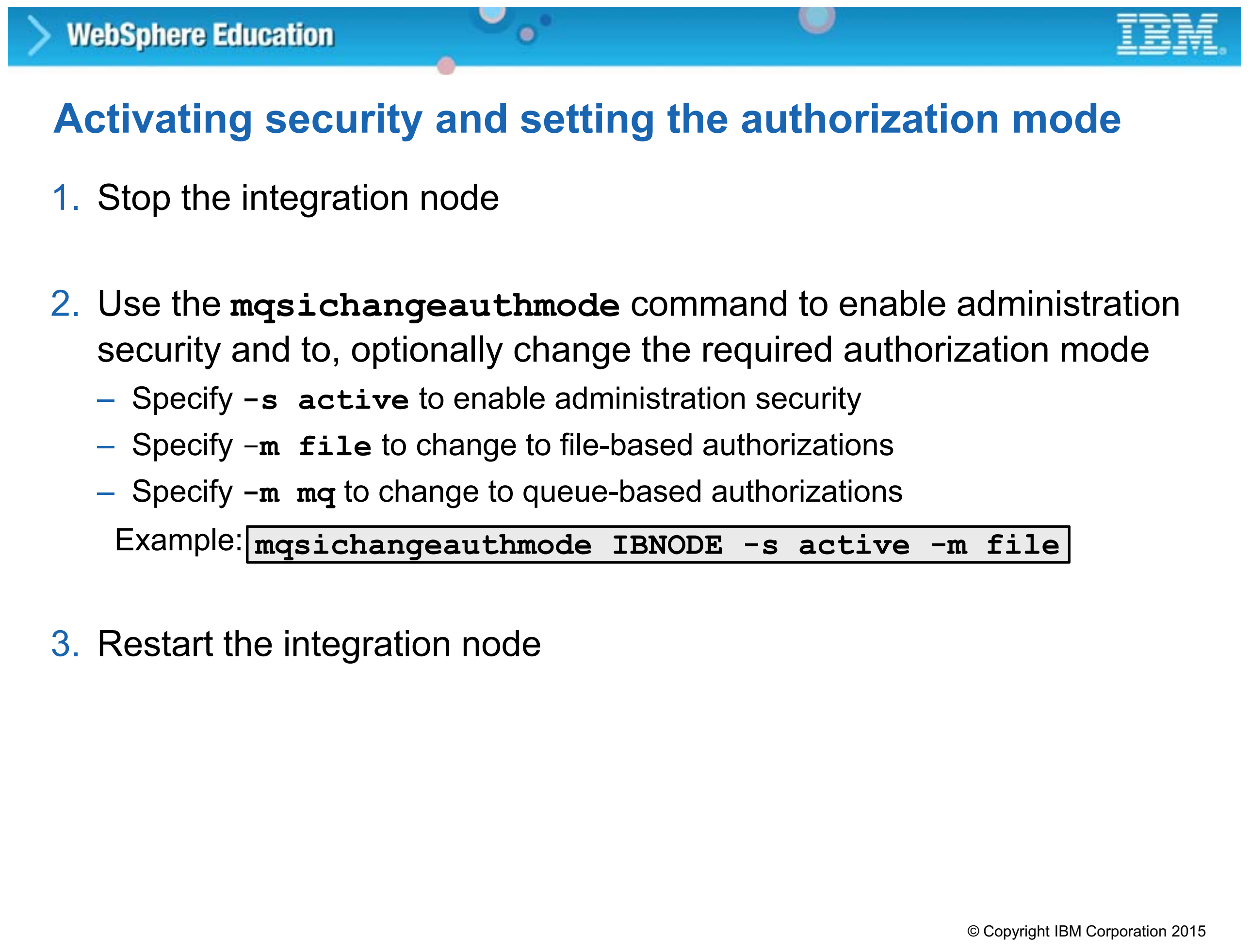
Figure 6-8. Flexible administration security WM6461.0

Notes:

In IBM Integration Bus V10, you can choose between two authorization modes for controlling access to the integration node and its resources.

- File-based administration security is configured by using the `mqsichangefileauth` command. This mode is the default mode if an integration node does not specify an associated IBM MQ queue manager.
- Queue-based administration uses IBM MQ authorization queues on the queue manager that is specified on the integration node. This mode s the default mode if an integration node specifies an associated IBM MQ queue. This mode is similar to administration security support in IBM Integration Bus V9 and WebSphere Message Broker.

You can change the administration authorization mode by using the `mqsichangeauthmode` command.



WebSphere Education IBM.

Activating security and setting the authorization mode

1. Stop the integration node
2. Use the **`mqsichangeauthmode`** command to enable administration security and to, optionally change the required authorization mode
   - Specify **`-s active`** to enable administration security
   - Specify **`–m file`** to change to file-based authorizations
   - Specify **`-m mq`** to change to queue-based authorizations

   Example:
   ```
   mqsichangeauthmode IBNODE -s active -m file
   ```
3. Restart the integration node

Figure 6-9. Activating security and setting the authorization mode WM6461.0

Notes:

This figure lists the steps for activating security and optionally change the authorization mode.

Important

You can use queue-based security only if you installed IBM MQ and a queue manager is specified on the integration node. If you specify queue-based security and the queue manager is later removed from the integration node while administration security is active (by using `-q` on the `mqsichangebroker` command), all access to the integration node is denied until a queue manager is specified on the integration node again, or until you change to file-based security and set the required permissions.

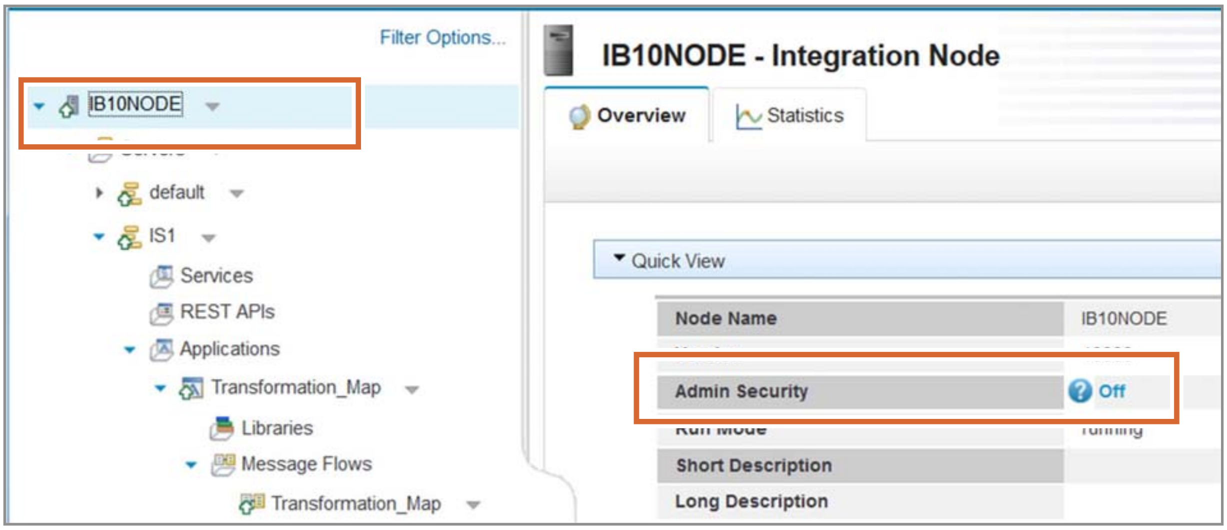
WebSphere Education IBM

Viewing authorization mode status

- Use the **mqsireportauthmode** command

  Example: `mqsireportauthmode IBNODE`

- Use the IBM Integration web user interface

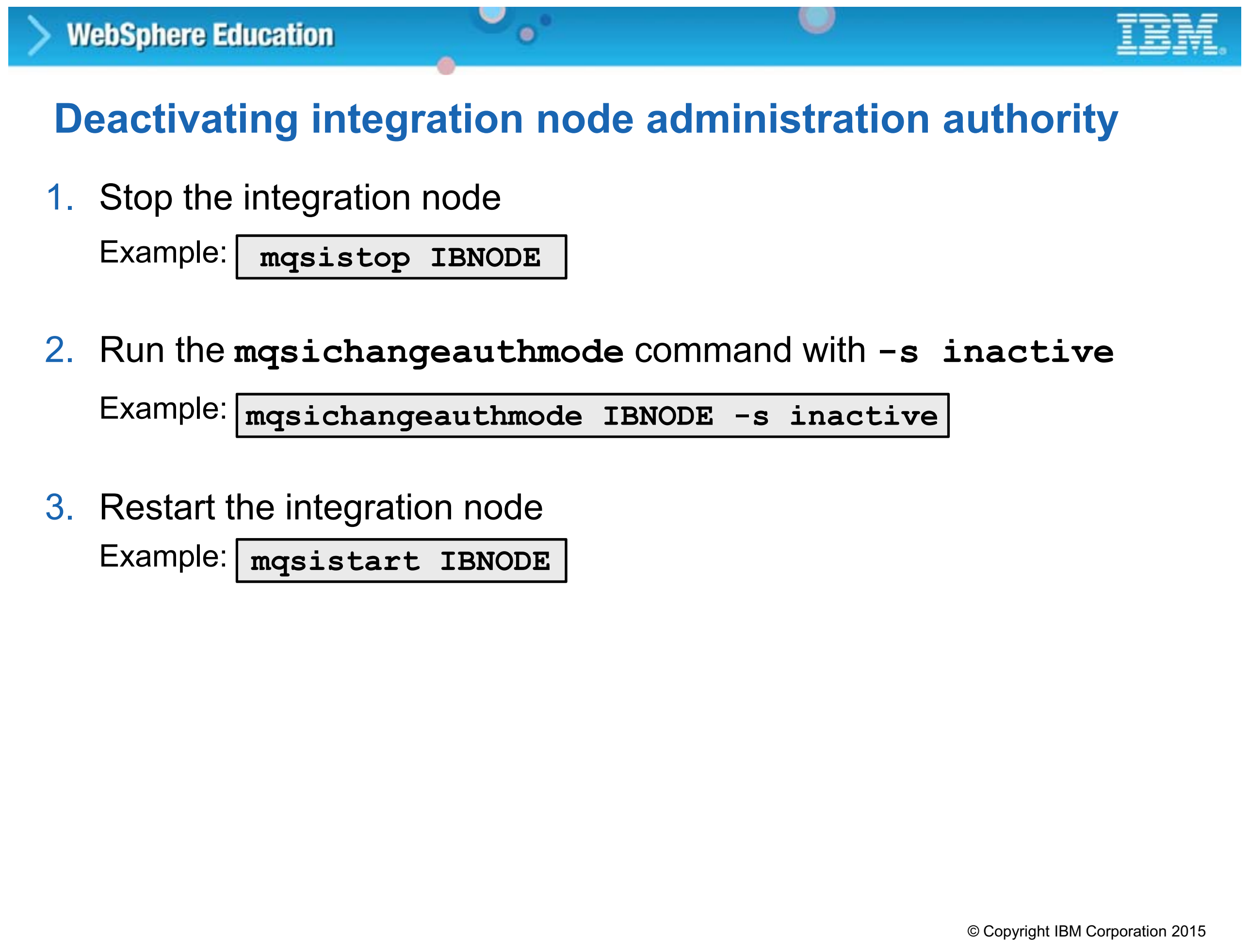


Node Name	IB10NODE
Admin Security	Off
Run Mode	running
Short Description	
Long Description	

Figure 6-10. Viewing authorization mode status WM6461.0

Notes:

You can use the `mqsireportauthmode` command and the IBM Integration web user interface to check the authorization mode status.



WebSphere Education

Deactivating integration node administration authority

1. Stop the integration node

   Example:
   ```
   mqsistop IBNODE
   ```

2. Run the **`mqsichangeauthmode`** command with **`-s inactive`**

   Example:
   ```
   mqsichangeauthmode IBNODE -s inactive
   ```

3. Restart the integration node

   Example:
   ```
   mqsistart IBNODE
   ```

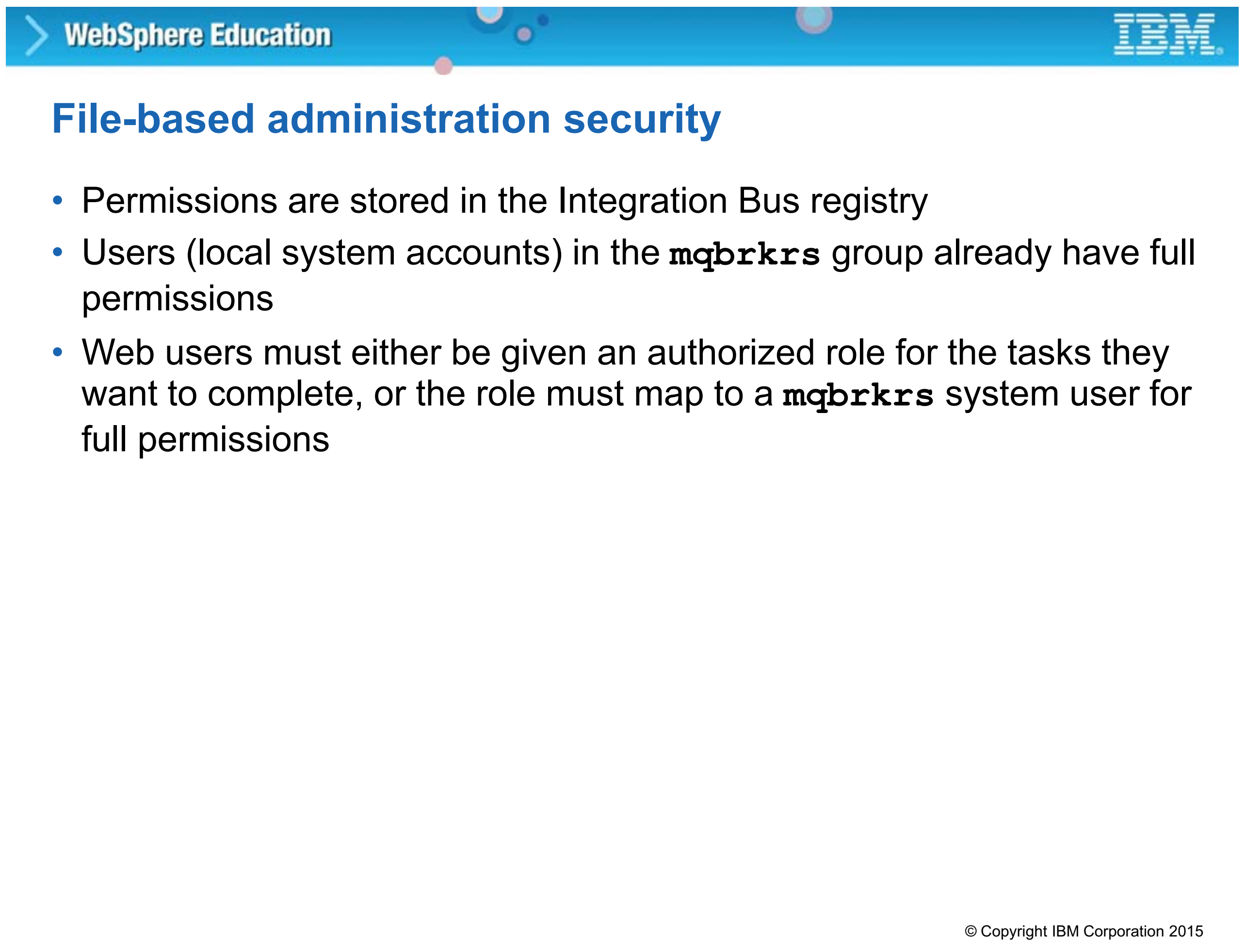
Figure 6-11. Deactivating integration node administration authority WM6461.0

Notes:

If necessary, you can deactivate administration security.

When you deactivate queue-based administration security, the authorization queues that are associated with this integration node are retained. If you activate security again, these queues are reused. Ensure that the authorizations that are defined by these queues are correct.

6.2. File-based administration security



WebSphere Education IBM

File-based administration security

- Permissions are stored in the Integration Bus registry
- Users (local system accounts) in the **`mqbrkrs`** group already have full permissions
- Web users must either be given an authorized role for the tasks they want to complete, or the role must map to a **`mqbrkrs`** system user for full permissions

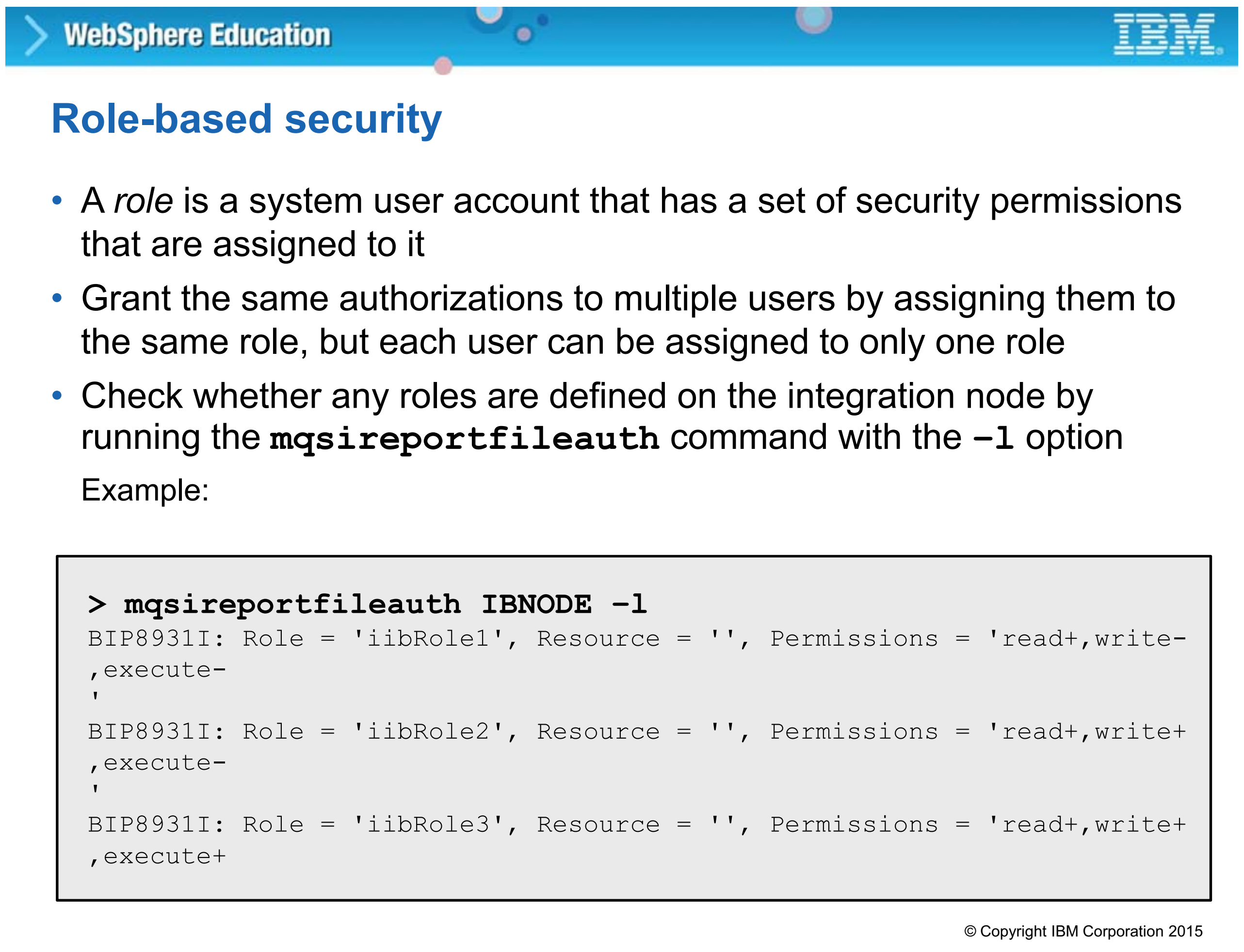
Figure 6-12. File-based administration security WM6461.0

Notes:

File-based permissions are stored in the integration node’s registry.

With file-based security, administrators can control the access to the integration node resources by assigning each user to a predefined role. You can authorize users with a particular role to complete specific actions. For example, you might allow users with one role to view integration node resources, while allowing users with another role to modify them.

If no permission is found for a role name, a check is conducted to see whether the name matches a system user ID. If that system user is a member of the `mqbrkrs` group, full permissions are given.



WebSphere Education

Role-based security

- A *role* is a system user account that has a set of security permissions that are assigned to it
- Grant the same authorizations to multiple users by assigning them to the same role, but each user can be assigned to only one role
- Check whether any roles are defined on the integration node by running the **mqsireportfileauth** command with the **-l** option

  Example:

```
> mqsireportfileauth IBNODE -l
BIP8931I: Role = 'iibRole1', Resource = '', Permissions = 'read+,write-
,execute-
'
BIP8931I: Role = 'iibRole2', Resource = '', Permissions = 'read+,write+
,execute-
'
BIP8931I: Role = 'iibRole3', Resource = '', Permissions = 'read+,write+
,execute+
```

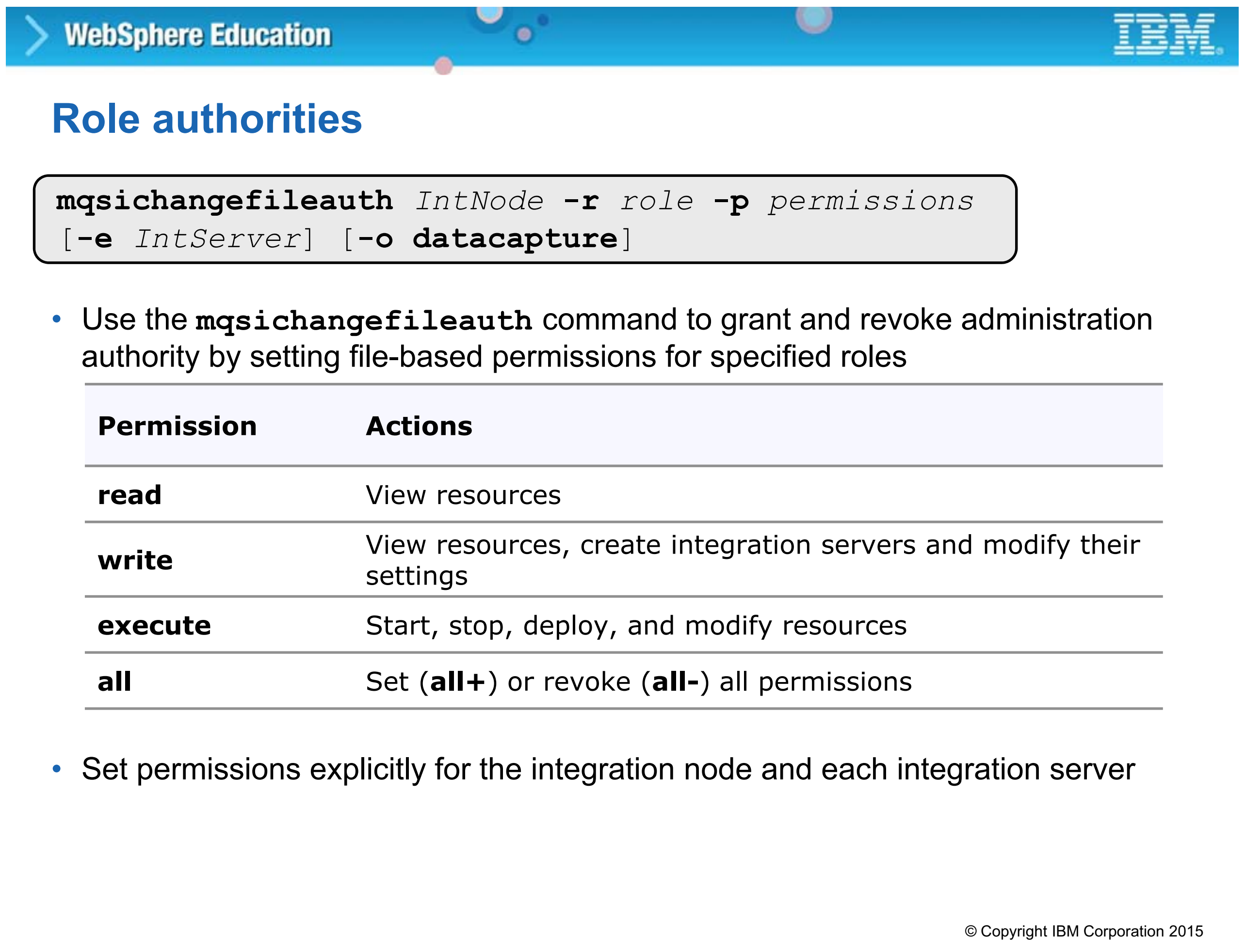
Figure 6-13. Role-based security WM6461.0

Notes:

You can control access to integration node and its resources by associating users with roles.

A *role* is a system user account that has a set of security permissions that are assigned to it, and each web user account is associated with a particular role. The permissions are checked to determine a web user's authorization to complete actions in the web user interface.

You can use the `mqsireportfileauth` command to check whether any roles are defined on the integration node.



WebSphere Education IBM

Role authorities

```
mqsichangefileauth IntNode -r role -p permissions
 [-e IntServer] [-o datacapture]
```

- Use the **mqsichangefileauth** command to grant and revoke administration authority by setting file-based permissions for specified roles

Permission	Actions
read	View resources
write	View resources, create integration servers and modify their settings
execute	Start, stop, deploy, and modify resources
all	Set (**all+**) or revoke (**all-**) all permissions

- Set permissions explicitly for the integration node and each integration server

Figure 6-14. Role authorities WM6461.0

Notes:

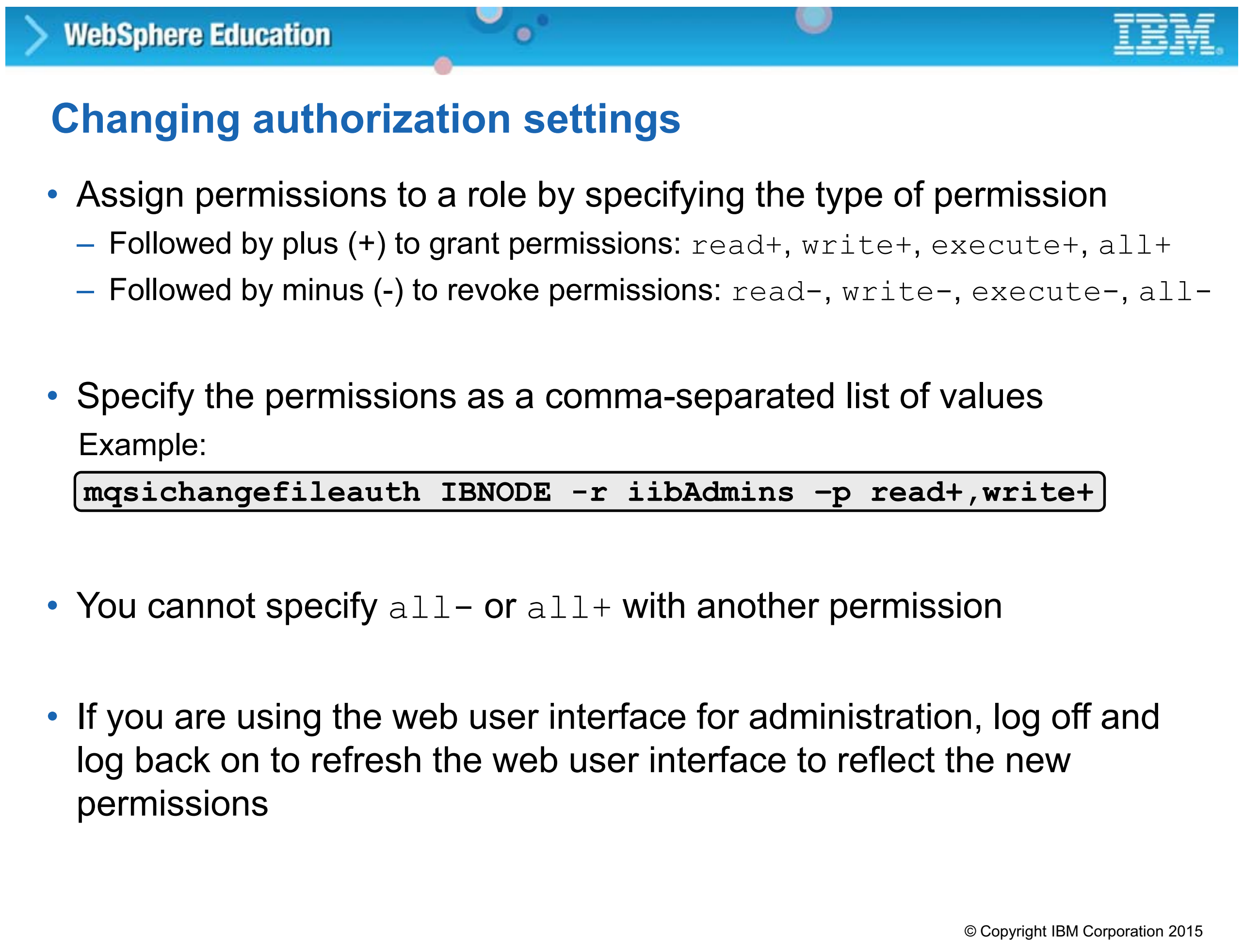
You use the `mqsichangefileauth` command to define roles and the administration authority of those roles.

Specify an integration server name with the `-e` parameter to add permissions to a specific integration server. If you specify an integration server, you cannot specify an object (resource) by using the `-o` parameter.

Specify the `-o datacapture` parameter to add permissions for record and replay data capture objects. To control users' ability to record, view, and replay data, you must enable integration node administration security and also configure security for data capture. If you specify this parameter, you cannot specify an integration server name by using the `-e` parameter.

Specify the permissions by using the `-p` parameter followed by the permissions. The following values are valid for this command:

- `read+` or `read-`
- `write+` or `write-`
- `execute+` or `execute-`
- `all+` or `all-`



WebSphere Education
IBM

Changing authorization settings

- Assign permissions to a role by specifying the type of permission
  - Followed by plus (+) to grant permissions: `read+`, `write+`, `execute+`, `all+`
  - Followed by minus (-) to revoke permissions: `read-`, `write-`, `execute-`, `all-`
- Specify the permissions as a comma-separated list of values

  Example:

```
mqsichangefileauth IBNODE -r iibAdmins -p read+,write+
```

- You cannot specify `all-` or `all+` with another permission
- If you are using the web user interface for administration, log off and log back on to refresh the web user interface to reflect the new permissions

Figure 6-15. Changing authorization settings WM6461.0

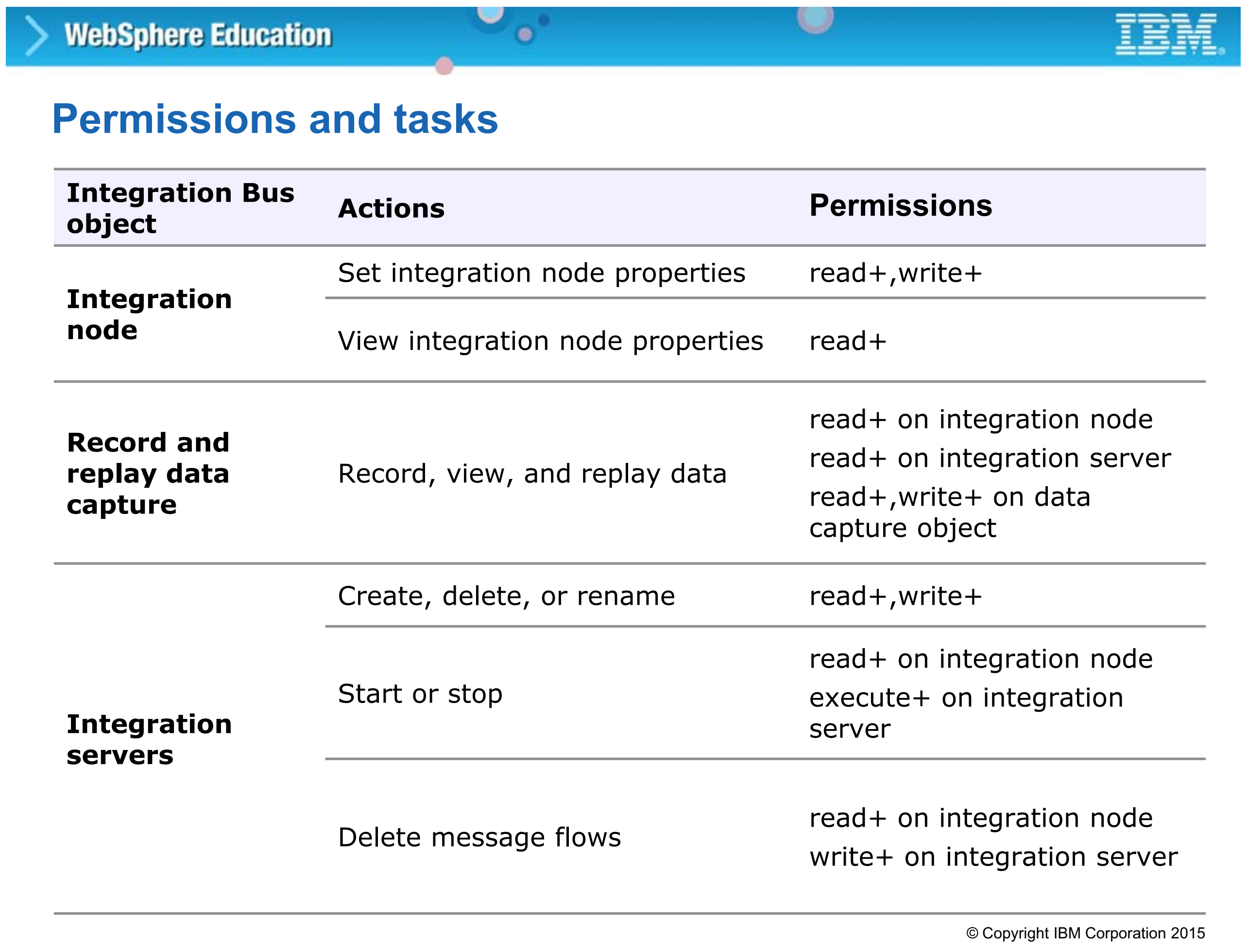
Notes:

The authorization permissions are specified as a comma-separated list of values on the `mqsichangefileauth` command.

A value can be specified for each permission only once in the list of values. For example, you cannot specify `all-,read+` because this command would set the read permission twice (one time explicitly, and again as part of `all`).

If `all` is specified, it must be the only value. If you specify `all-`, all permission records in the registry are removed.

If you are using the web user interface for administration, log off and log back on to refresh the web user interface to reflect the new permissions. If permissions are revoked, the change takes effect immediately and actions that require that permission fail because access is denied.



WebSphere Education

Permissions and tasks

Integration Bus object	Actions	Permissions
Integration node	Set integration node properties	read+,write+
	View integration node properties	read+
Record and replay data capture	Record, view, and replay data	read+ on integration node read+ on integration server read+,write+ on data capture object
Integration servers	Create, delete, or rename	read+,write+
	Start or stop	read+ on integration node execute+ on integration server
	Delete message flows	read+ on integration node write+ on integration server

© Copyright IBM Corporation 2015

Figure 6-16. Permissions and tasks WM6461.0

Notes:

This table summarizes the permissions to set to enable permissions for each administrative action and Integration Bus object.

If you want to control users' ability to record, view, and replay data, you must enable integration node administration security and also configure security for data capture.

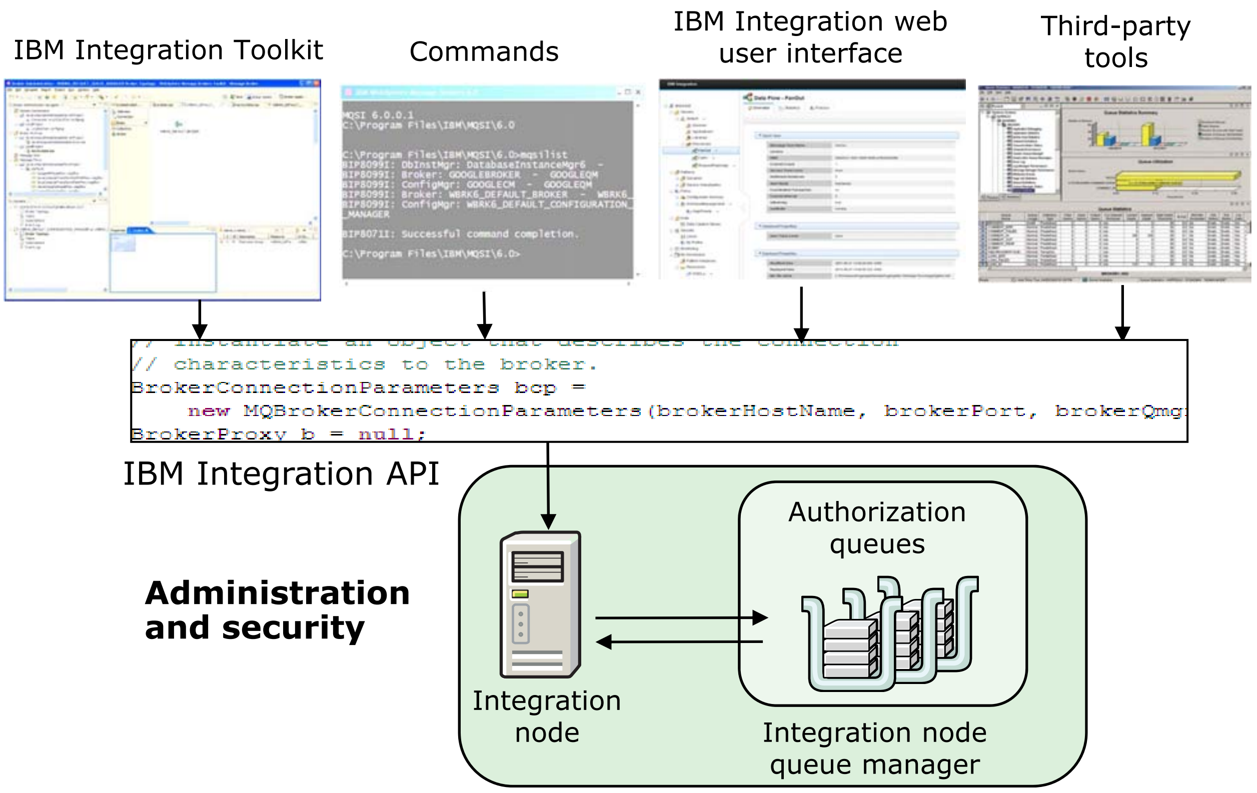
Important

If you grant permissions to a role at the integration node level, that permission is not applied to the node's integration servers; you must set permissions explicitly for individual integration servers.

6.3. Queue-based administration security

WebSphere Education IBM

Queue-based administration security



```
MQSI 6.0.0.1
C:\Program Files\IBM\MQSI\6.0


C:\Program Files\IBM\MQSI\6.0>mqsilist
BIP8099I: DbInstMgr: DatabaseInstanceMgr6  -
BIP8099I: Broker: GOOGLEBROKER  -  GOOGLEQM
BIP8099I: ConfigMgr: GOOGLECM  -  GOOGLEQM
BIP8099I: Broker: WBRK6_DEFAULT_BROKER  -  WBRK6_
BIP8099I: ConfigMgr: WBRK6_DEFAULT_CONFIGURATION_
_MANAGER

BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\6.0>
```

```
// instantiate an object that describes the connection
// characteristics to the broker.
BrokerConnectionParameters bcp =
    new MQBrokerConnectionParameters(brokerHostName, brokerPort, brokerQmg
BrokerProxy b = null;
```

Figure 6-17. Queue-based administration security WM6461.0

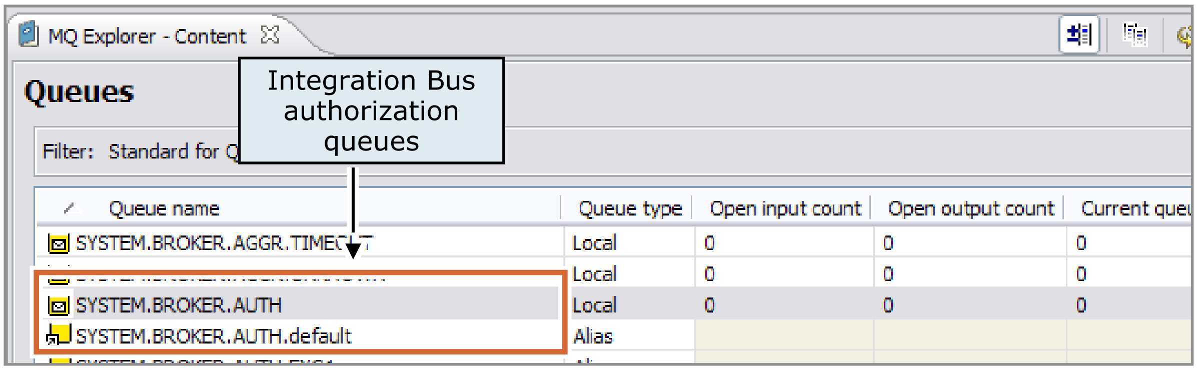
Notes:

This figure shows another way to view the IBM Integration Bus architecture. The top layer shows all the communication links to the integration node through the IBM Integration API. These links include the IBM Integration Toolkit and IBM Integration web user interface, command interface, and third-party tools that might be using the Administration API to communicate with the integration node.

When queue-based authorization is enabled, the integration node communicates with authorization queues that are used to grant or deny users authority to act on integration node objects.

WebSphere Education

Authorization queues



Queue name	Queue type	Open input count	Open output count	Current quei
SYSTEM.BROKER.AGGR.TIMEOUT	Local	0	0	0
	Local	0	0	0
SYSTEM.BROKER.AUTH	Local	0	0	0
SYSTEM.BROKER.AUTH.default	Alias			

- Integration node creates authorization queues when queue-based authority is enabled
  - One local queue for integration node authorization
  - One alias queue for each integration server, for integration server authorization

Figure 6-18. Authorization queues WM6461.0

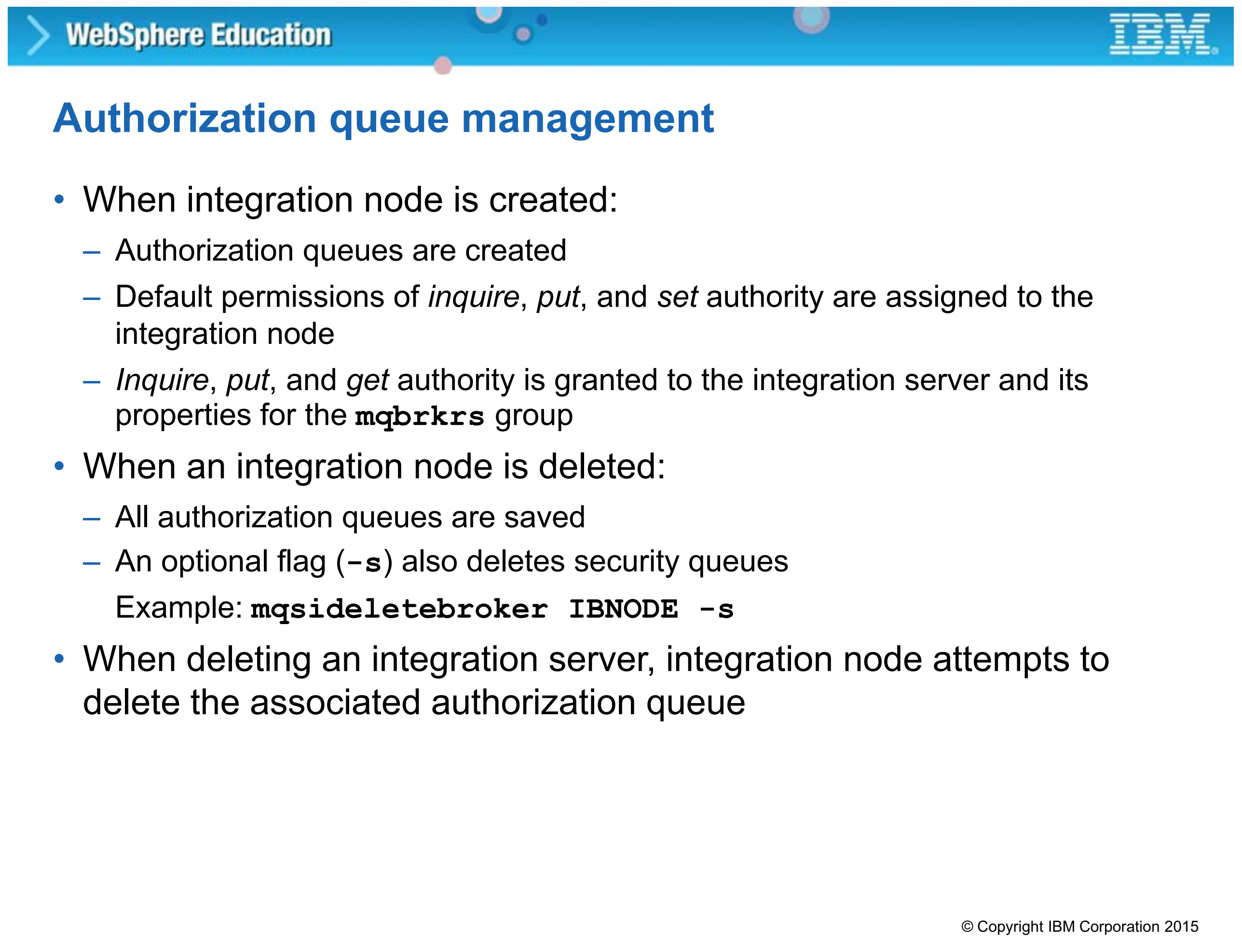
Notes:

If you are using queue-based security, the integration node queue manager contains one authorization queue for the integration node (SYSTEM.BROKER.AUTH) and one authorization queue for each integration server. An integration server queue that is named SYSTEM.BROKER.AUTH.*integration_server_name* is created when you create an integration server on an integration node for which queue-based administrative security is enabled.

The Integration Bus integration node group (`mqbrkrs`) is given authorization to the authorization queues by default when queue-based security is enabled. Users are prevented from completing any actions on integration nodes or integration servers until they are given authorization by the administrator.

No messages are written to the authorization queues; they are only used to identify integration node and integration server permissions. The depth on any authorization queues should always be zero.

In IBM MQ Explorer, you can click the **Show system queues** icon to show the authorization queues.



WebSphere Education

Authorization queue management

- When integration node is created:
  - Authorization queues are created
  - Default permissions of *inquire*, *put*, and *set* authority are assigned to the integration node
  - *Inquire*, *put*, and *get* authority is granted to the integration server and its properties for the **`mqbrkrs`** group
- When an integration node is deleted:
  - All authorization queues are saved
  - An optional flag (**`-s`**) also deletes security queues

    Example: **`mqsideletebroker IBNODE -s`**
- When deleting an integration server, integration node attempts to delete the associated authorization queue

© Copyright IBM Corporation 2015

Figure 6-19. Authorization queue management WM6461.0

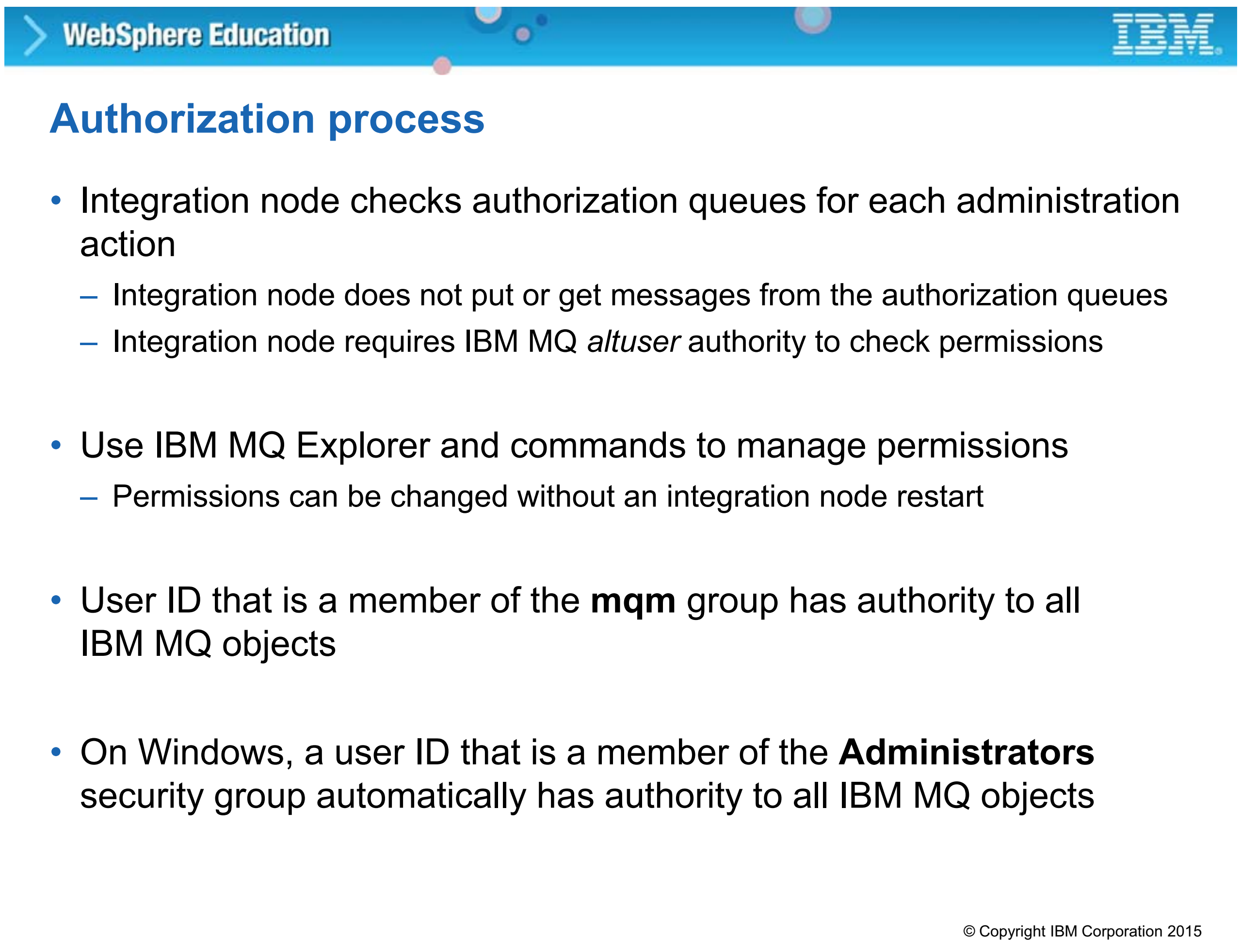
Notes:

The integration node authorization queue is created when you create an integration node that is associated with a queue manager or when you enable queue-based security. If, for some reason, the integration node cannot create the queues dynamically, a message is written to the system log. In this case, the integration node administrator must then either manually create the queue, or run the `mqsichangebroker` to create the queue.

An integration server authorization queue is created when an integration server is created and queue-based authorization is enabled. When an integration server is renamed, an authorization queue is created. The old queue is then deleted.

The integration node authorization queue is not deleted when the integration node is deleted unless you specifically indicate that the queue should be deleted.

If integration servers are renamed, the existing authorization is deleted, and an authorization queue is created, if it does not exist. The permissions that were set on the original queue are lost.



WebSphere Education

Authorization process

- Integration node checks authorization queues for each administration action
  - Integration node does not put or get messages from the authorization queues
  - Integration node requires IBM MQ *altuser* authority to check permissions
- Use IBM MQ Explorer and commands to manage permissions
  - Permissions can be changed without an integration node restart
- User ID that is a member of the **mqm** group has authority to all IBM MQ objects
- On Windows, a user ID that is a member of the **Administrators** security group automatically has authority to all IBM MQ objects

© Copyright IBM Corporation 2015

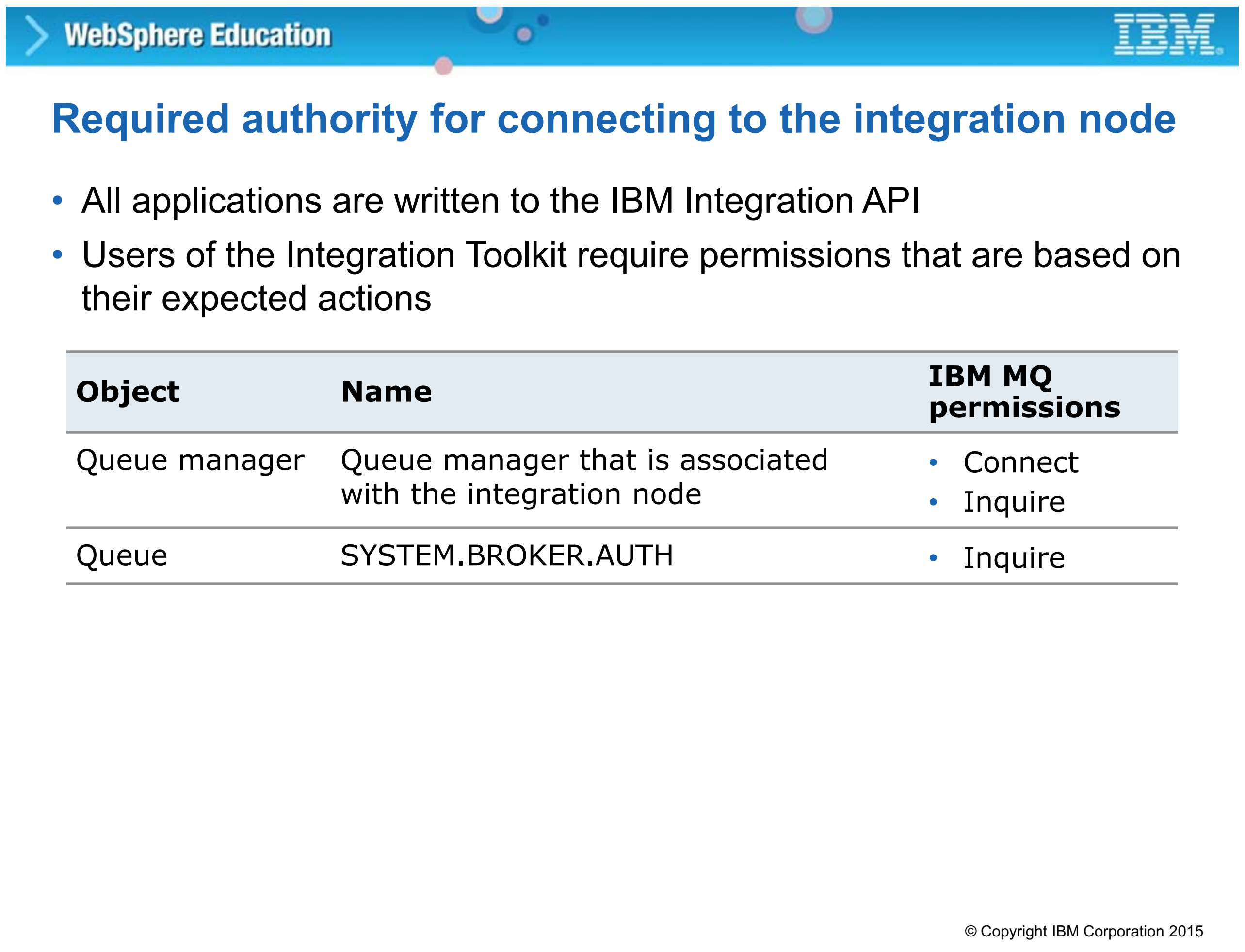
Figure 6-20. Authorization process WM6461.0

Notes:

If queue-based authorization is enabled, the integration node checks the IBM MQ authorization queues for every action. If permission is granted, the action is done.

The administrator can change permissions at any time without restarting the integration node. The integration node automatically detects the change the next time the user tries the action.

Any user that has IBM MQ alternative user command authority can check the integration node authorization permissions.



Required authority for connecting to the integration node

- All applications are written to the IBM Integration API
- Users of the Integration Toolkit require permissions that are based on their expected actions

Object	Name	IBM MQ permissions
Queue manager	Queue manager that is associated with the integration node	• Connect • Inquire
Queue	SYSTEM.BROKER.AUTH	• Inquire

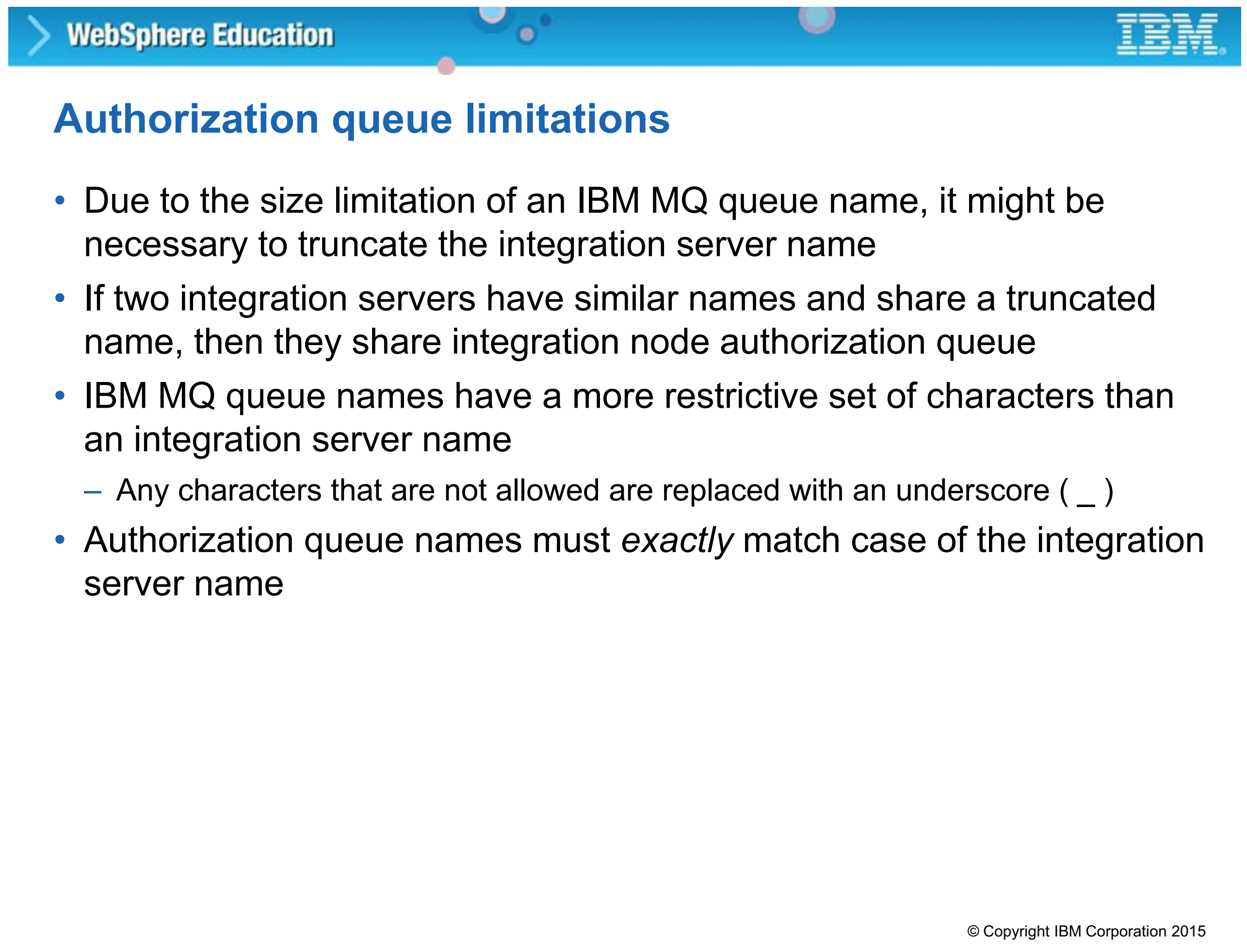
Figure 6-21. Required authority for connecting to the integration node WM6461.0

Notes:

If a user or application wants to connect to an integration node, you must grant them the appropriate permissions. All applications that are written to the IBM Integration API, and users of the IBM Integration Toolkit, require permissions that are based on their expected actions.

The table in the figure shows the IBM MQ permissions that are required to connect to the integration node when queue-based authorization is enabled.

The SYSTEM.BROKER.AUTH queue stores authorization records for administration requests against the integration node. Users and applications can connect to the integration node without “inquire” permissions on the SYSTEM.BROKER.AUTH queue, but they cannot request actions against the integration node, including viewing properties.



WebSphere Education

Authorization queue limitations

- Due to the size limitation of an IBM MQ queue name, it might be necessary to truncate the integration server name
- If two integration servers have similar names and share a truncated name, then they share integration node authorization queue
- IBM MQ queue names have a more restrictive set of characters than an integration server name
  - Any characters that are not allowed are replaced with an underscore (_)
- Authorization queue names must *exactly* match case of the integration server name

© Copyright IBM Corporation 2015

Figure 6-22. Authorization queue limitations WM6461.0

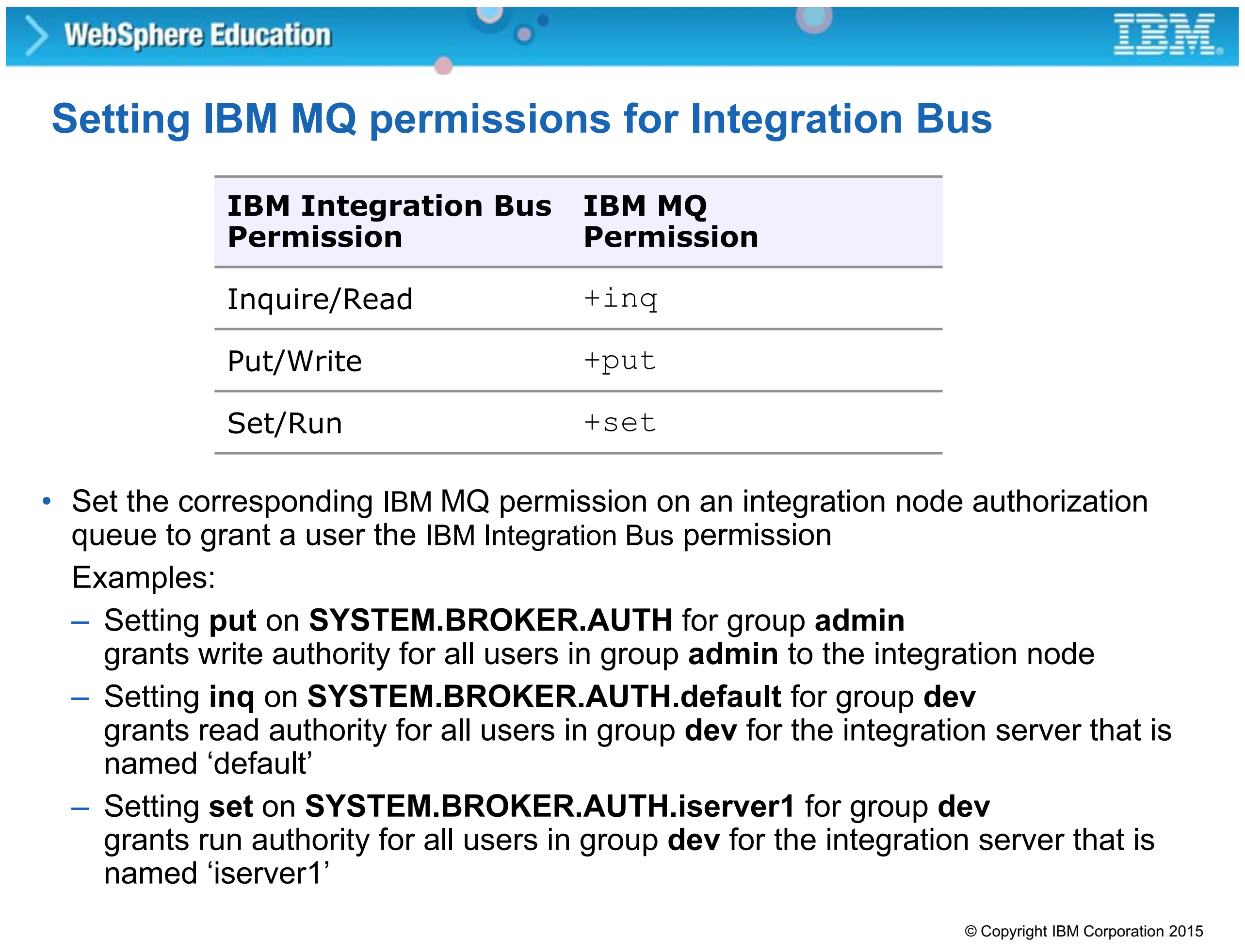
Notes:

Integration server names are used to create authorization queue names when queue-based authorization is enabled. For this reason, you should be aware of the limitations on authorization queue names so that you can name your integration servers correctly.

If you create an integration server with a name longer than 30 characters, it is truncated to 30 characters and the truncated name is used for the authorization queue name. If two integration servers have the same truncated name, they share an authorization queue. For example, if integration server names of two integration servers are 40 characters, but the first 30 characters are the same, they would share a queue.

Important

The queue names must exactly match the case of the integration server name. When IBM Integration Bus creates the queues, naming is handled automatically. Mismatched names might be a problem if the administrator creates the queues in advance or if the queues are renamed.



WebSphere Education

Setting IBM MQ permissions for Integration Bus

IBM Integration Bus Permission	IBM MQ Permission
Inquire/Read	`+inq`
Put/Write	`+put`
Set/Run	`+set`

- Set the corresponding IBM MQ permission on an integration node authorization queue to grant a user the IBM Integration Bus permission

  Examples:
  - Setting **put** on **SYSTEM.BROKER.AUTH** for group **admin** grants write authority for all users in group **admin** to the integration node
  - Setting **inq** on **SYSTEM.BROKER.AUTH.default** for group **dev** grants read authority for all users in group **dev** for the integration server that is named ‘default’
  - Setting **set** on **SYSTEM.BROKER.AUTH.iserver1** for group **dev** grants run authority for all users in group **dev** for the integration server that is named ‘iserver1’

© Copyright IBM Corporation 2015

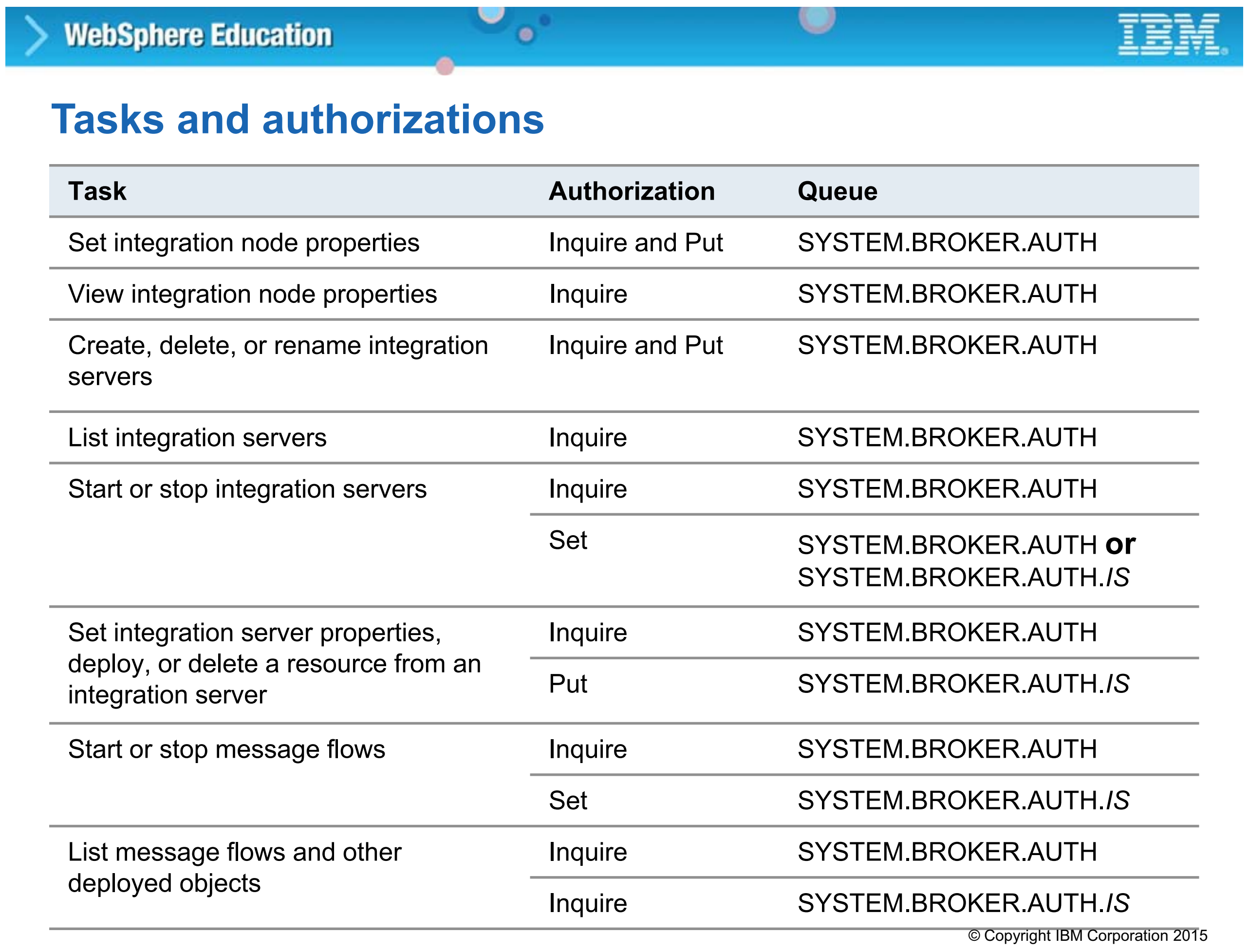
Figure 6-23. Setting IBM MQ permissions for Integration Bus WM6461.0

Notes:

This table shows the mapping between Integration Bus permissions and IBM MQ permissions:

- Read equals `+inq`
- Write equals `+put`
- Run equals `+set`

For example, to grant “write” authority to the integration node for all users in the `admin` group, enable “put” on the SYSTEM.BROKER.AUTH queue.



WebSphere Education

Tasks and authorizations

Task	Authorization	Queue
Set integration node properties	Inquire and Put	SYSTEM.BROKER.AUTH
View integration node properties	Inquire	SYSTEM.BROKER.AUTH
Create, delete, or rename integration servers	Inquire and Put	SYSTEM.BROKER.AUTH
List integration servers	Inquire	SYSTEM.BROKER.AUTH
Start or stop integration servers	Inquire	SYSTEM.BROKER.AUTH
	Set	SYSTEM.BROKER.AUTH **or** SYSTEM.BROKER.AUTH.*IS*
Set integration server properties, deploy, or delete a resource from an integration server	Inquire	SYSTEM.BROKER.AUTH
	Put	SYSTEM.BROKER.AUTH.*IS*
Start or stop message flows	Inquire	SYSTEM.BROKER.AUTH
	Set	SYSTEM.BROKER.AUTH.*IS*
List message flows and other deployed objects	Inquire	SYSTEM.BROKER.AUTH
	Inquire	SYSTEM.BROKER.AUTH.*IS*

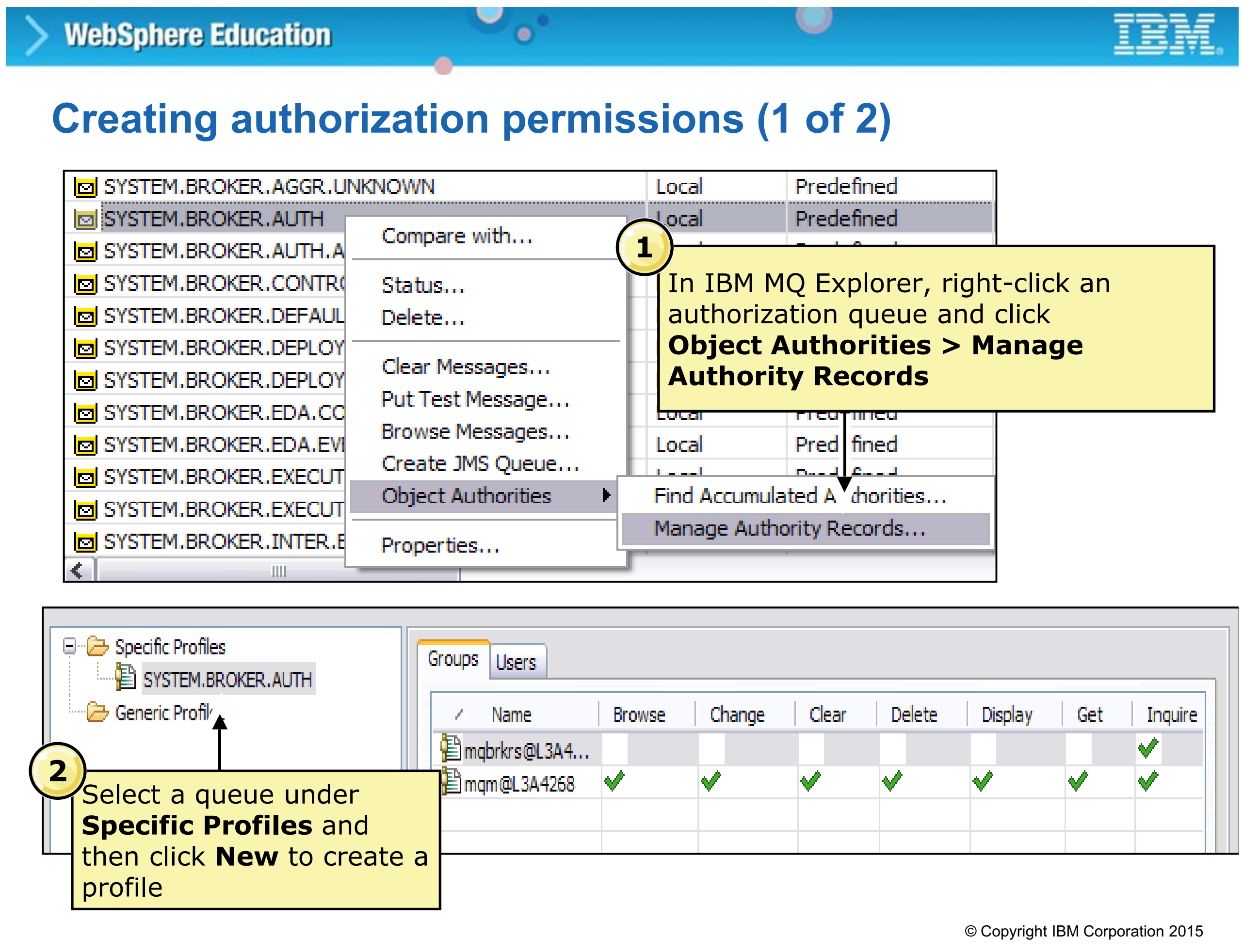
Figure 6-24. Tasks and authorizations WM6461.0

Notes:

The table lists some common tasks and the related authorization that must be set on the authorization queue.

“IS” in the table represents integration server. For example, if a developer must be able to start and stop message flows on an integration server that is named IS1, “set” permission must be enabled on the SYSTEM.BROKER.AUTH.IS1 queue.

For a complete list of tasks and permissions, see the IBM Knowledge Center.



Creating authorization permissions (1 of 2)

1 In IBM MQ Explorer, right-click an authorization queue and click **Object Authorities > Manage Authority Records**

Name	Browse	Change	Clear	Delete	Display	Get	Inquire
mqbrkrs@L3A4...							✓
mqm@L3A4268	✓	✓	✓	✓	✓	✓	✓

2 Select a queue under **Specific Profiles** and then click **New** to create a profile

© Copyright IBM Corporation 2015

Figure 6-25. Creating authorization permissions (1 of 2) WM6461.0

Notes:

IBM MQ Explorer can be used to set authorization permissions, as shown in the figure.

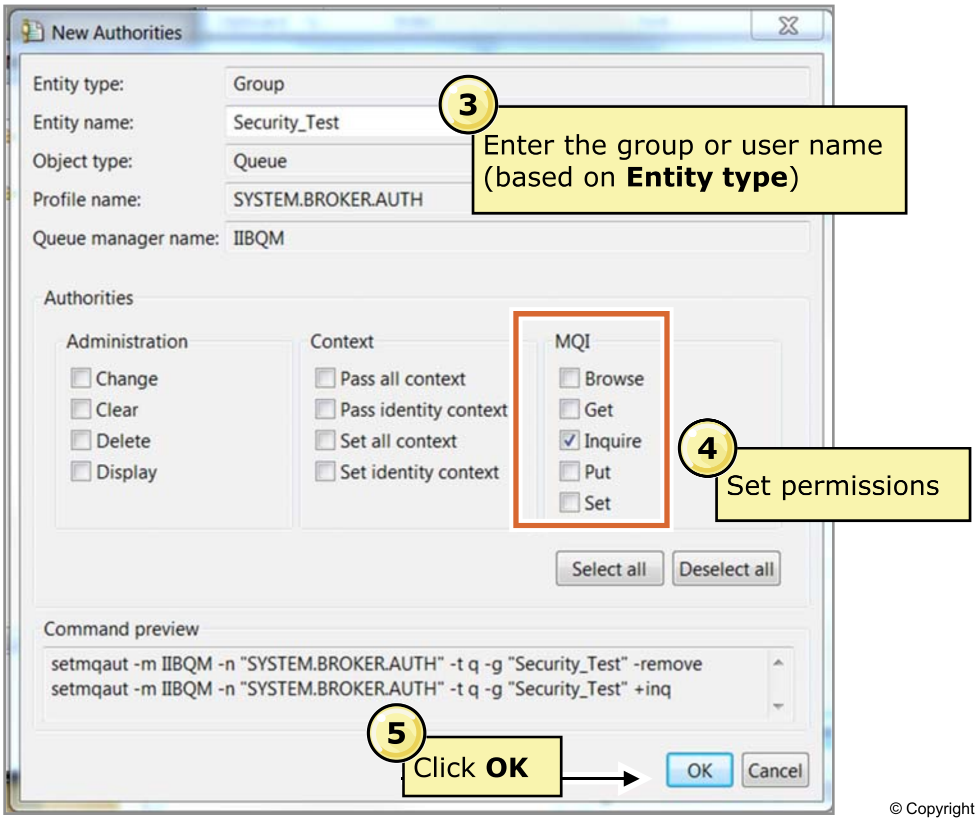
To set authorization on a specific queue:

1. Right-click the SYSTEM.BROKER.AUTH or SYSTEM.BROKER.AUTH.*IS* queue in the **Queues** content view and then click **Object Authorities > Manage Authority Records**.
2. The security profile for the selected queue is displayed. The screen capture at the bottom of the figure shows the permissions page for SYSTEM.BROKER.AUTH.

   From this page, you can either add users or groups to the authorization profile or modify existing permissions.

WebSphere Education

Creating authorization permissions (2 of 2)



Entity type: Group
Entity name: Security_Test
Object type: Queue
Profile name: SYSTEM.BROKER.AUTH
Queue manager name: IIBQM

Authorities

Administration
- [ ] Change
- [ ] Clear
- [ ] Delete
- [ ] Display

Context
- [ ] Pass all context
- [ ] Pass identity context
- [ ] Set all context
- [ ] Set identity context

MQI
- [ ] Browse
- [ ] Get
- [x] Inquire
- [ ] Put
- [ ] Set

Command preview

```
setmqaut -m IIBQM -n "SYSTEM.BROKER.AUTH" -t q -g "Security_Test" -remove
setmqaut -m IIBQM -n "SYSTEM.BROKER.AUTH" -t q -g "Security_Test" +inq
```

Figure 6-26. Creating authorization permissions (2 of 2) WM6461.0

Notes:

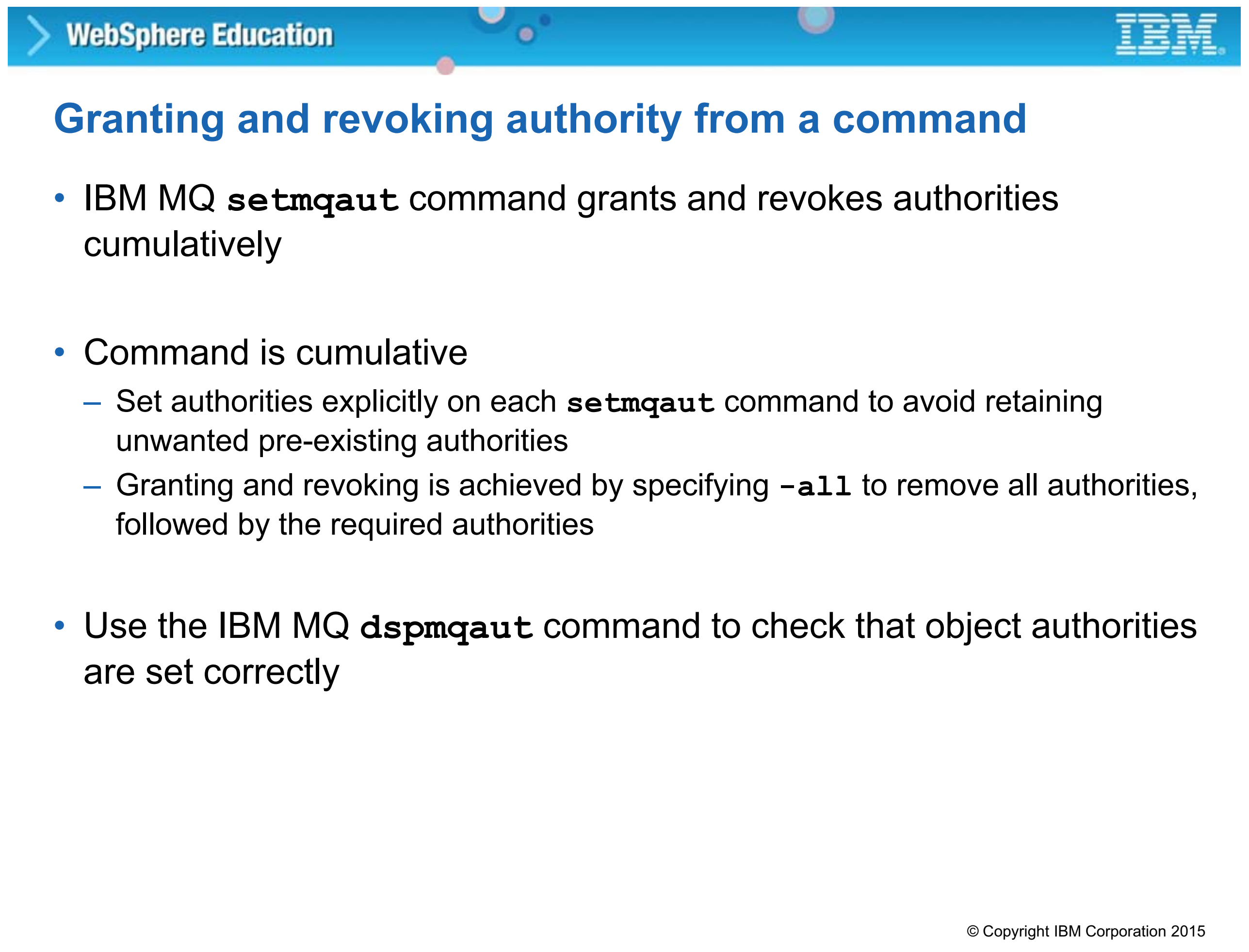
The next steps define the authorities.

3. You can assign authority by group or by user. The **Entity type** field identifies the entity as a group or a user.

   If you are adding a group, enter the group name in the **Entity name** field.

   If you are adding a user, enter the user name in the **Entity name** field.

4. Set the permissions. Integration Bus permissions are set under the **MQI** column.
5. Click **OK** to save the changes. The changes are immediate; you do not have to restart the integration node.



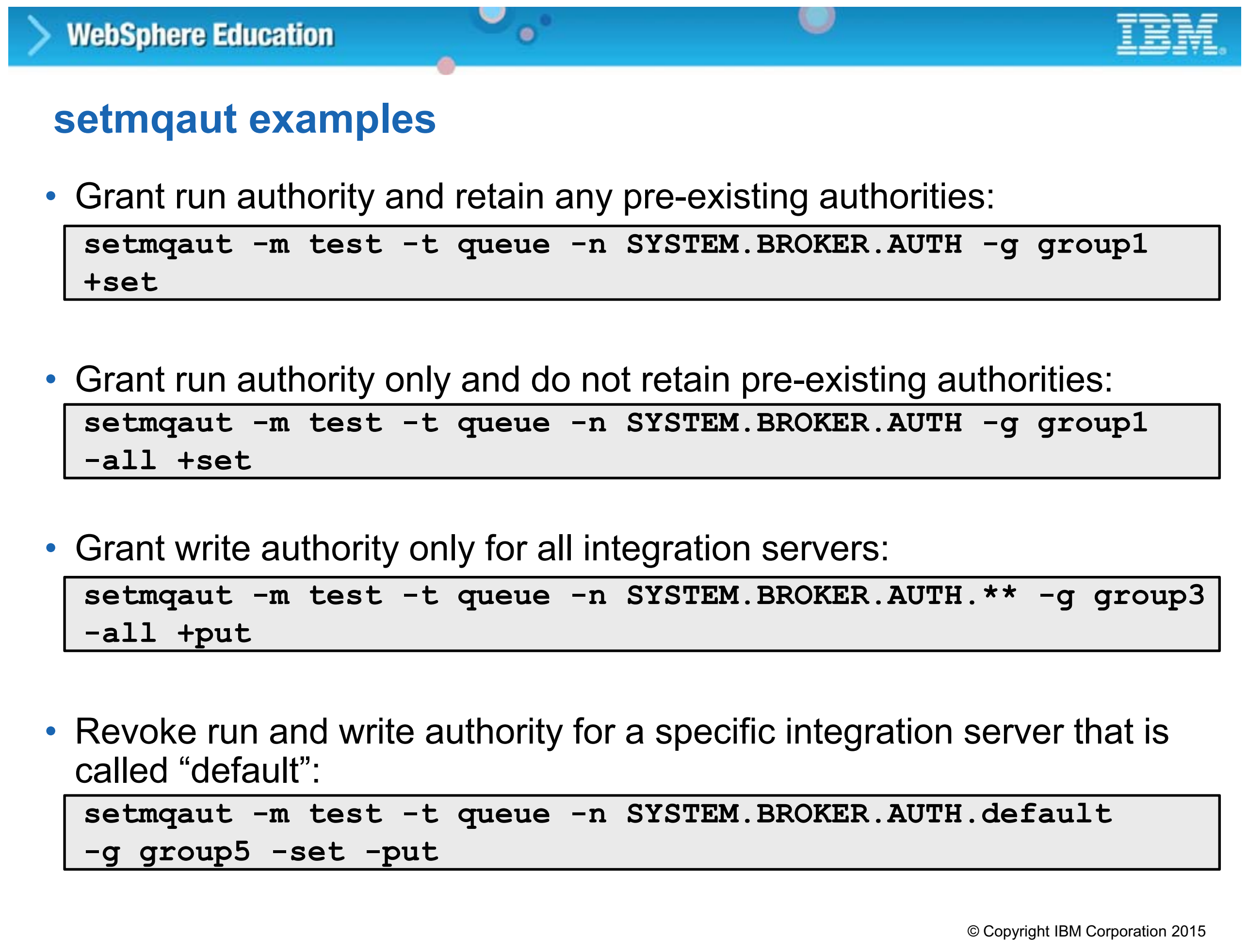
Granting and revoking authority from a command

- IBM MQ **`setmqaut`** command grants and revokes authorities cumulatively
- Command is cumulative
  - Set authorities explicitly on each **`setmqaut`** command to avoid retaining unwanted pre-existing authorities
  - Granting and revoking is achieved by specifying **`-all`** to remove all authorities, followed by the required authorities
- Use the IBM MQ **`dspmqaut`** command to check that object authorities are set correctly

Figure 6-27. Granting and revoking authority from a command WM6461.0

Notes:

You can use the IBM MQ `setmqaut` command to grant and revoke Integration Bus permissions; however, commands are cumulative so always use the `dspmqaut` command to check that the authorities are correct.



WebSphere Education

setmqaut examples

- Grant run authority and retain any pre-existing authorities:

```
setmqaut -m test -t queue -n SYSTEM.BROKER.AUTH -g group1
+set
```

- Grant run authority only and do not retain pre-existing authorities:

```
setmqaut -m test -t queue -n SYSTEM.BROKER.AUTH -g group1
-all +set
```

- Grant write authority only for all integration servers:

```
setmqaut -m test -t queue -n SYSTEM.BROKER.AUTH.** -g group3
-all +put
```

- Revoke run and write authority for a specific integration server that is called “default”:

```
setmqaut -m test -t queue -n SYSTEM.BROKER.AUTH.default
-g group5 -set -put
```

Figure 6-28. `setmqaut examples` WM6461.0

Notes:

The figure shows examples of using the `setmqaut` command to set Integration Bus administrative permissions.

A plus sign before the authorization, adds the permission. For example, `+set` grants “set” (run) authority.

A minus sign before the permission, removes the permission. For example, `-put` removes “put” (write) authority.

The following options are shown on the `setmqaut` command:

`-m` specifies the queue manager name

`-t` identifies the object type as queue

`-n` identifies the Integration Bus authorization queue

`-g` specifies the group name

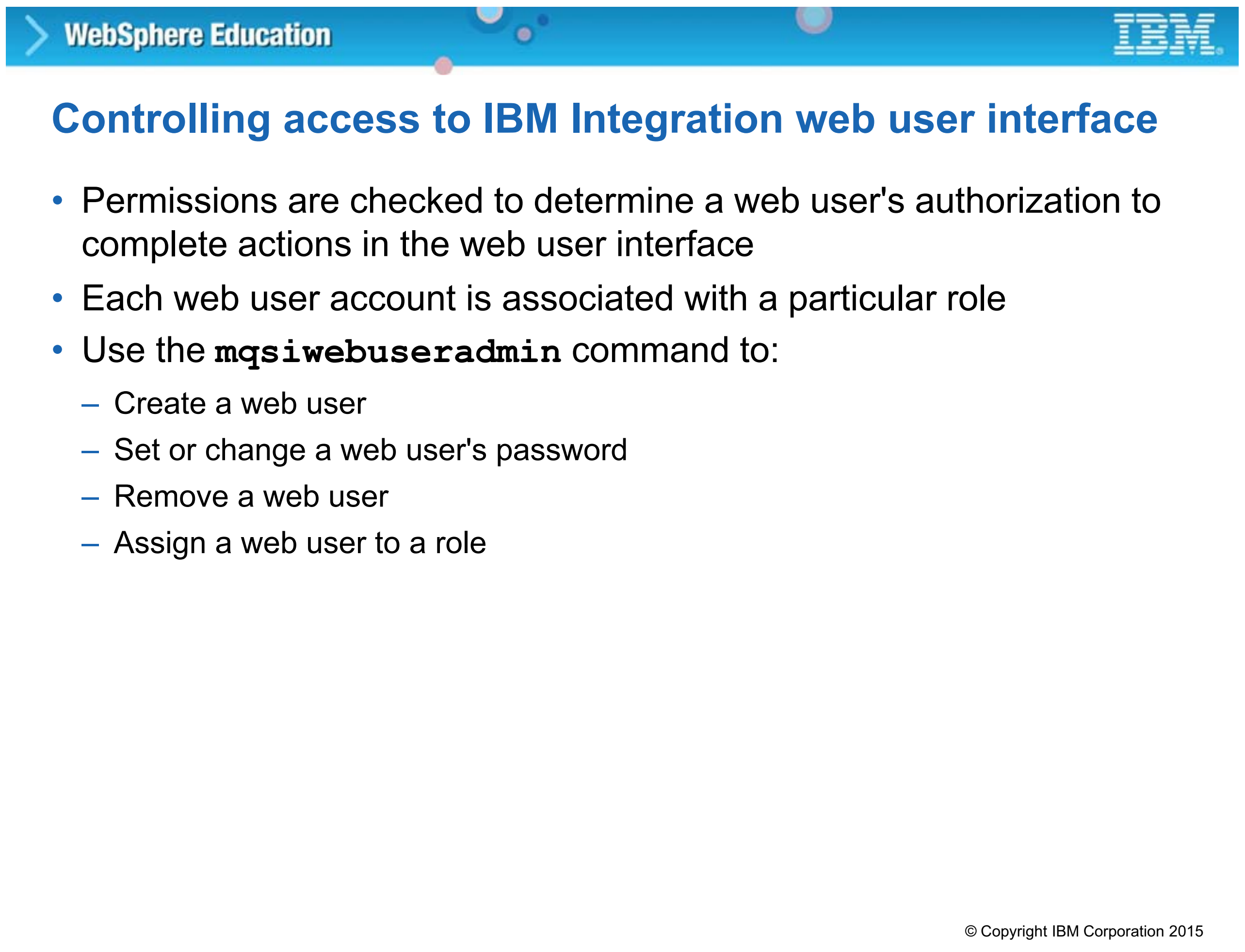
`-p` specifies the principal (user) name

`-all` removes all authorities before granting the authorities that follow this parameter

To avoid retaining unwanted pre-existing authorities, set authorities explicitly on each `setmqaut` command, rather than granting and revoking individual authorities.

Granting and revoking is achieved by specifying `-all` (to remove all authorities) followed by the required authorities.

6.4. IBM Integration web user interface security



WebSphere Education

Controlling access to IBM Integration web user interface

- Permissions are checked to determine a web user's authorization to complete actions in the web user interface
- Each web user account is associated with a particular role
- Use the **`mqsiwebuseradmin`** command to:
  - Create a web user
  - Set or change a web user's password
  - Remove a web user
  - Assign a web user to a role

© Copyright IBM Corporation 2015

Figure 6-29. Controlling access to IBM Integration web user interface WM6461.0

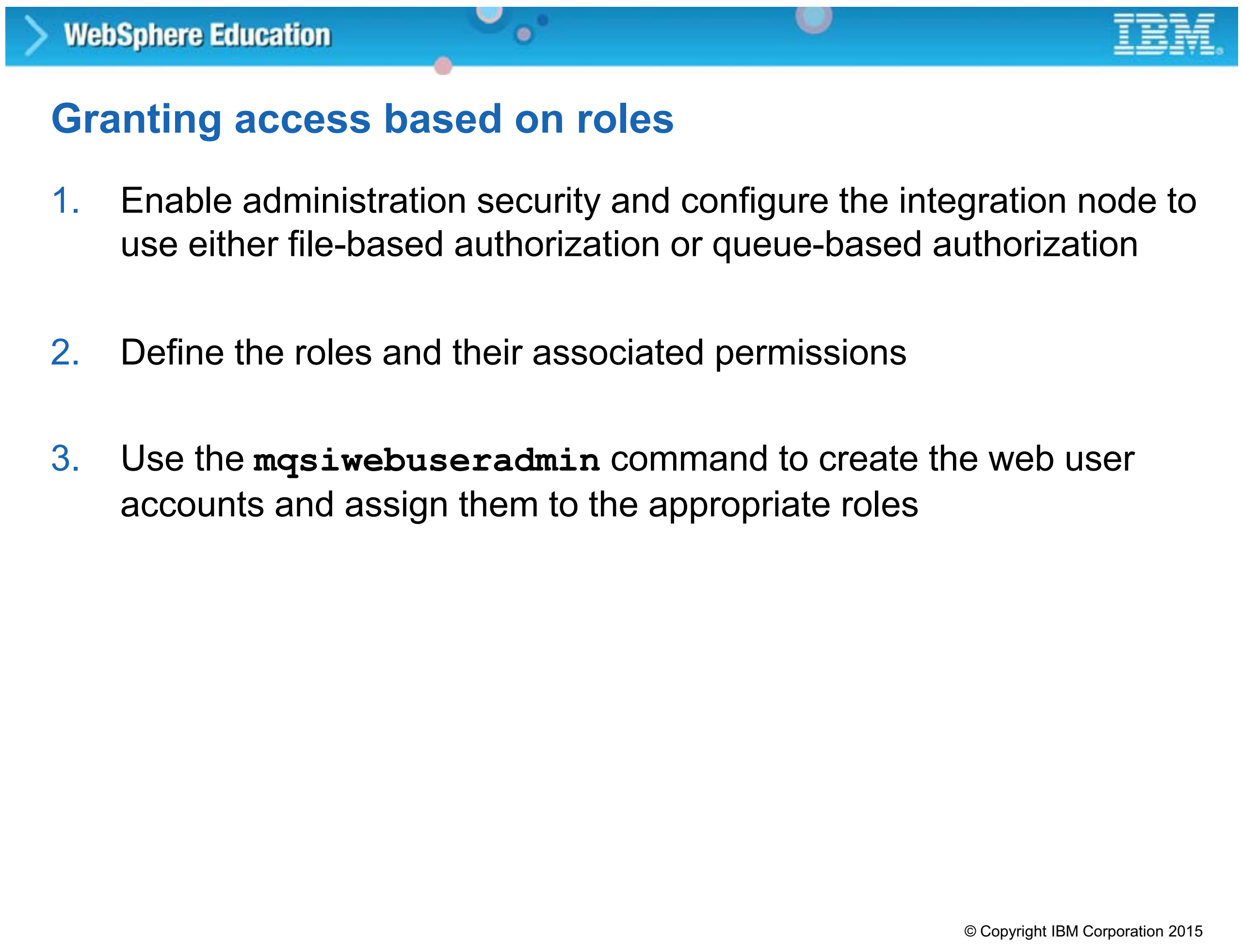
Notes:

You can control a web user's access to IBM Integration Bus resources by associating the web user ID with a role, which has security permissions that are assigned to it.

Administrators can use the `mqsiwebuseradmin` command to complete the following tasks:

- Create a web user
- Set or change a web user's password
- Remove a web user
- Assign a web user to a role

As an Integration Bus administrator, you can set up multiple system user IDs on the system that the integration node is running on, with different permissions by using file-based or web-based authorizations. These permissions then apply to web users through their assigned role.



Granting access based on roles

1. Enable administration security and configure the integration node to use either file-based authorization or queue-based authorization
2. Define the roles and their associated permissions
3. Use the **`mqsiwebuseradmin`** command to create the web user accounts and assign them to the appropriate roles

Figure 6-30. Granting access based on roles WM6461.0

Notes:

Integration administrators can also allow web users to start and stop integration servers, applications, and message flows from the web user interface, by granting permissions to the roles with which the web users are associated.

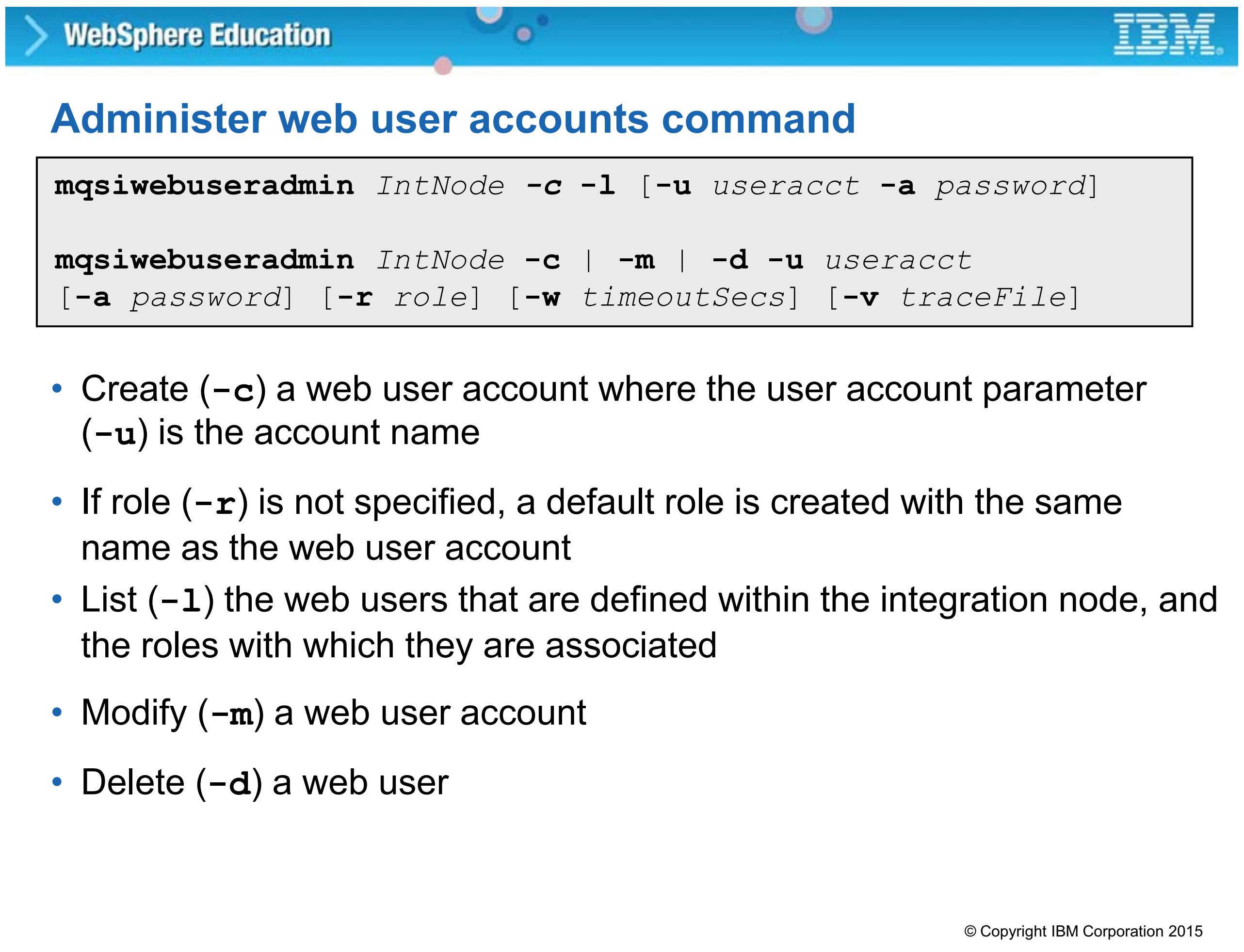
First, enable administration security for your integration node by using the `mqsichangeauthmode` command with the `-s active` parameter.

Next, define the roles and associated permissions. For example, you might decide that your web users can be categorized into two main roles: web administrators and web users. These users typically require different authorizations to complete tasks, such as permission to view or modify resources, according to their role.

- If the integration node is configured to use file-based authorization, define the roles and associated permissions on the integration node, by using the `mqsichangefileauth` command.
- If the integration node is configured to use queue-based authorization, you must create a system user ID on the operating system for each role that you identify. You must then assign permissions to the system user ID, which is then used as a role.

Finally, use the `mqsiwebuseradmin` command to create the web user accounts and assign them to the appropriate roles. If you are using queue-based authorization, the user ID that is used to run this command requires the following authorizations on the SYSTEM.BROKER.AUTH queue:

- To list users, the user ID requires `+inq` (read) authority
- To create, modify, or delete users, the user ID requires `+inq` (read) and `+put` (write) authority



Administer web user accounts command

```
mqsiwebuseradmin IntNode -c -l [-u useracct -a password]

mqsiwebuseradmin IntNode -c | -m | -d -u useracct
[-a password] [-r role] [-w timeoutSecs] [-v traceFile]
```

- Create (`-c`) a web user account where the user account parameter (`-u`) is the account name
- If role (`-r`) is not specified, a default role is created with the same name as the web user account
- List (`-l`) the web users that are defined within the integration node, and the roles with which they are associated
- Modify (`-m`) a web user account
- Delete (`-d`) a web user

Figure 6-31. Administer web user accounts command WM6461.0

Notes:

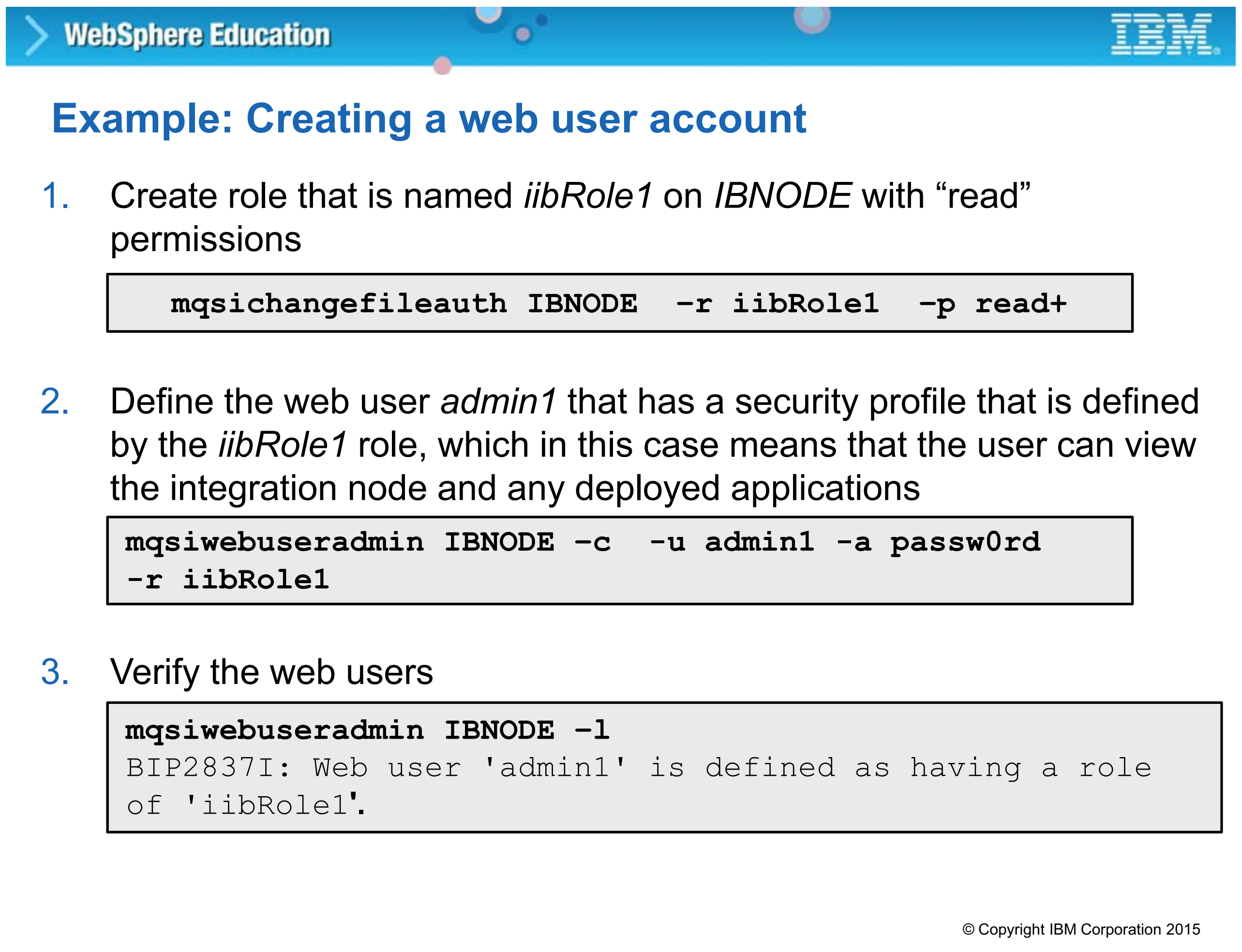
Use the `mqsiwebuseradmin` command to administer user accounts for the web user interface.

Note

The integration node must be started when you run the command. The command takes effect immediately.

You must specify at least one parameter to identify the target integration node for this command, in one of the following forms:

- Name of a locally defined integration node
- Name of a file with the connection details for a local or remote integration node
- Connection details (IP address, port, and queue manager) for the integration node



Example: Creating a web user account

1. Create role that is named *iibRole1* on *IBNODE* with “read” permissions

```
  mqsichangefileauth IBNODE  -r iibRole1  -p read+
```

2. Define the web user *admin1* that has a security profile that is defined by the *iibRole1* role, which in this case means that the user can view the integration node and any deployed applications

```
mqsiwebuseradmin IBNODE -c  -u admin1 -a passw0rd
-r iibRole1
```

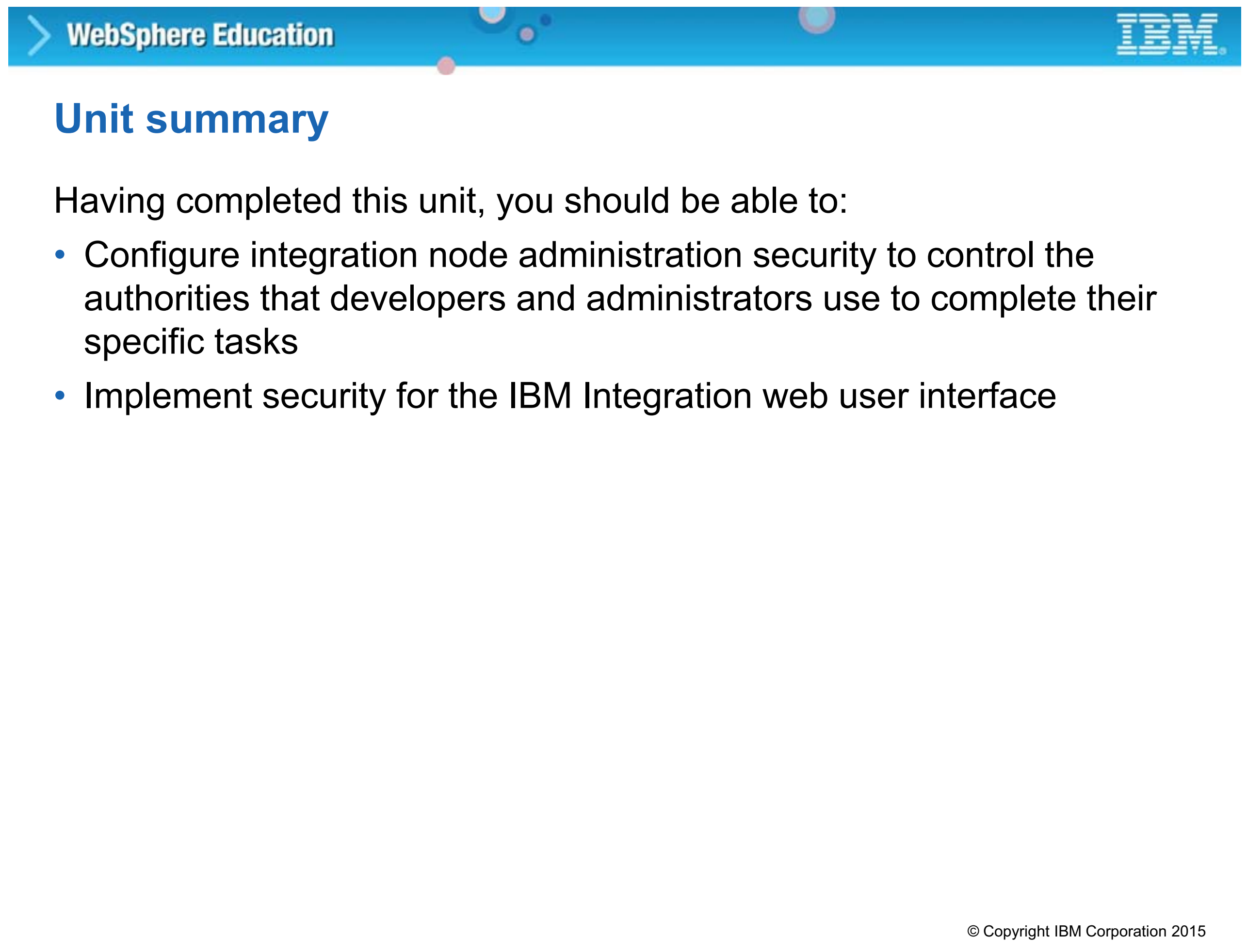
3. Verify the web users

```
mqsiwebuseradmin IBNODE -l
BIP2837I: Web user 'admin1' is defined as having a role
of 'iibRole1'.
```

Figure 6-32. Example: Creating a web user account WM6461.0

Notes:

The figure shows and example of how to create a web user account by using the `mqsichangefileauth` and `mqsiwebuseradmin` commands.



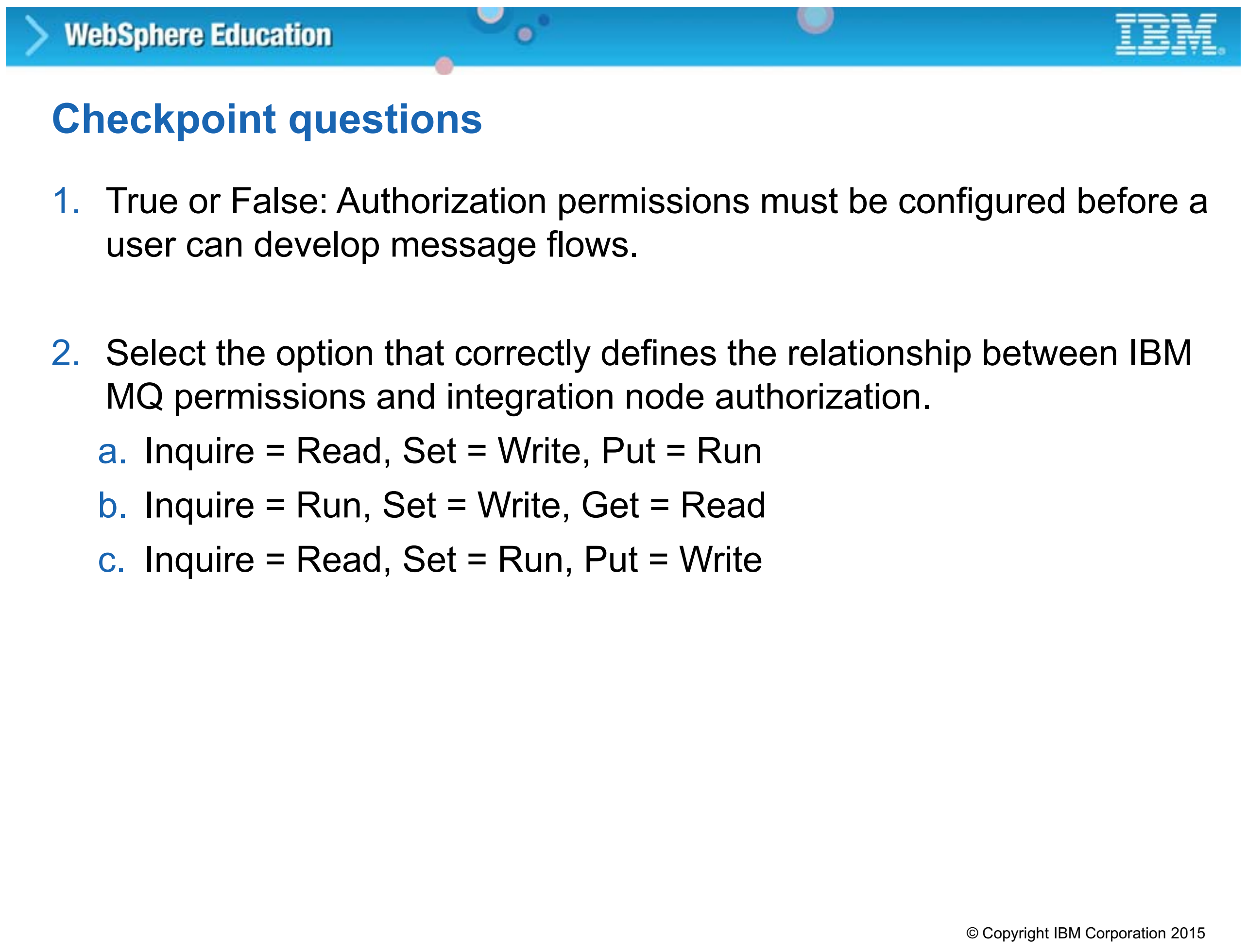
Unit summary

Having completed this unit, you should be able to:

- Configure integration node administration security to control the authorities that developers and administrators use to complete their specific tasks
- Implement security for the IBM Integration web user interface

Figure 6-33. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. True or False: Authorization permissions must be configured before a user can develop message flows.

2. Select the option that correctly defines the relationship between IBM MQ permissions and integration node authorization.
   a. Inquire = Read, Set = Write, Put = Run
   b. Inquire = Run, Set = Write, Get = Read
   c. Inquire = Read, Set = Run, Put = Write

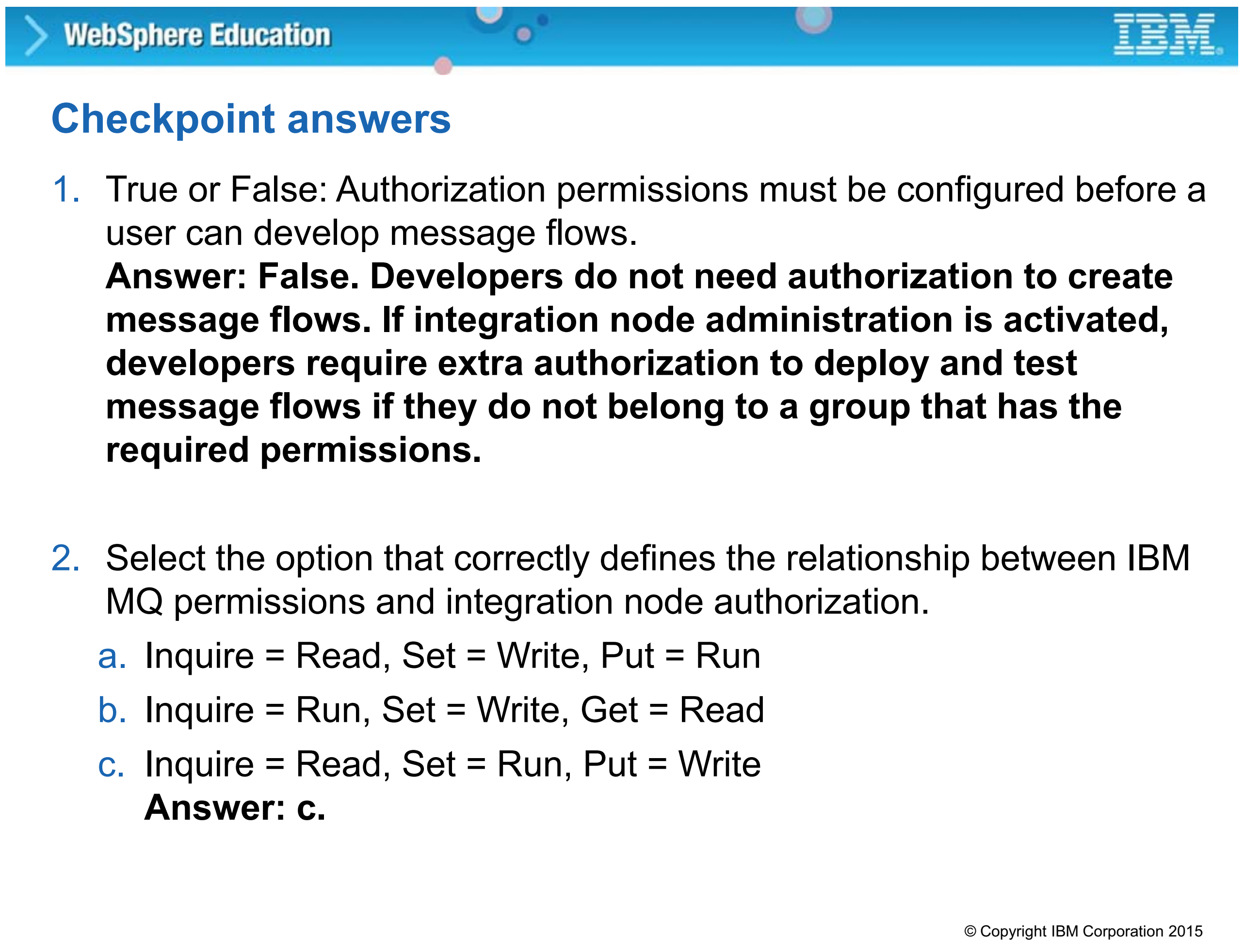
Figure 6-34. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education IBM

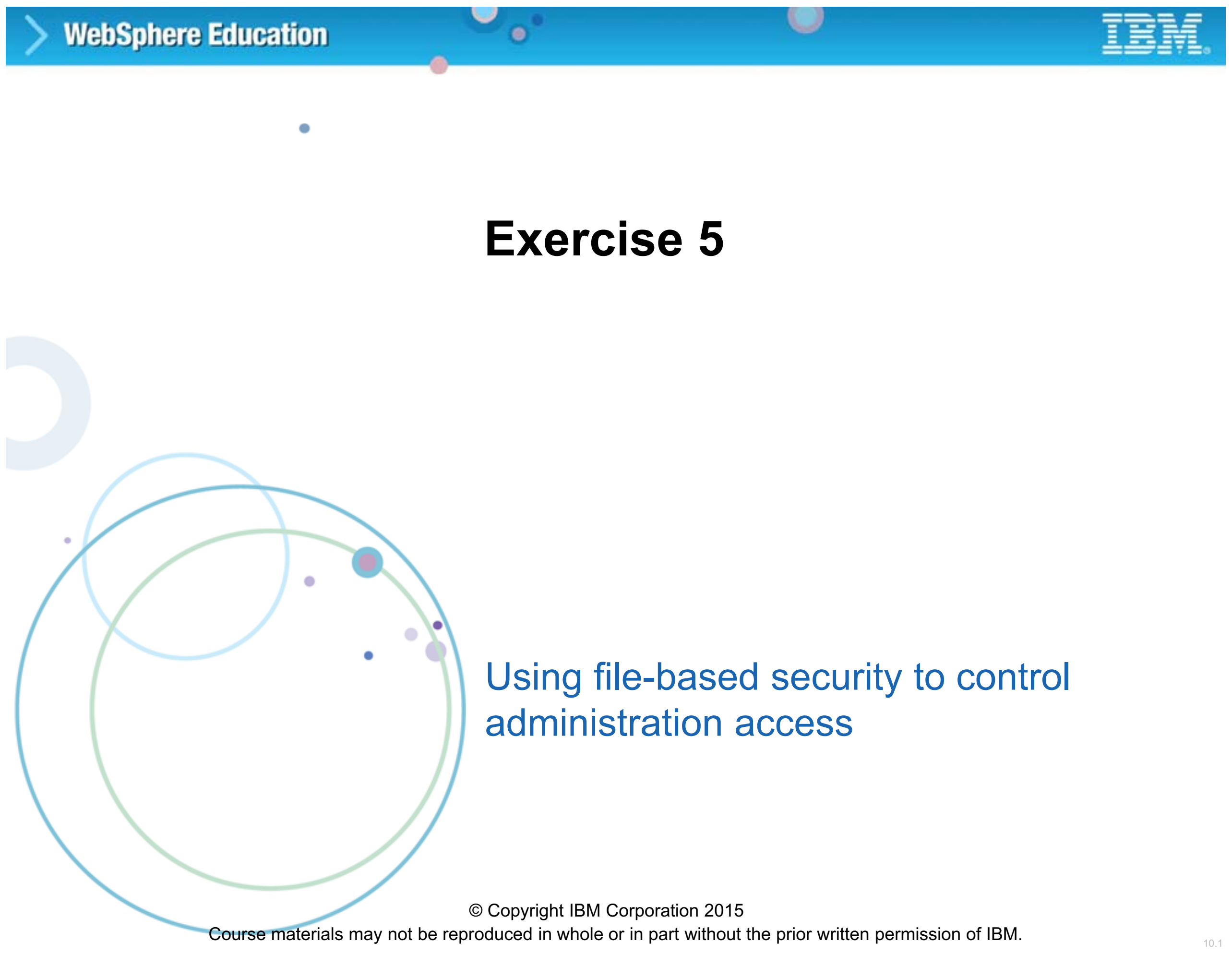
Checkpoint answers

1. True or False: Authorization permissions must be configured before a user can develop message flows.
   Answer: False. Developers do not need authorization to create message flows. If integration node administration is activated, developers require extra authorization to deploy and test message flows if they do not belong to a group that has the required permissions.

2. Select the option that correctly defines the relationship between IBM MQ permissions and integration node authorization.
   a. Inquire = Read, Set = Write, Put = Run
   b. Inquire = Run, Set = Write, Get = Read
   c. Inquire = Read, Set = Run, Put = Write
   Answer: c.

Figure 6-35. Checkpoint answers WM6461.0

Notes:



WebSphere Education

Exercise 5

Using file-based security to control administration access

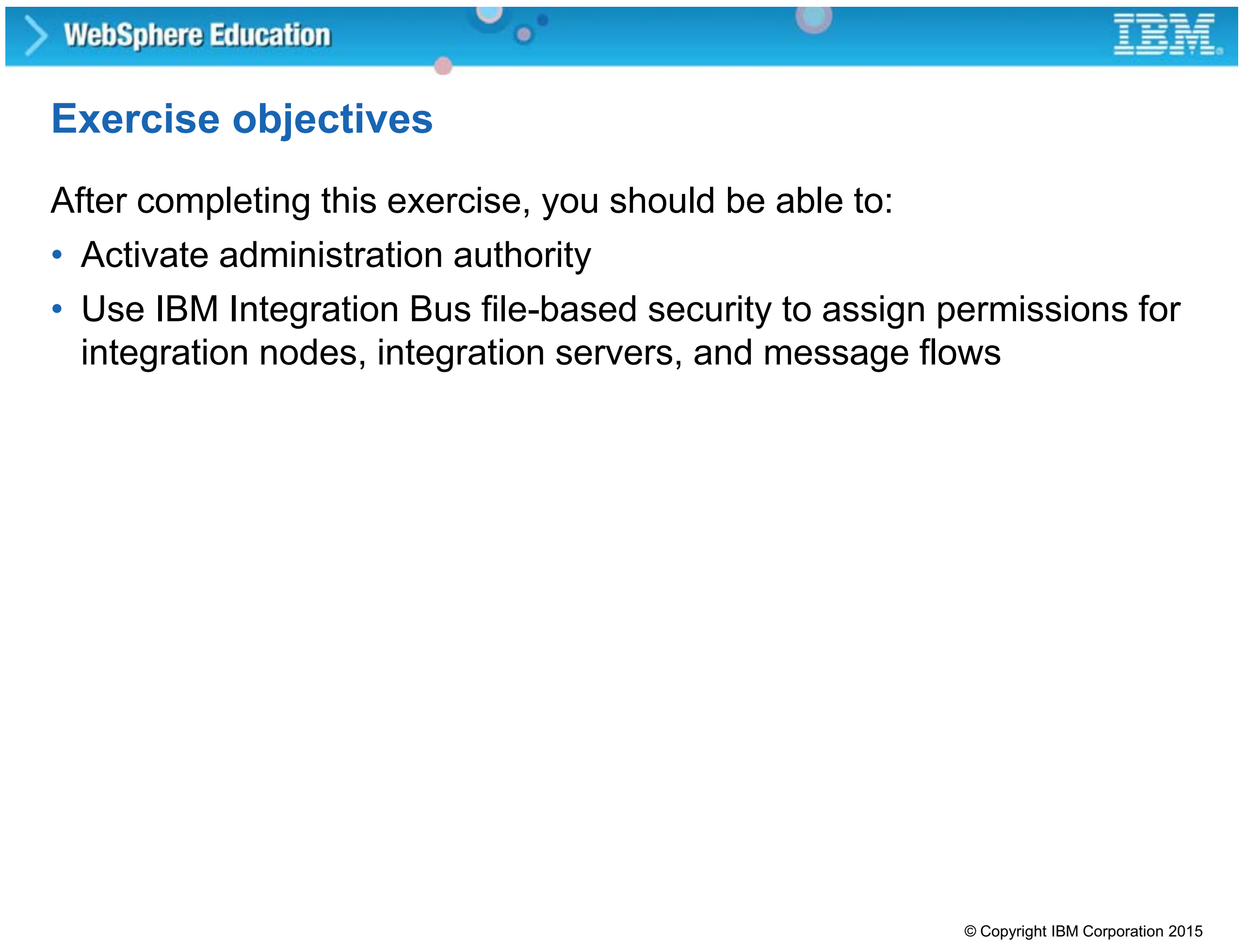
© Copyright IBM Corporation 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

10.1

Figure 6-36. Exercise 5 WM6461.0

Notes:

In this exercise, you use roles and file-based security to control administration access to the IBM Integration web user interface, integration nodes, integration servers, and message flows.



Exercise objectives

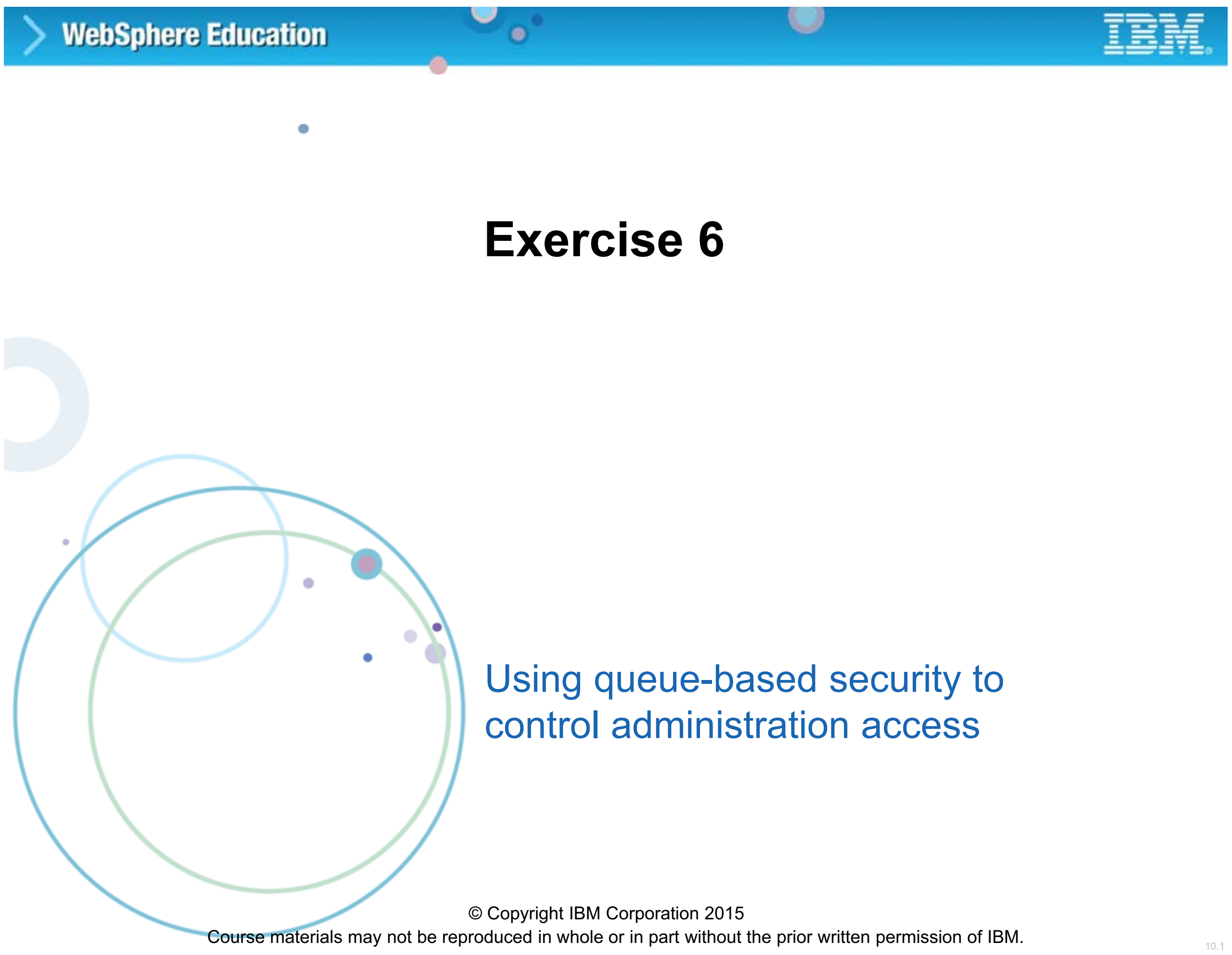
After completing this exercise, you should be able to:

- Activate administration authority
- Use IBM Integration Bus file-based security to assign permissions for integration nodes, integration servers, and message flows

Figure 6-37. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.


WebSphere Education
IBM
10.1

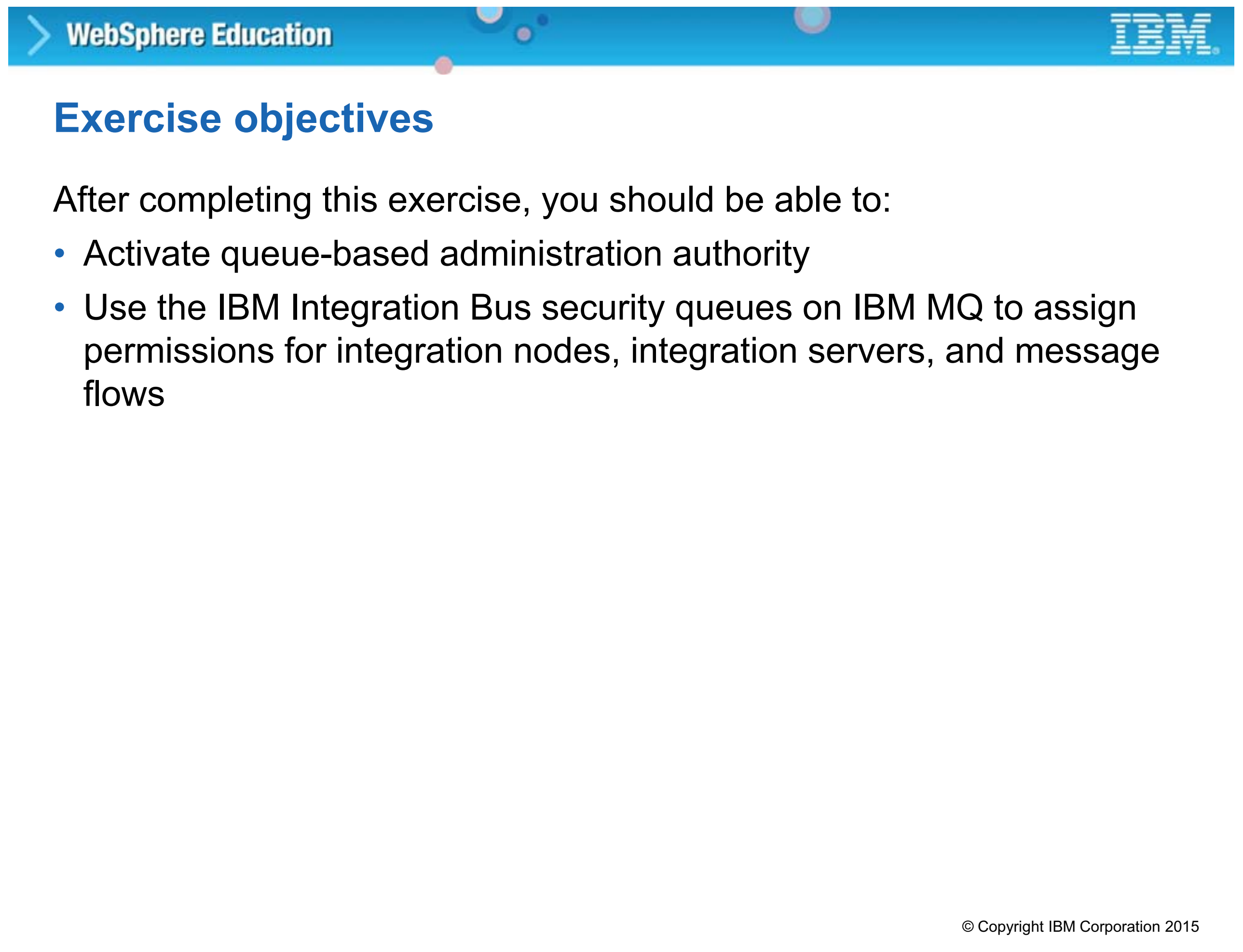
Exercise 6

Using queue-based security to control administration access

Figure 6-38. Exercise 6 WM6461.0

Notes:

In this exercise, you set and test IBM Integration Bus access control for the integration nodes, integration servers, and message flows.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Activate queue-based administration authority
- Use the IBM Integration Bus security queues on IBM MQ to assign permissions for integration nodes, integration servers, and message flows

© Copyright IBM Corporation 2015

Figure 6-39. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 7. Implementing IBM Integration Bus message flow security

What this unit is about

This unit describes how to implement message flow security by using the Lightweight Directory Access Protocol (LDAP) and a WS-Trust V1.3 compliant Security Token Service (STS), such as Tivoli Federated Identity Manager V6.2.

What you should be able to do

After completing this unit, you should be able to:

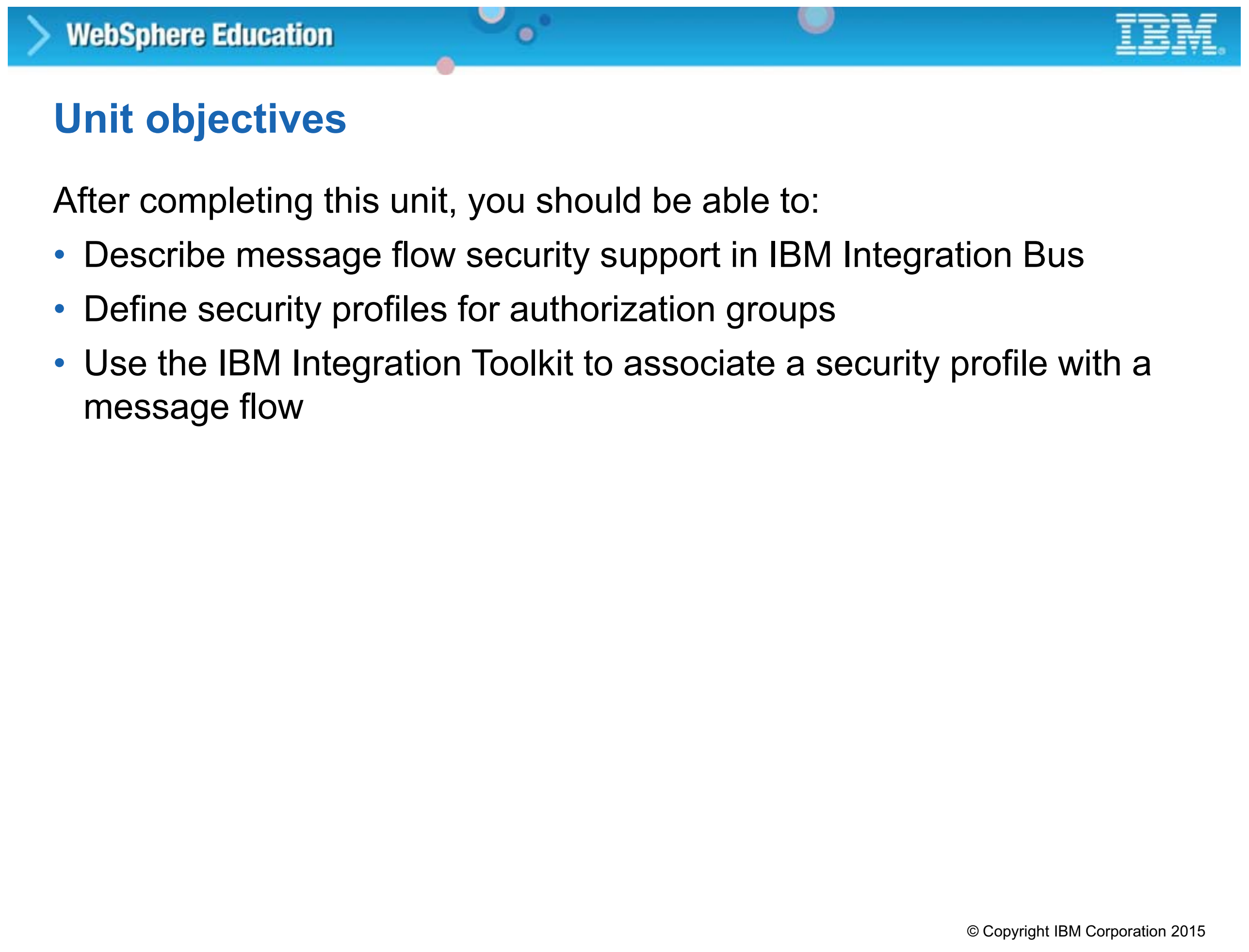
- Describe message flow security support in IBM Integration Bus
- Define security profiles for authorization groups
- Use the IBM Integration Toolkit to associate a security profile with a message flow

How you will check your progress

- Checkpoints

References

IBM Knowledge Center



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe message flow security support in IBM Integration Bus
- Define security profiles for authorization groups
- Use the IBM Integration Toolkit to associate a security profile with a message flow

© Copyright IBM Corporation 2015

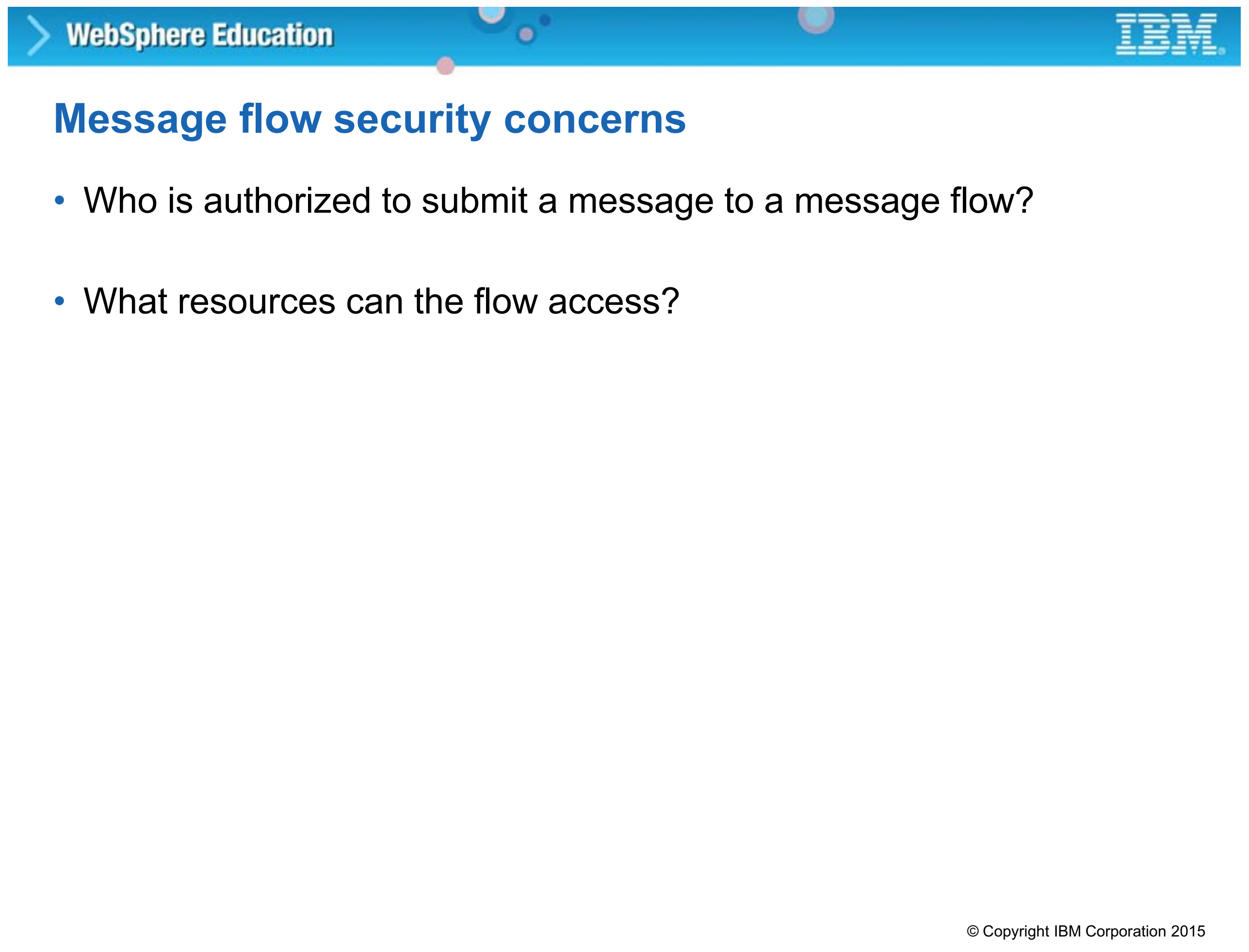
Figure 7-1. Unit objectives WM6461.0

Notes:

Security is an important consideration for developers of IBM Integration Bus applications. When a developer designs an IBM Integration Bus application, they must consider the security measures that are needed to protect the data and data content.

This unit describes the implementation of message flow security.

7.1. Message flow security overview



WebSphere Education

Message flow security concerns

- Who is authorized to submit a message to a message flow?
- What resources can the flow access?

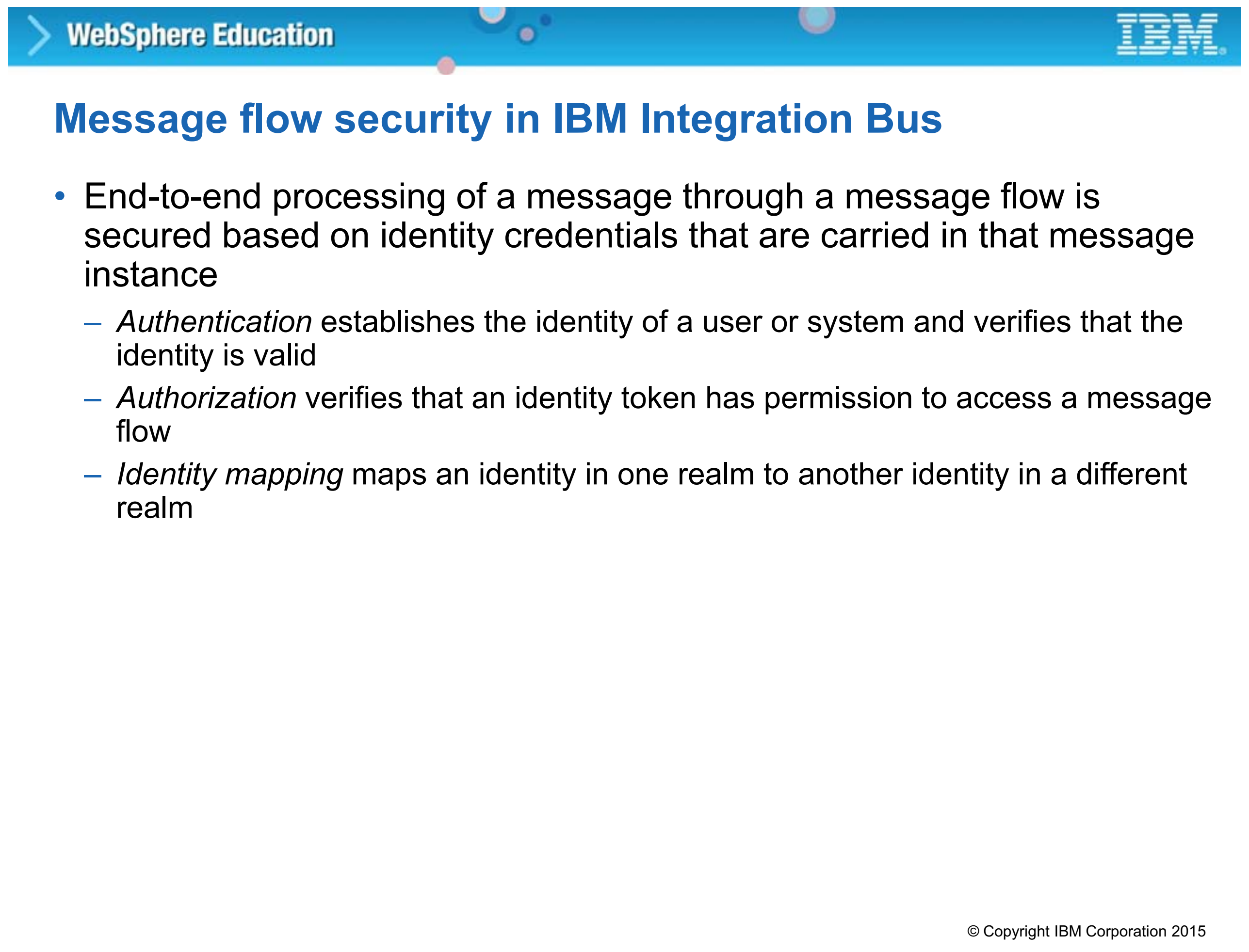
© Copyright IBM Corporation 2015

Figure 7-2. Message flow security concerns WM6461.0

Notes:

Integration Bus provides a security manager to control access to individual messages in a message flow, by using the identity of the message.

You can configure an integration node for processing end-to-end an identity that is carried in a message through a message flow by using a security profile.



Message flow security in IBM Integration Bus

- End-to-end processing of a message through a message flow is secured based on identity credentials that are carried in that message instance
  - *Authentication* establishes the identity of a user or system and verifies that the identity is valid
  - *Authorization* verifies that an identity token has permission to access a message flow
  - *Identity mapping* maps an identity in one realm to another identity in a different realm

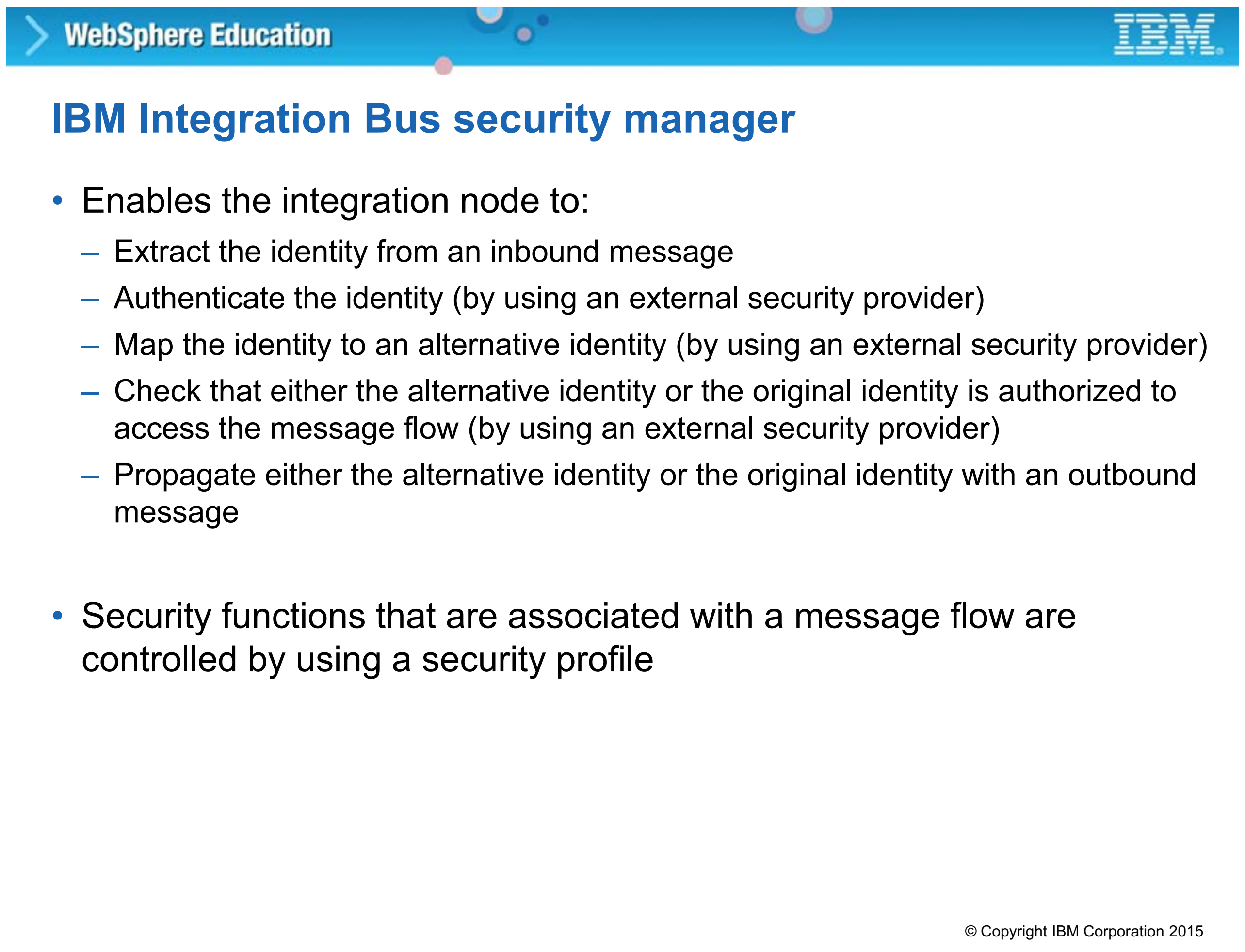
Figure 7-3. Message flow security in IBM Integration Bus WM6461.0

Notes:

IBM Integration Bus provides a security manager, which controls access to individual messages in a message flow, by using the identity of the message.

You can configure the integration node to do end-to-end processing of identity credentials that are carried in a message through a message flow. This security manager can:

- Extract the identity from an inbound message
- Authenticate the identity (by using an external security provider)
- Map the identity to an alternative identity (by using an external security provider)
- Check that either the alternative identity or the original identity is authorized to access the message flow (by using an external security provider)
- Propagate either the alternative identity or the original identity with an outbound message



WebSphere Education

IBM Integration Bus security manager

- Enables the integration node to:
  - Extract the identity from an inbound message
  - Authenticate the identity (by using an external security provider)
  - Map the identity to an alternative identity (by using an external security provider)
  - Check that either the alternative identity or the original identity is authorized to access the message flow (by using an external security provider)
  - Propagate either the alternative identity or the original identity with an outbound message
- Security functions that are associated with a message flow are controlled by using a security profile

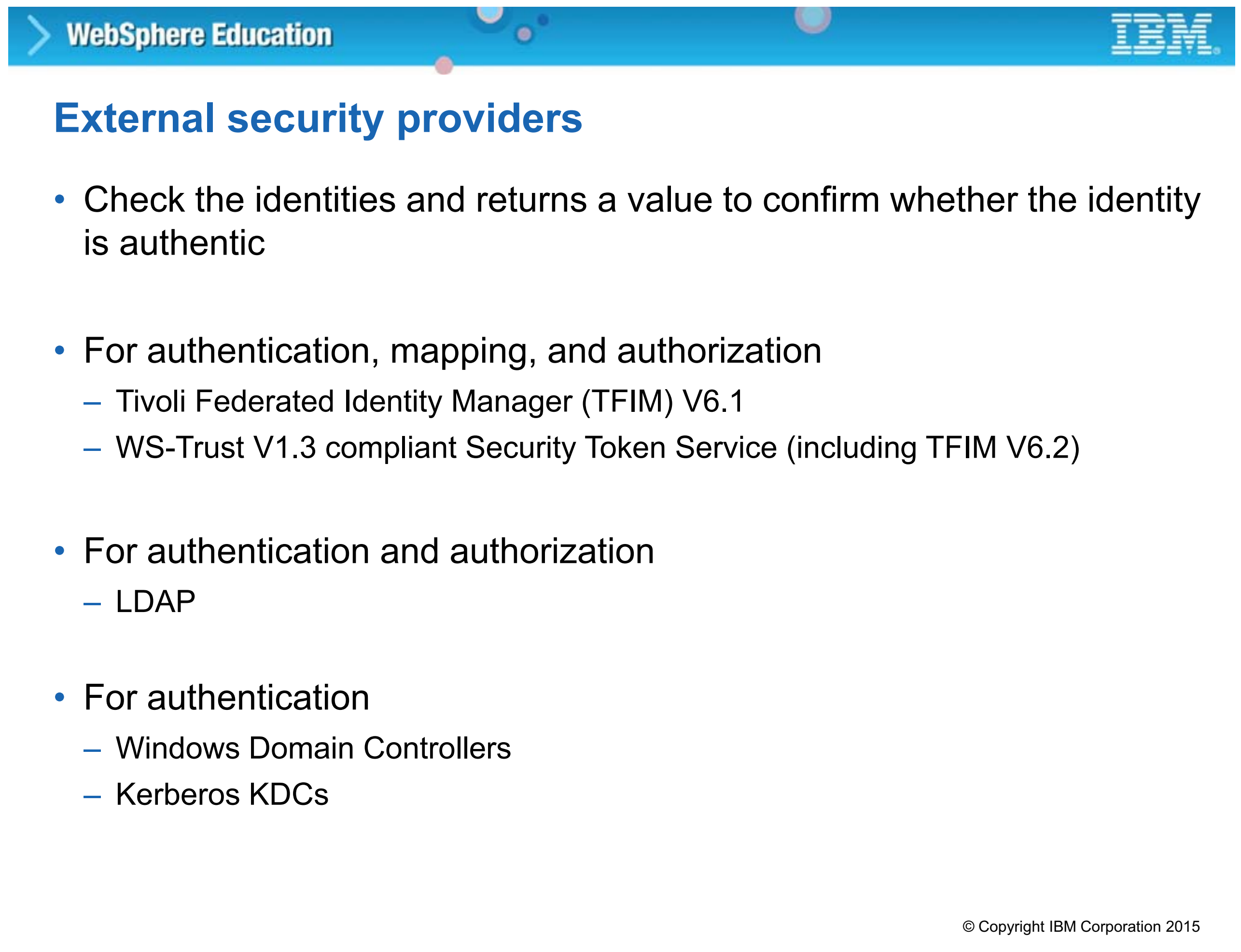
© Copyright IBM Corporation 2015

Figure 7-4. IBM Integration Bus security manager WM6461.0

Notes:

The figure lists the processes that the security manager enables in the integration node.

By creating a security profile, you can configure security for a message flow to control access based on the identity that is associated with the message. A security profile provides a security mechanism that is independent of both the transport type and message format.



WebSphere Education

External security providers

- Check the identities and returns a value to confirm whether the identity is authentic
- For authentication, mapping, and authorization
  - Tivoli Federated Identity Manager (TFIM) V6.1
  - WS-Trust V1.3 compliant Security Token Service (including TFIM V6.2)
- For authentication and authorization
  - LDAP
- For authentication
  - Windows Domain Controllers
  - Kerberos KDCs

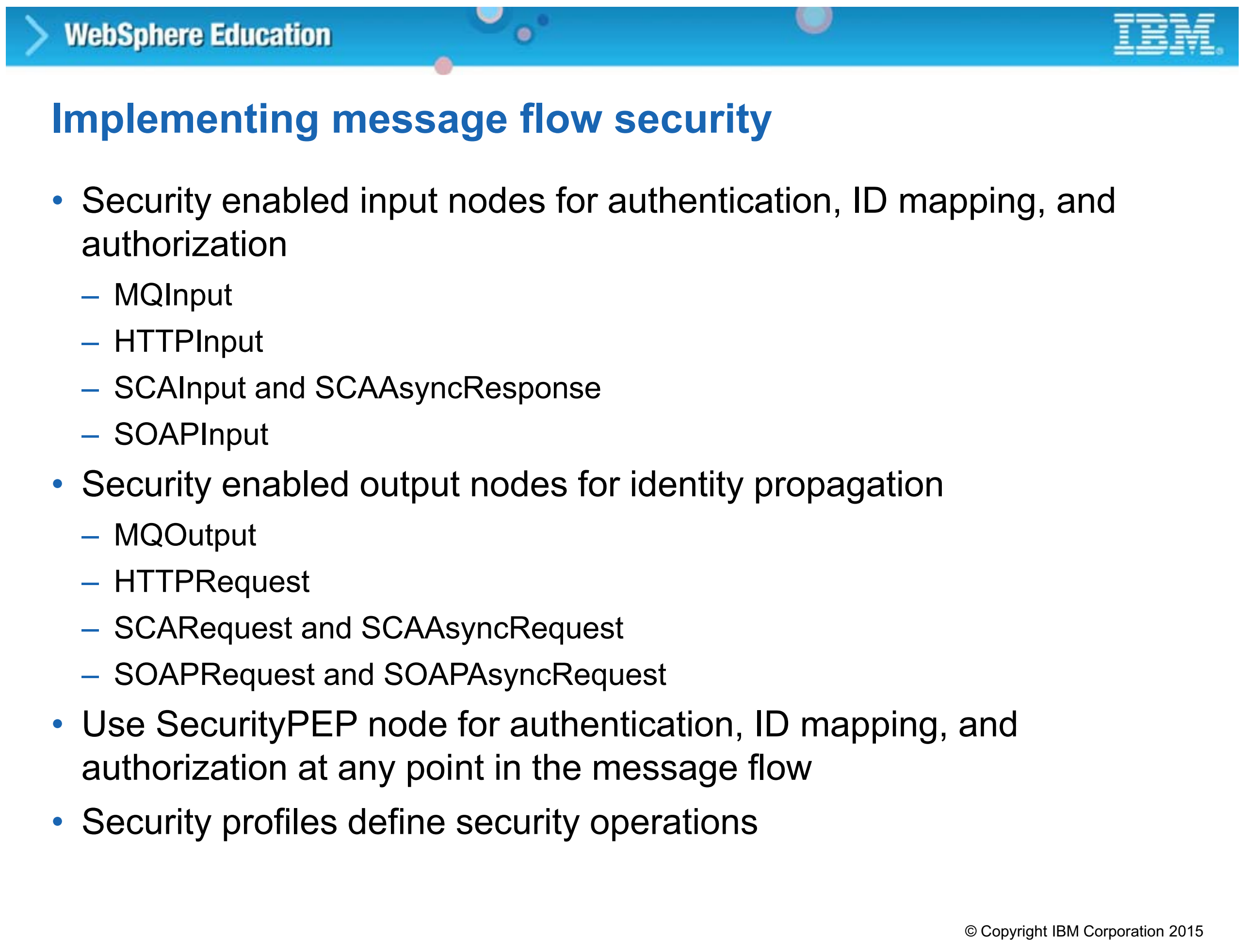
Figure 7-5. External security providers WM6461.0

Notes:

The security functions that are associated with a message flow are controlled by using security profiles. An administrator creates the security profiles that the security manager accesses at run time.

Integration Bus supports the following external security providers:

- WS-Trust V1.3 compliant Security Token Service (including Tivoli Federated Identity Manager V6.2) for authentication, mapping, and authorization
- Tivoli Federated Identity Manager V6.1 for authentication, mapping, and authorization
- Lightweight Directory Access Protocol (LDAP) for authentication and authorization
- Windows Domain Controllers for authentication
- Kerberos KDCs for authentication



Implementing message flow security

- Security enabled input nodes for authentication, ID mapping, and authorization
  - MQInput
  - HTTPInput
  - SCAInput and SCAAsyncResponse
  - SOAPInput
- Security enabled output nodes for identity propagation
  - MQOutput
  - HTTPRequest
  - SCARequest and SCAAsyncRequest
  - SOAPRequest and SOAPAsyncRequest
- Use SecurityPEP node for authentication, ID mapping, and authorization at any point in the message flow
- Security profiles define security operations

Figure 7-6. Implementing message flow security WM6461.0

Notes:

You can start message flow security by configuring either a security enabled input node or a SecurityPEP node.

You can use the SecurityPEP node to start the message flow security manager at any point in the message flow between an input node and an output (or request) node.

A security profile defines the security operations.

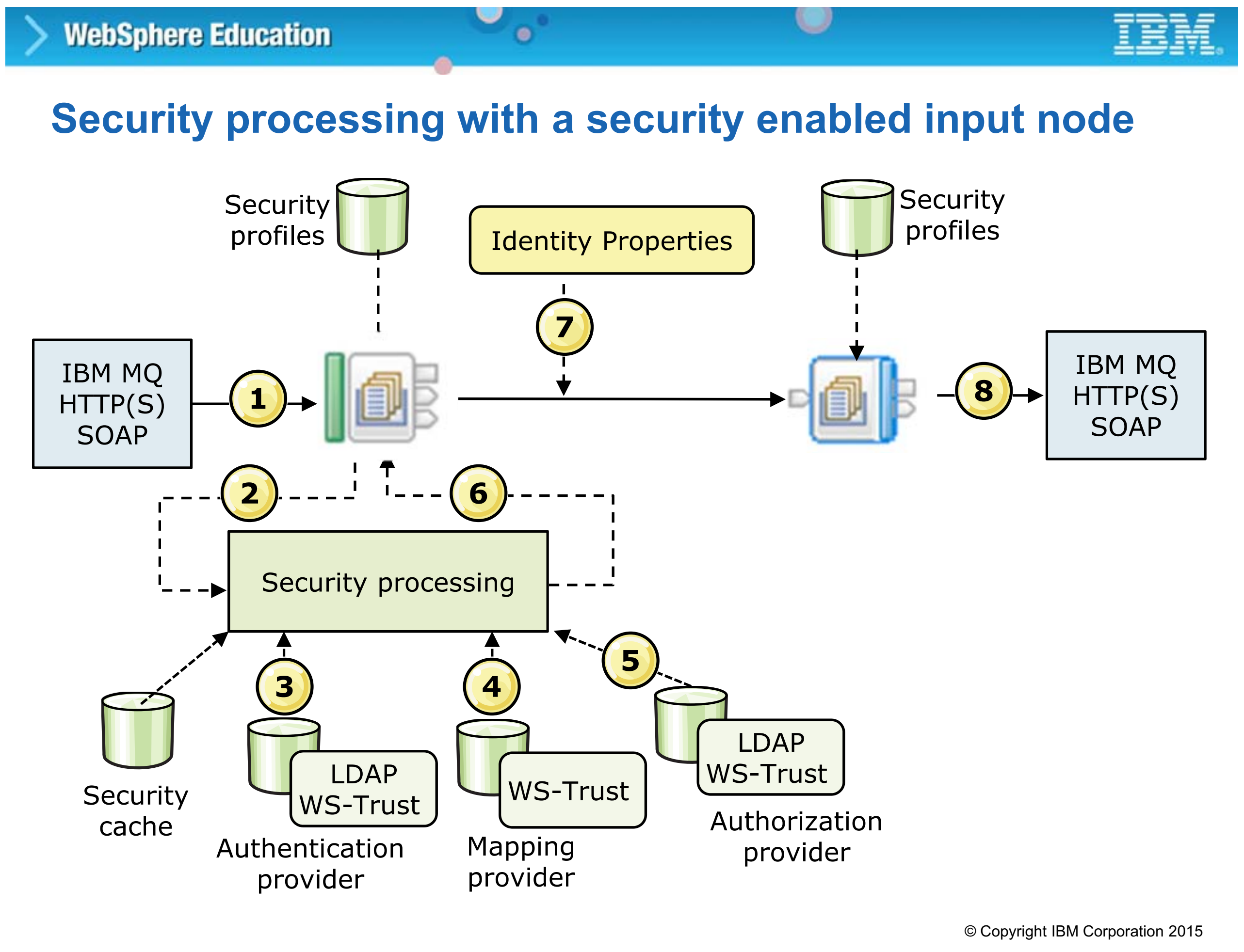


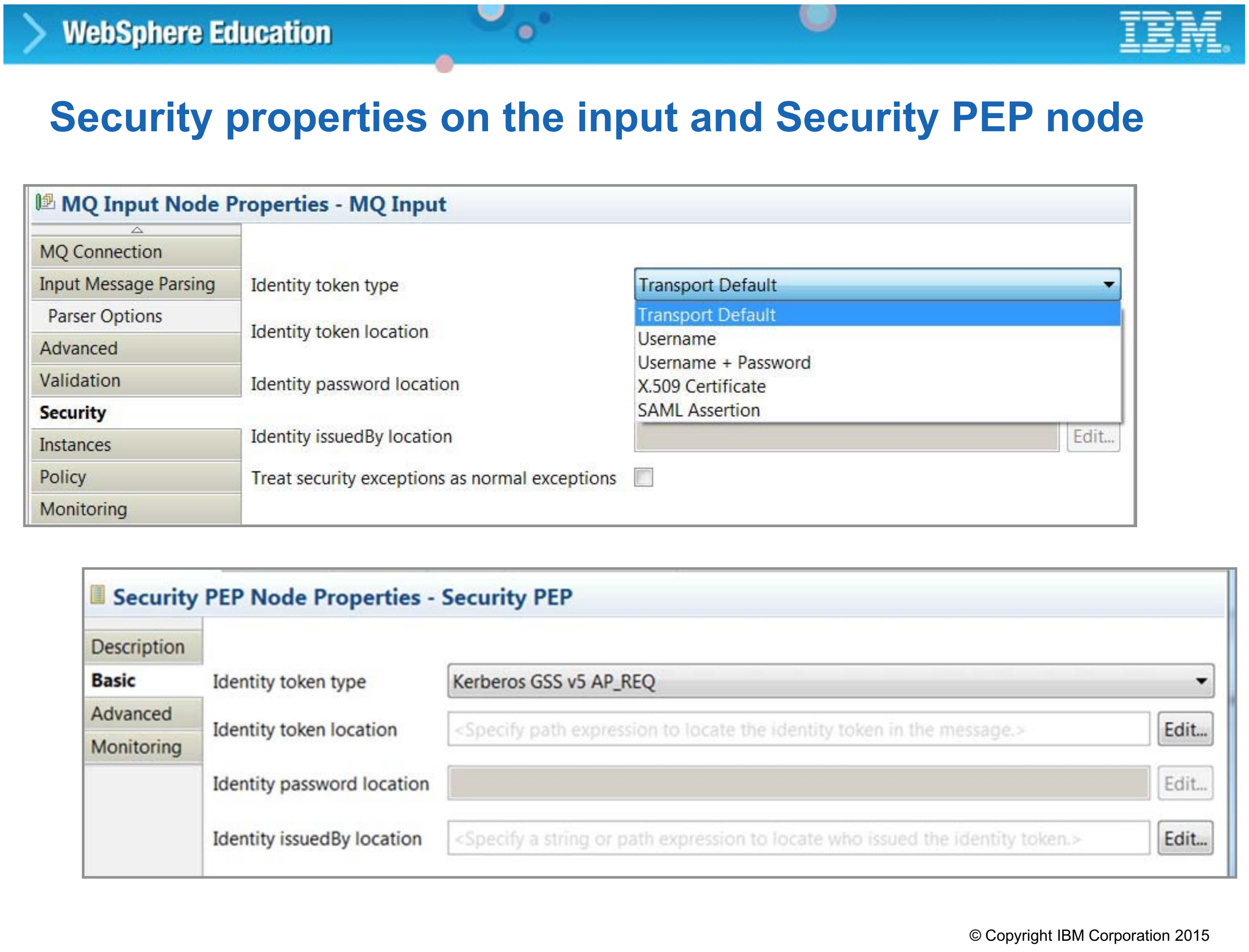
Figure 7-7. Security processing with a security enabled input node WM6461.0

Notes:

This figure shows that sequence of events that occur when security enabled input node in the message flow receives an input message.

1. When a message arrives at a security enabled input node (IBM MQ, HTTP, SCA, or SOAP), the presence of a security profile that is associated with the node indicates whether message flow security is configured.
2. If a security profile is associated with the node or message flow, the security enabled input node extracts the identity information from the input message (based on the configuration on the node's **Security** properties page). It then sets the **Source Identity** elements in the **Properties** folder.
3. If authentication is specified in the security profile, the security manager calls the configured security provider to authenticate the identity. A security cache is provided for the authentication result, which enables subsequent messages (with the same credentials) arriving at the message flow to be completed with the cached result, if it did not expire.
4. If identity mapping is specified in the security profile, the security manager calls the configured security provider to map the identity to an alternative identity.

5. If authorization is specified in the security profile, the security manager calls the configured security provider to authorize that the identity (either mapped or source) has access to this message flow. A security cache is provided for the authorization result.
6. When all security processing is complete, or when the message flow security manager raises a security exception, control returns to the input node.
7. The message, including the **Properties** folder and its source and mapped identity information, is propagated down the message flow.
8. The propagated identity is included in the appropriate message header when it is sent.



Security properties on the input and Security PEP node

MQ Input Node Properties - MQ Input

MQ Connection		
Input Message Parsing	Identity token type	Transport Default
Parser Options	Identity token location	
Advanced		
Validation	Identity password location	
Security		
Instances	Identity issuedBy location	
Policy	Treat security exceptions as normal exceptions	☐
Monitoring		

Identity token type options: Transport Default, Username, Username + Password, X.509 Certificate, SAML Assertion

Security PEP Node Properties - Security PEP

Description		
Basic	Identity token type	Kerberos GSS v5 AP_REQ
Advanced	Identity token location	<Specify path expression to locate the identity token in the message.>
Monitoring	Identity password location	
	Identity issuedBy location	<Specify a string or path expression to locate who issued the identity token.>

Figure 7-8. Security properties on the input and Security PEP node WM6461.0

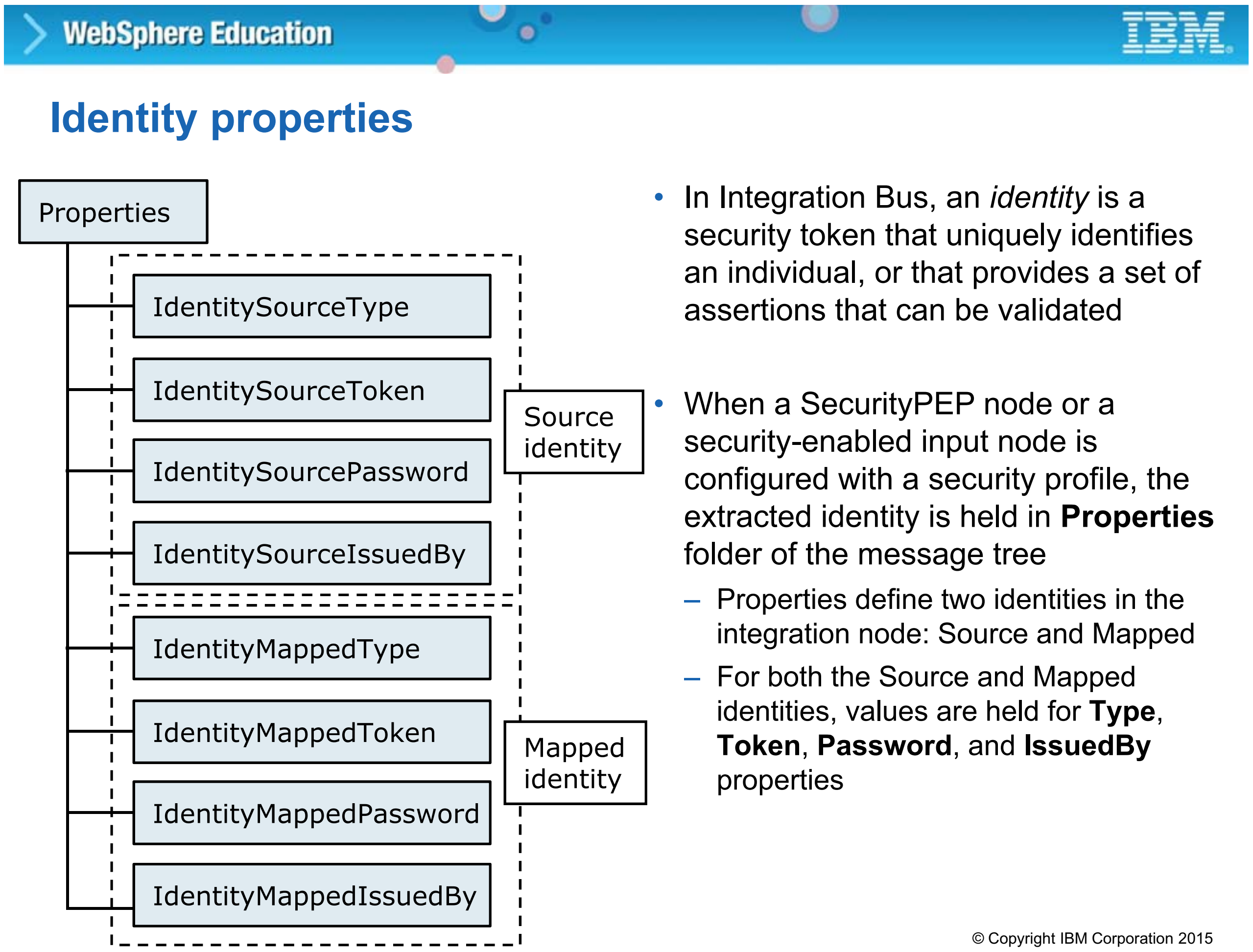
Notes:

This figure shows the security properties that can be configured on an input node (on the **Security** tab) and on the Security PEP node.

For IBM MQ, HTTP, or SCA input nodes, the **Security** properties page is used to configure the extraction of the identity. The **Security** properties page allows the Identity token type and its location to be explicitly configured to control the extraction. This source identity information can be in a message header, the message body, or both.

The SecurityPEP node enables security to be applied in a message flow in the following situations:

- The message flow input node is not enabled for security.
- The message flow input node is enabled for security and might be configured to authenticate the user, but the message flow is required to route or filter the message before the business function is known. In this case, authorization needs to be done later in the message flow logic.
- When the message flow includes multiple output or request nodes, which require a specific identity mapping to be done before each node, to obtain the appropriate security tokens for propagation.



Identity properties

- In Integration Bus, an *identity* is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated
- When a SecurityPEP node or a security-enabled input node is configured with a security profile, the extracted identity is held in **Properties** folder of the message tree
  - Properties define two identities in the integration node: Source and Mapped
  - For both the Source and Mapped identities, values are held for **Type**, **Token**, **Password**, and **IssuedBy** properties

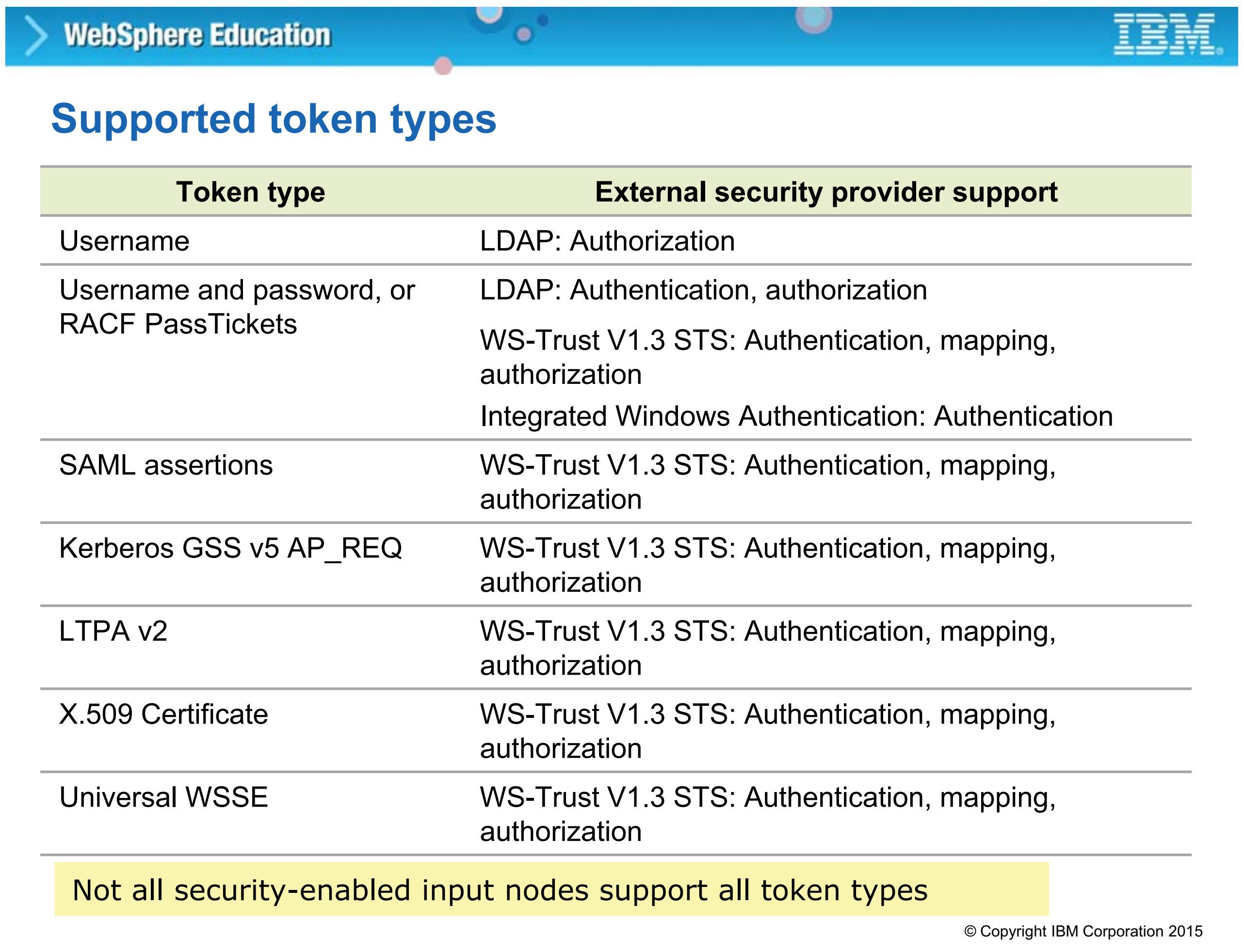
Figure 7-9. Identity properties WM6461.0

Notes:

If a security profile is associated with the node or message flow, the security-enabled input node extracts the identity information from the input message (based on the configuration on the node's **Security** properties page). It then sets the **Source Identity** elements in the **Properties** folder.

In Integration Bus, an identity is a security token that uniquely identifies an individual, or that provides a set of assertions that can be validated.

When a SecurityPEP node or a supported input node is configured with a security profile, the extracted identity is held in the integration node as eight properties in the **Properties** folder of the message tree structure. These properties define two identities in the integration node: source and mapped.



Supported token types

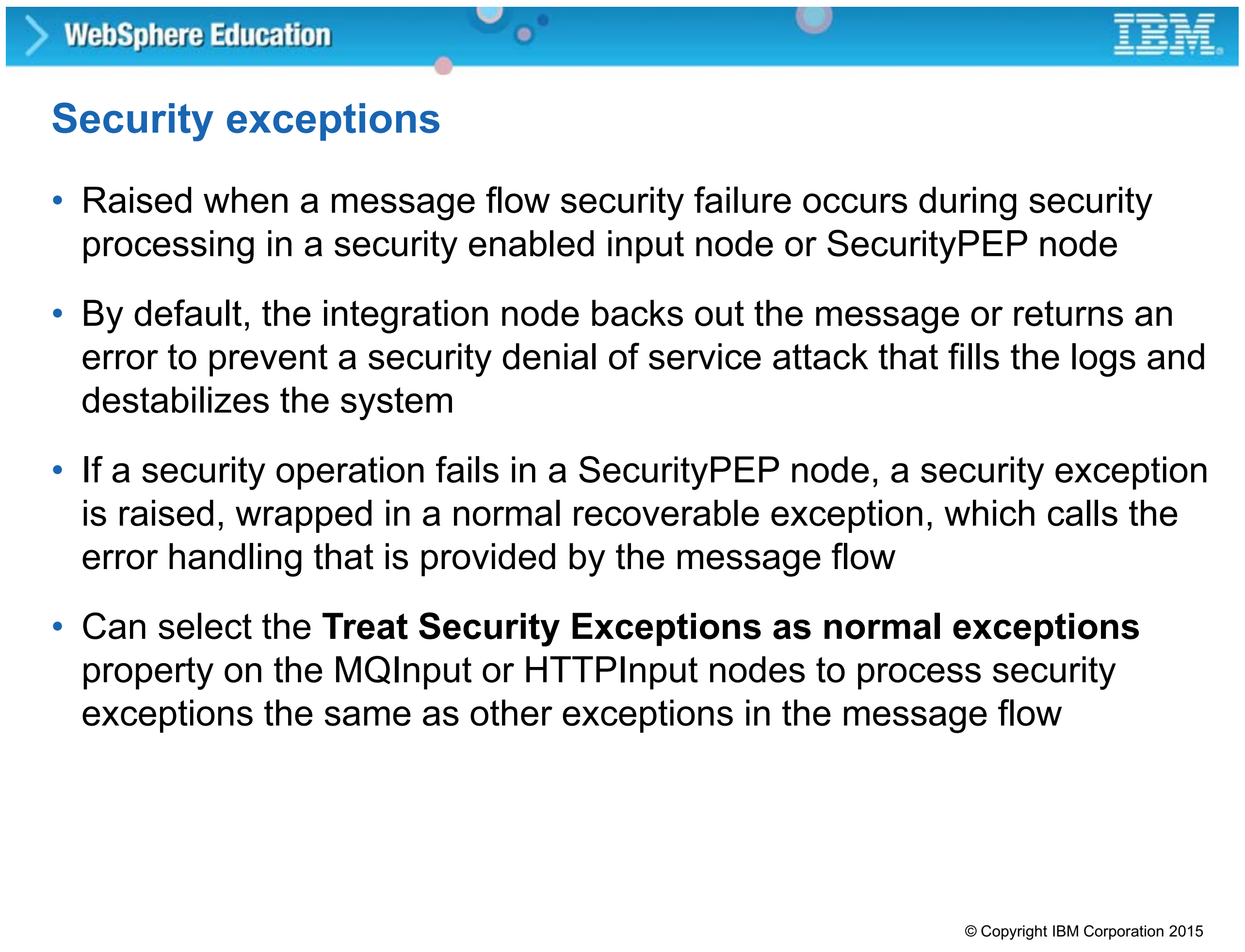
Token type	External security provider support
Username	LDAP: Authorization
Username and password, or RACF PassTickets	LDAP: Authentication, authorization WS-Trust V1.3 STS: Authentication, mapping, authorization Integrated Windows Authentication: Authentication
SAML assertions	WS-Trust V1.3 STS: Authentication, mapping, authorization
Kerberos GSS v5 AP_REQ	WS-Trust V1.3 STS: Authentication, mapping, authorization
LTPA v2	WS-Trust V1.3 STS: Authentication, mapping, authorization
X.509 Certificate	WS-Trust V1.3 STS: Authentication, mapping, authorization
Universal WSSE	WS-Trust V1.3 STS: Authentication, mapping, authorization

Not all security-enabled input nodes support all token types

Figure 7-10. Supported token types WM6461.0

Notes:

This table lists the identity token types and the external security provider that Integration Bus supports.



WebSphere Education

Security exceptions

- Raised when a message flow security failure occurs during security processing in a security enabled input node or SecurityPEP node
- By default, the integration node backs out the message or returns an error to prevent a security denial of service attack that fills the logs and destabilizes the system
- If a security operation fails in a SecurityPEP node, a security exception is raised, wrapped in a normal recoverable exception, which calls the error handling that is provided by the message flow
- Can select the **Treat Security Exceptions as normal exceptions** property on the MQInput or HTTPInput nodes to process security exceptions the same as other exceptions in the message flow

© Copyright IBM Corporation 2015

Figure 7-11. Security exceptions WM6461.0

Notes:

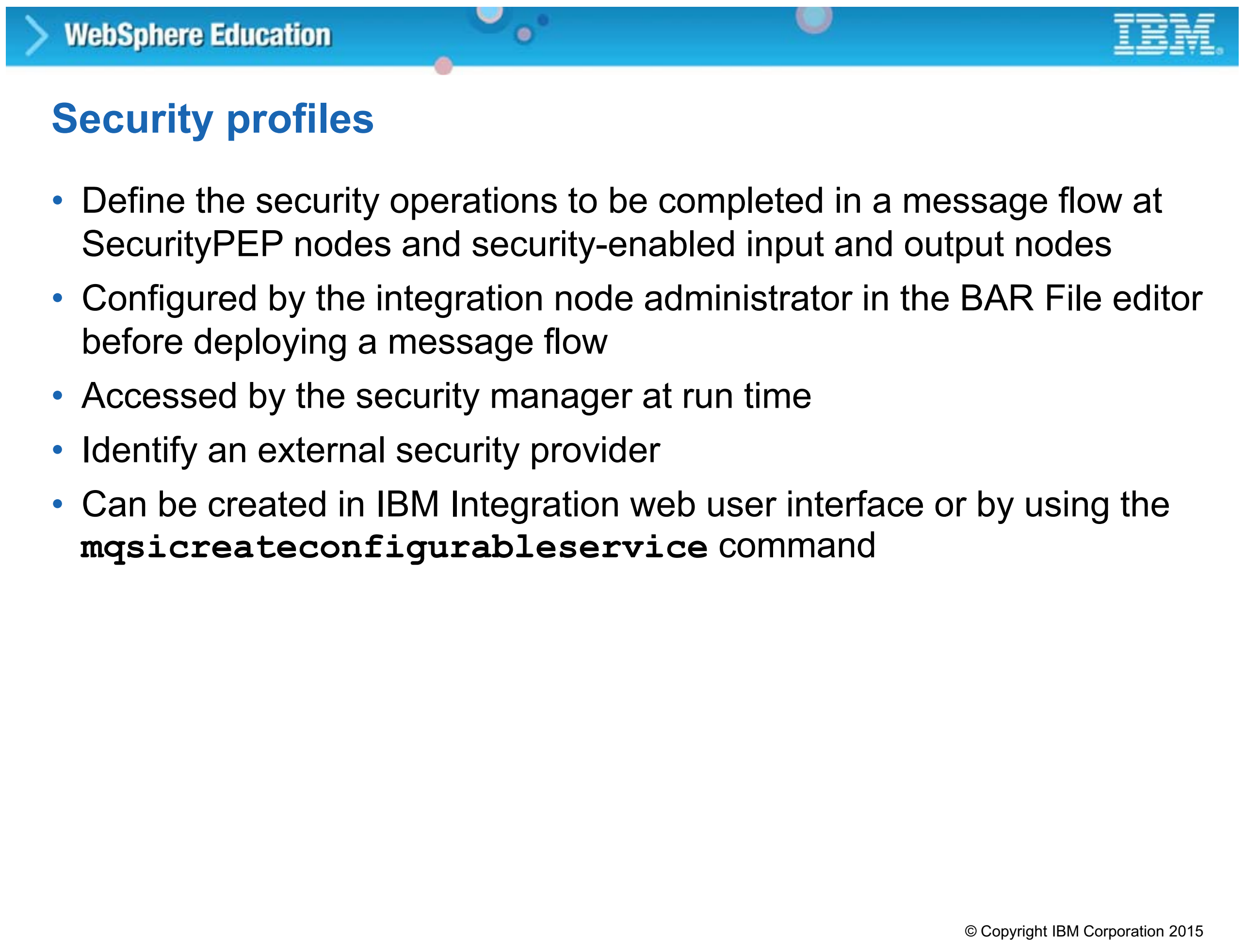
A security exception is raised when a message flow security failure occurs during security processing in an input node or SecurityPEP node.

Security exceptions are processed in a different way from other errors on the input node. An error is typically caught on the input node and routed down the **Failure** terminal for error processing. By default, the integration node does not allow security exceptions to be caught within the message flow, but backs out the message or returns an error (as in the case of HTTP). Security exceptions in input nodes are managed in this way to prevent a security denial of service attack that fills the logs and destabilizes the system.

Security exceptions in SecurityPEP nodes are managed in a different way. If a security operation fails in a SecurityPEP node, a security exception is raised, wrapped in a normal recoverable exception, which starts the error handling that is provided by the message flow.

If the message flow is designed to run in a secure area and you want to explicitly process security exceptions, select the **Treat Security Exceptions as normal exceptions** property on the MQInput or HTTPInput nodes. This property causes security exceptions to be processed in the same way as other exceptions in the message flow.

7.2. Security profiles



WebSphere Education IBM

Security profiles

- Define the security operations to be completed in a message flow at SecurityPEP nodes and security-enabled input and output nodes
- Configured by the integration node administrator in the BAR File editor before deploying a message flow
- Accessed by the security manager at run time
- Identify an external security provider
- Can be created in IBM Integration web user interface or by using the **`mqsicreateconfigurableservice`** command

© Copyright IBM Corporation 2015

Figure 7-12. Security profiles WM6461.0

Notes:

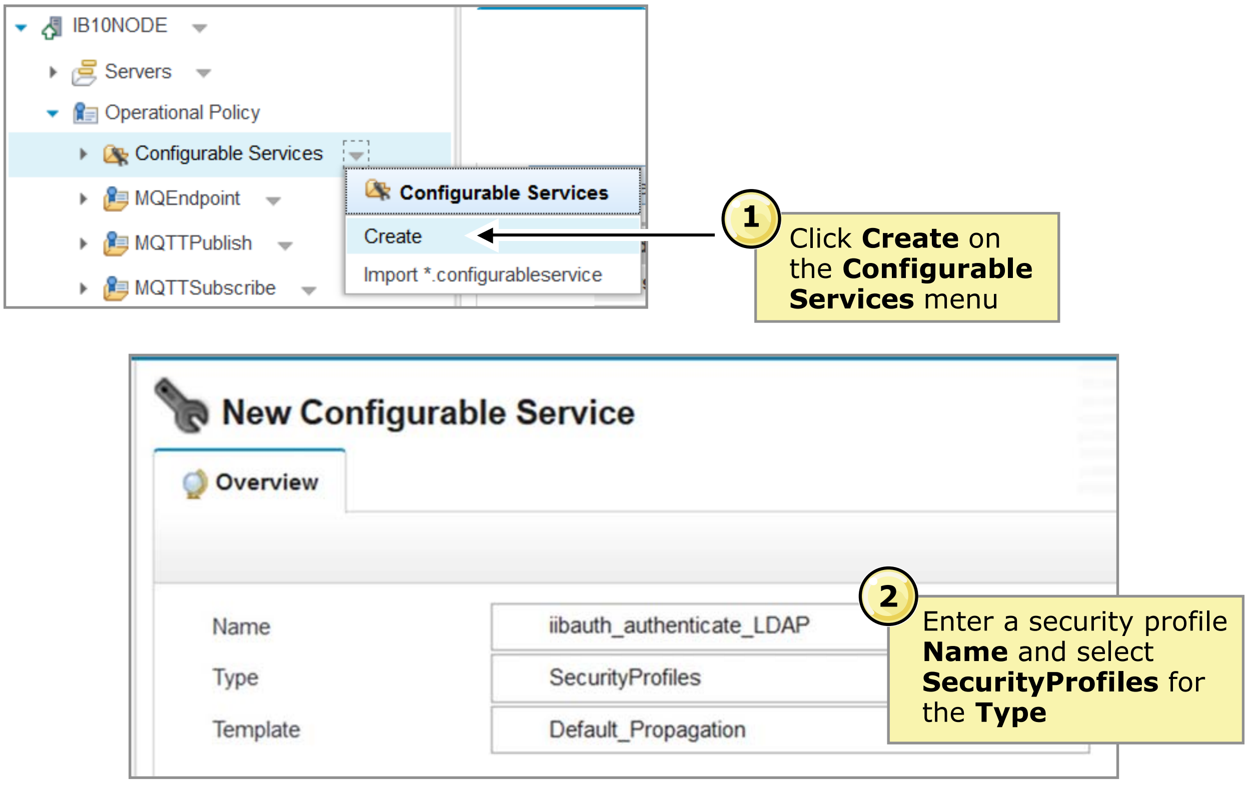
The integration node administrator creates *security profiles* that the security manager accesses at run time.

Security profiles apply to the SecurityPEP node and to security enabled input, output, and request nodes.

The administrator configures these profiles at deployment time in the BAR file editor or by using the `mqsicreateconfigurableservice` command.

WebSphere Education

Creating a security profile (1 of 2)


IB10NODE
Servers
Operational Policy
Configurable Services
MQEndpoint
MQTTPublish
MQTTSubscribe
Configurable Services
Create
Import *.configurableservice
1
Click **Create** on the **Configurable Services** menu
New Configurable Service
Overview
2
Enter a security profile **Name** and select **SecurityProfiles** for the **Type**

Name	iibauth_authenticate_LDAP
Type	SecurityProfiles
Template	Default_Propagation

© Copyright IBM Corporation 2015

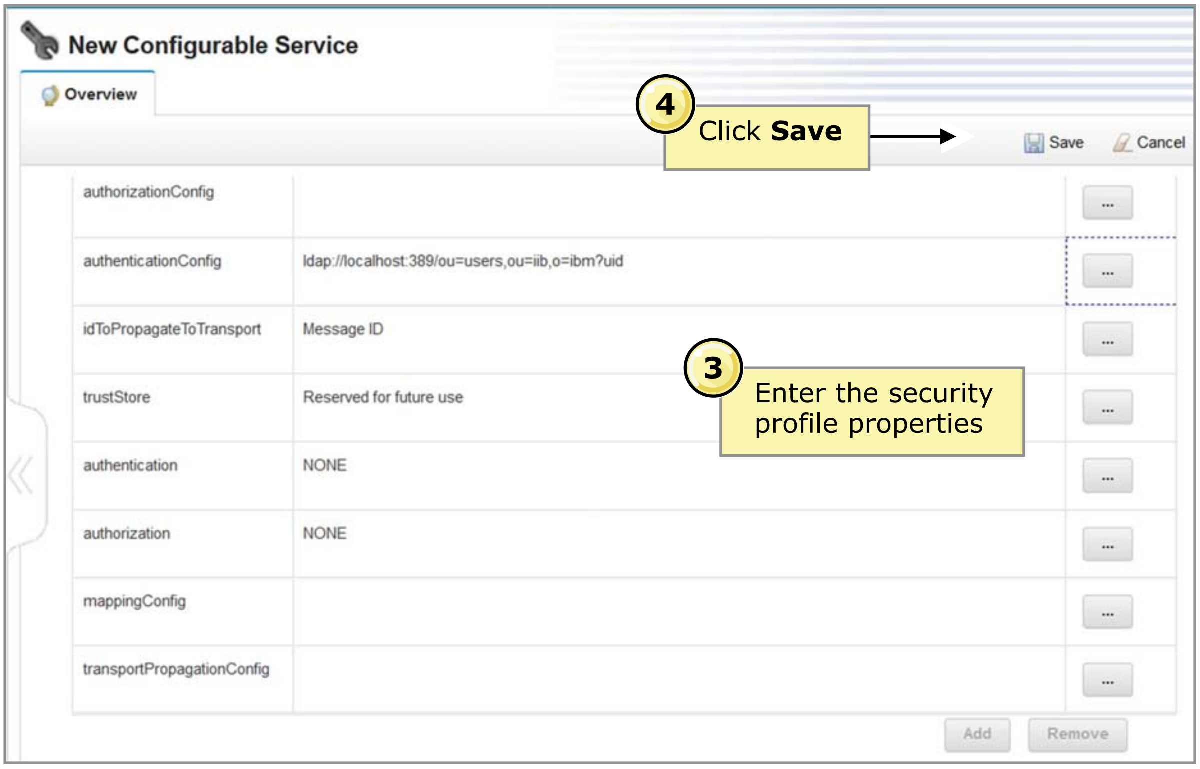
Figure 7-13. Creating a security profile (1 of 2) WM6461.0

Notes:

You can create security profiles in the IBM Integration web user interface.

1. Click **Create** on the **Operational Policy > Configurable Services** menu.
2. On the New Configurable Service page, enter a descriptive name for the security profile, select **SecurityProfiles** as the **Type**, and then select the **Template**.

Creating a security profile (2 of 2)



New Configurable Service

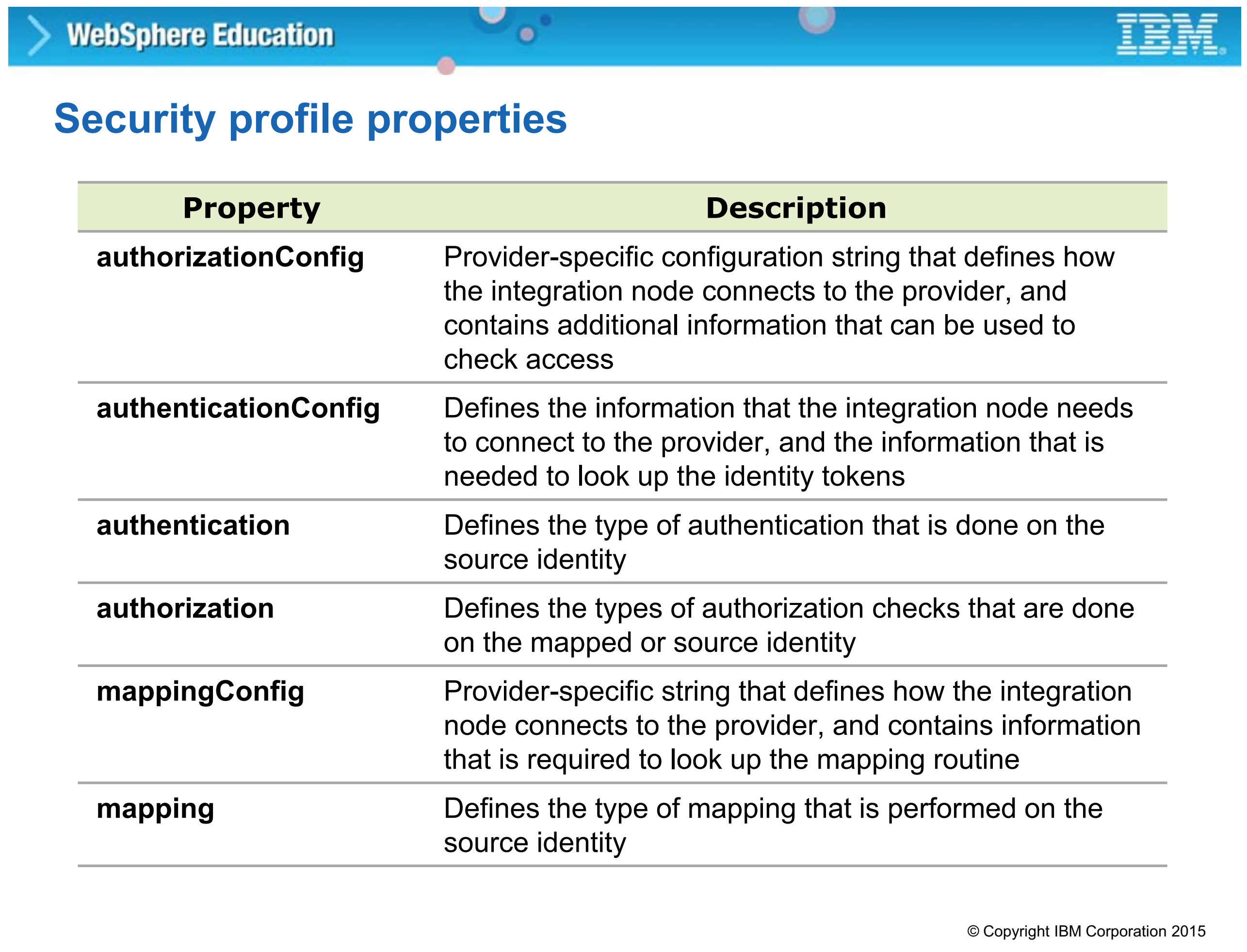
authorizationConfig	
authenticationConfig	ldap://localhost:389/ou=users,ou=iib,o=ibm?uid
idToPropagateToTransport	Message ID
trustStore	Reserved for future use
authentication	NONE
authorization	NONE
mappingConfig	
transportPropagationConfig	

Figure 7-14. Creating a security profile (2 of 2) WM6461.0

Notes:

To continue creating a security profile:

3. Enter the security profile properties.
4. Click **Save** to save the security profile.



WebSphere Education IBM

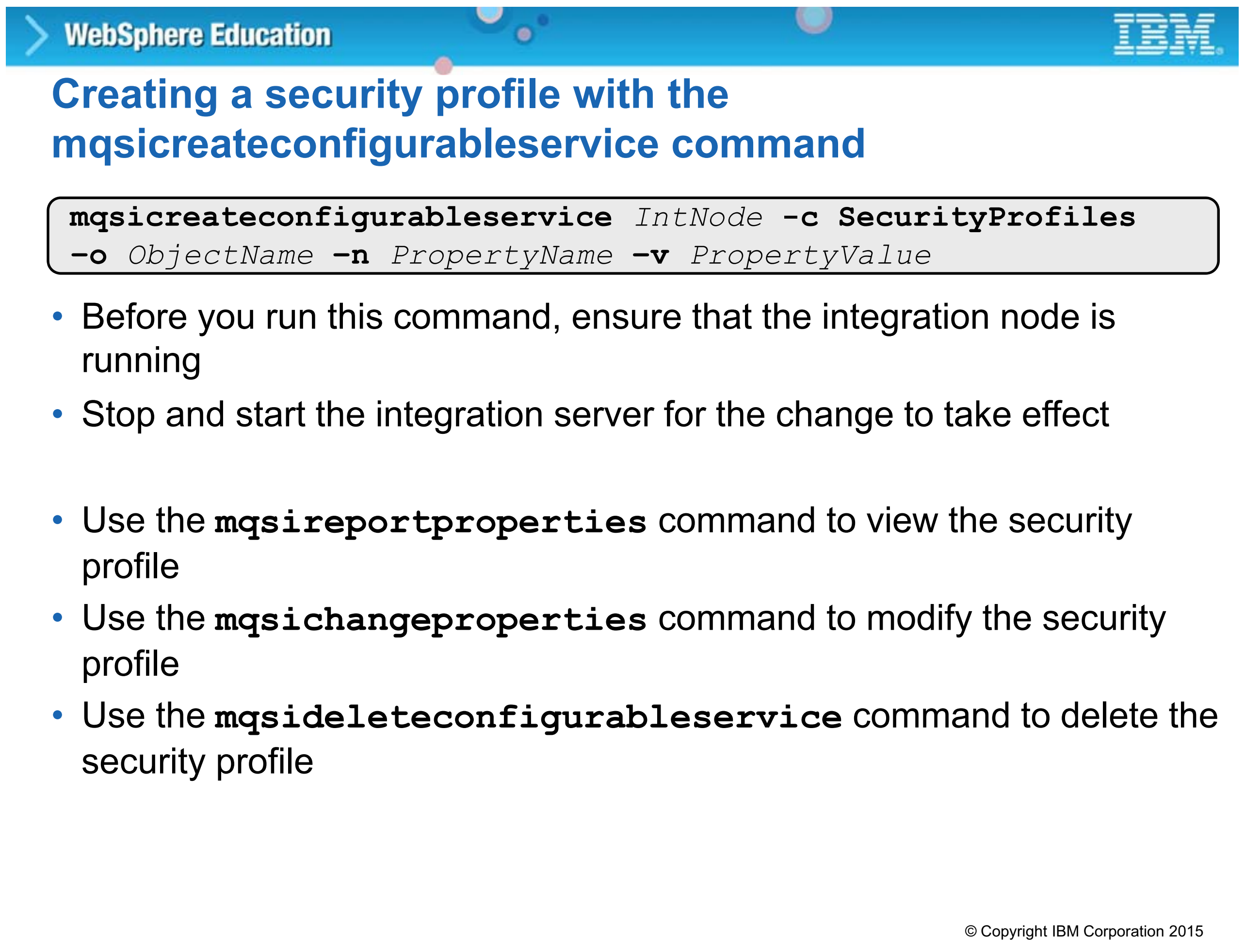
Security profile properties

Property	Description
authorizationConfig	Provider-specific configuration string that defines how the integration node connects to the provider, and contains additional information that can be used to check access
authenticationConfig	Defines the information that the integration node needs to connect to the provider, and the information that is needed to look up the identity tokens
authentication	Defines the type of authentication that is done on the source identity
authorization	Defines the types of authorization checks that are done on the mapped or source identity
mappingConfig	Provider-specific string that defines how the integration node connects to the provider, and contains information that is required to look up the mapping routine
mapping	Defines the type of mapping that is performed on the source identity

Figure 7-15. Security profile properties WM6461.0

Notes:

This table describes some of the security profile properties. For more information, see the IBM Knowledge Center for IBM Integration Bus V10.


WebSphere Education
IBM

Creating a security profile with the mqsicreateconfigurableservice command

```
mqsicreateconfigurableservice IntNode -c SecurityProfiles
-o ObjectName -n PropertyName -v PropertyValue
```

- Before you run this command, ensure that the integration node is running
- Stop and start the integration server for the change to take effect

- Use the **`mqsireportproperties`** command to view the security profile
- Use the **`mqsichangeproperties`** command to modify the security profile
- Use the **`mqsideleteconfigurableservice`** command to delete the security profile

© Copyright IBM Corporation 2015

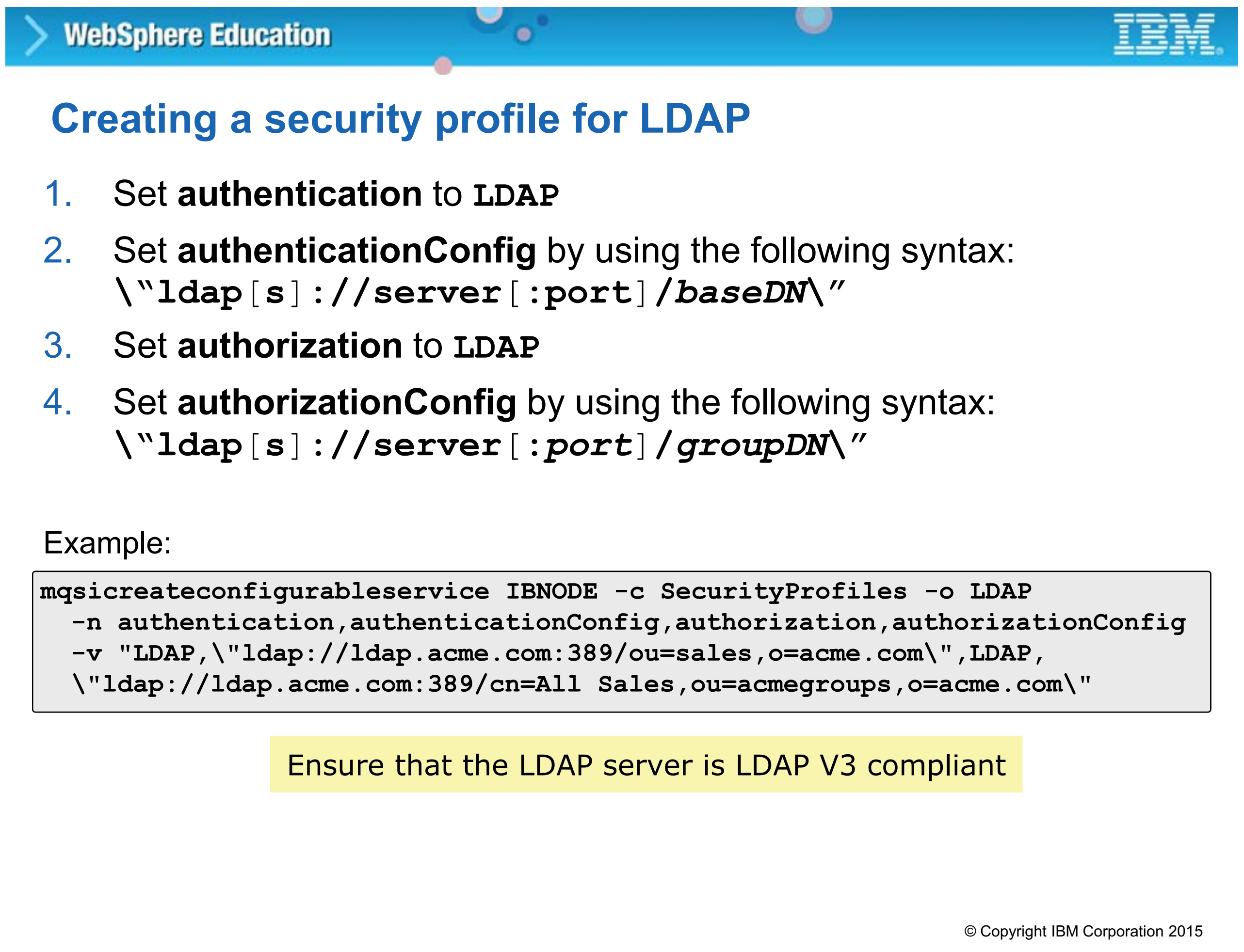
Figure 7-16. Creating a security profile with the mqsicreateconfigurableservice command WM6461.0

Notes:

You can create a security profile by using the `mqsicreateconfigurableservice` command.

The command example in the figure shows the syntax for creating a security profile configurable service. The arguments after `-n` are the property names, which are separated by commas. The arguments after `-v` are the values for each property that is separated by commas.

You can use the IBM Integration web user interface or the `msqsireportproperties` command to view the configurable service.



WebSphere Education

Creating a security profile for LDAP

1. Set **authentication** to **`LDAP`**
2. Set **authenticationConfig** by using the following syntax: **`\"ldap[s]://server[:port]/`*`baseDN`*`\"`**
3. Set **authorization** to **`LDAP`**
4. Set **authorizationConfig** by using the following syntax: **`\"ldap[s]://server[:`*`port`*`]/`*`groupDN`*`\"`**

Example:

```
mqsicreateconfigurableservice IBNODE -c SecurityProfiles -o LDAP
  -n authentication,authenticationConfig,authorization,authorizationConfig
  -v "LDAP,\"ldap://ldap.acme.com:389/ou=sales,o=acme.com\",LDAP,
  \"ldap://ldap.acme.com:389/cn=All Sales,ou=acmegroups,o=acme.com\"
```

Ensure that the LDAP server is LDAP V3 compliant

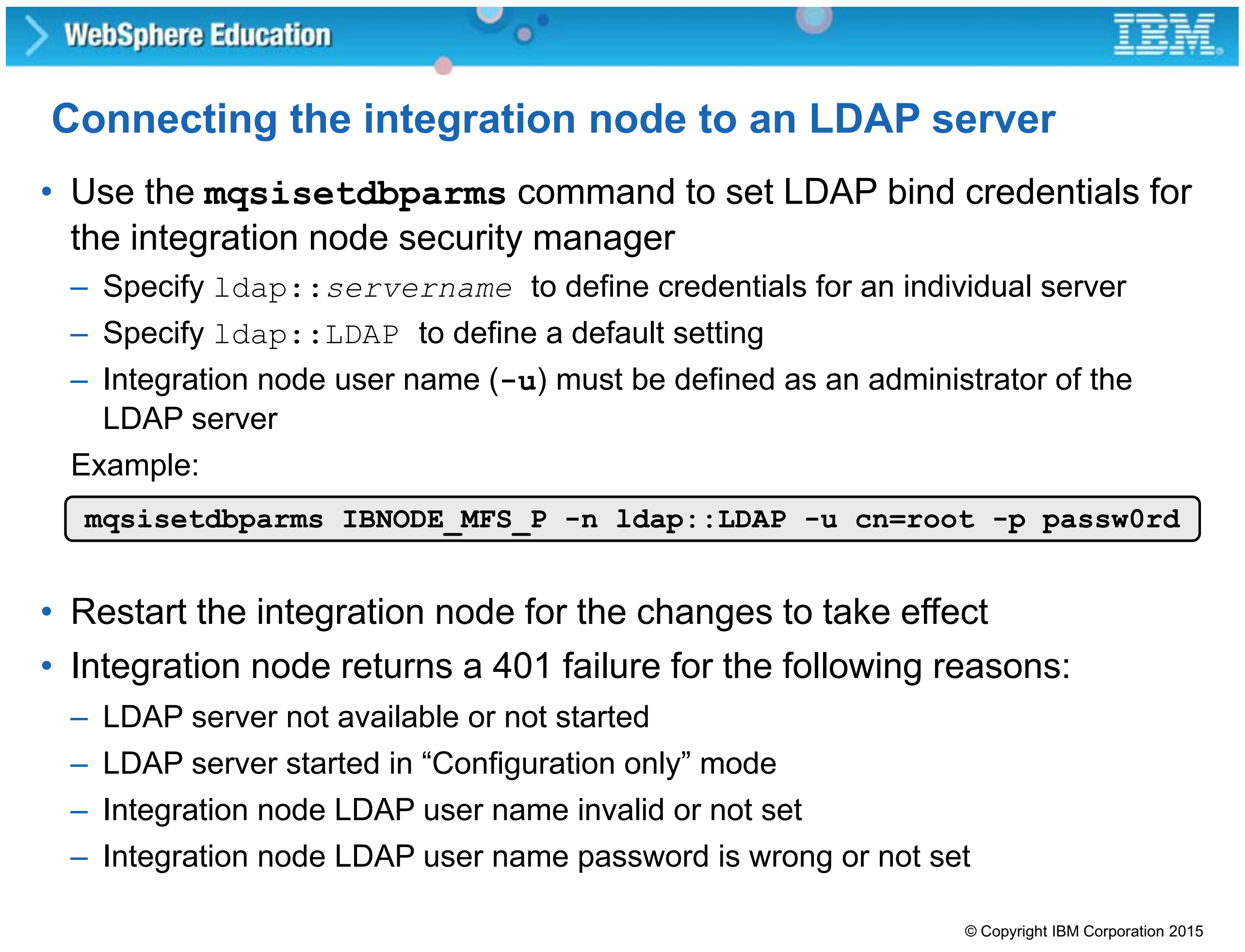
Figure 7-17. Creating a security profile for LDAP WM6461.0

Notes:

This example shows how to create a security profile for LDAP. It sets the `authentication` property to `LDAP`, the `authenticationConfig` property to `ldap://ldap.acme.com:389/ou=sales, o=acme.com\`

Note

You must enclose the LDAP URL (which contains commas) with escaped double quotation marks (\") so that the URL commas are not confused with the comma separator of the value parameter of `mqsicreateconfigurableservice` command.



Connecting the integration node to an LDAP server

- Use the **`mqsisetdbparms`** command to set LDAP bind credentials for the integration node security manager
  - Specify `ldap::`*`servername`* to define credentials for an individual server
  - Specify `ldap::LDAP` to define a default setting
  - Integration node user name (**`-u`**) must be defined as an administrator of the LDAP server

  Example:

```
mqsisetdbparms IBNODE_MFS_P -n ldap::LDAP -u cn=root -p passw0rd
```

- Restart the integration node for the changes to take effect
- Integration node returns a 401 failure for the following reasons:
  - LDAP server not available or not started
  - LDAP server started in “Configuration only” mode
  - Integration node LDAP user name invalid or not set
  - Integration node LDAP user name password is wrong or not set

Figure 7-18. Connecting the integration node to an LDAP server WM6461.0

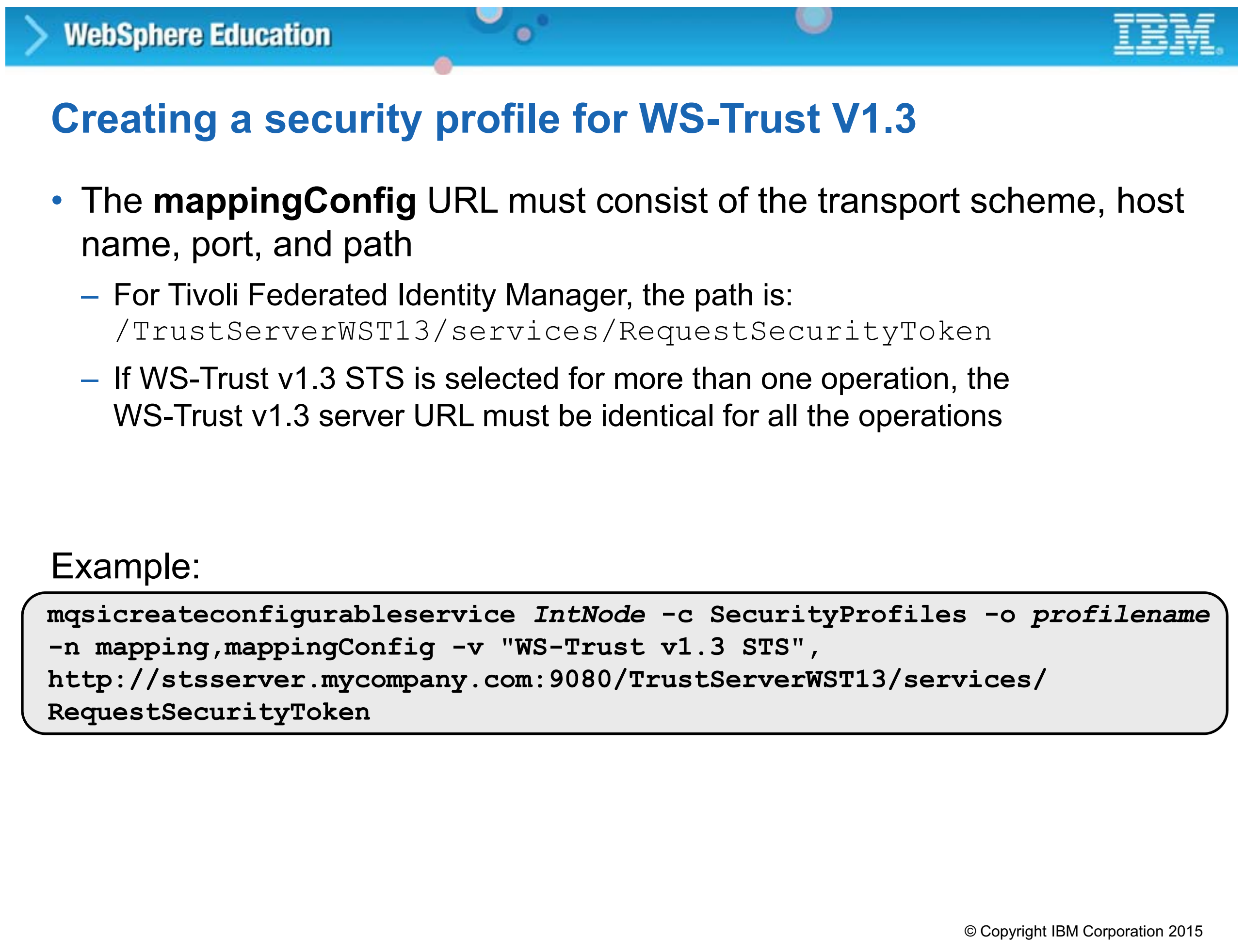
Notes:

If you are using LDAP for authentication or authorization, you must provide the integration node with connection information to the LDAP server by using the **`msqisetdbparms`** command.

Specify `ldap::`*servername* to define credentials for an individual server.

If you want the integration node to bind anonymously to this server, specify anonymous as the user ID.

Specify `ldap::`LDAP to define a default setting. The integration node uses the specified user ID and password values for all servers that do not have an explicit `ldap::`*servername* entry. So all servers that previously used anonymous bind by default start to use the details that are defined in an `ldap::`LDAP entry.



WebSphere Education
IBM

Creating a security profile for WS-Trust V1.3

- The **mappingConfig** URL must consist of the transport scheme, host name, port, and path
  - For Tivoli Federated Identity Manager, the path is: `/TrustServerWST13/services/RequestSecurityToken`
  - If WS-Trust v1.3 STS is selected for more than one operation, the WS-Trust v1.3 server URL must be identical for all the operations

Example:

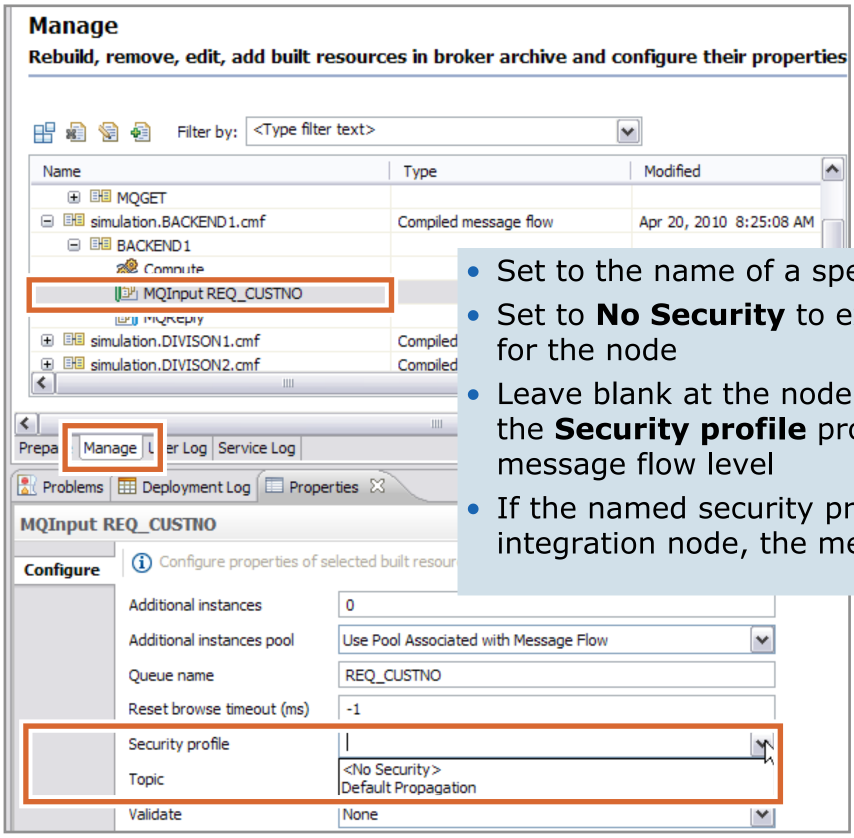
```
mqsicreateconfigurableservice IntNode -c SecurityProfiles -o profilename
-n mapping,mappingConfig -v "WS-Trust v1.3 STS",
http://stsserver.mycompany.com:9080/TrustServerWST13/services/
RequestSecurityToken
```

Figure 7-19. Creating a security profile for WS-Trust V1.3 WM6461.0

Notes:

This figure provides an example of creating a security profile for a WS-Trust V1.3 external security provider.

Setting the Security profile property on the BAR file


Manage

Rebuild, remove, edit, add built resources in broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
MQGET		
simulation.BACKEND1.cmf	Compiled message flow	Apr 20, 2010 8:25:08 AM
BACKEND1		
Compute		
MQInput REQ_CUSTNO		
MQReply		
simulation.DIVISON1.cmf	Compiled	
simulation.DIVISON2.cmf	Compiled	

MQInput REQ_CUSTNO

Configure properties of selected built resour

Additional instances	0
Additional instances pool	Use Pool Associated with Message Flow
Queue name	REQ_CUSTNO
Reset browse timeout (ms)	-1
Security profile	
Topic	
Validate	None

<No Security>
Default Propagation

- Set to the name of a specific security profile
- Set to **No Security** to explicitly turn off security for the node
- Leave blank at the node level if the node inherits the **Security profile** property that is set at the message flow level
- If the named security profile does not exist on the integration node, the message flow fails to deploy

Figure 7-20. Setting the Security profile property on the BAR file WM6461.0

Notes:

Nodes that support runtime security has a **Security profile** property.

- Set the **Security profile** property to **<No Security>** to explicitly turn off security for the node.
- If you leave the **Security profile** property blank, the node inherits the security profile property that is set at the message flow level. If you leave the **Security profile** property blank at both node and message flow levels, security is turned off for the node.
- When the **Security profile** property is set to the name of a specific security profile, that profile determines the message flow security configuration. If the named security profile does not exist in the runtime environment, the message flow fails to deploy.
- If you want to extract and propagate an identity without security enforcement or mapping, you can use the supplied security profile that is called **Default Propagation**.

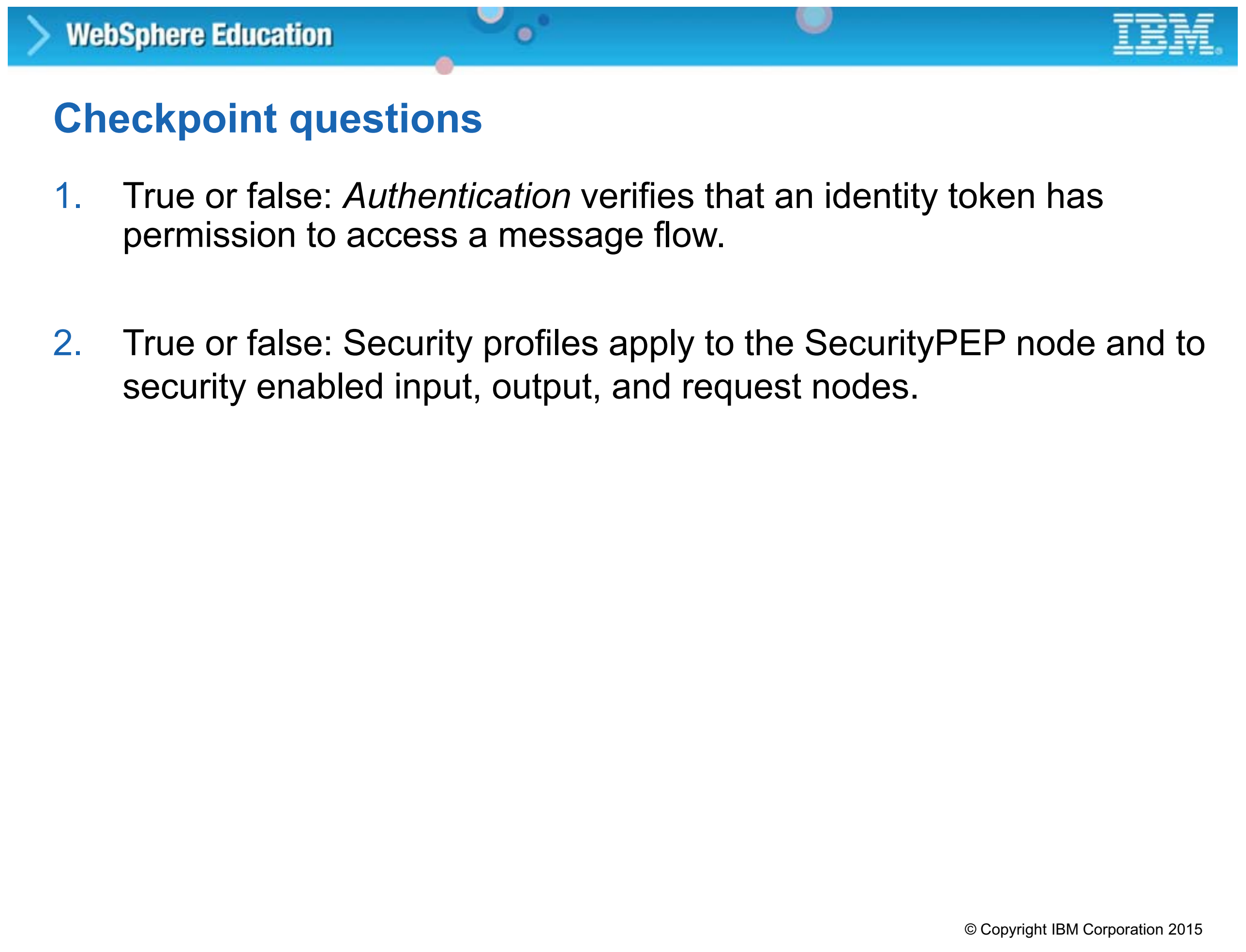
Unit summary

Having completed this unit, you should be able to:

- Describe message flow security support in IBM Integration Bus
- Define security profiles for authorization groups
- Use the IBM Integration Toolkit to associate a security profile with a message flow

Figure 7-21. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. True or false: *Authentication* verifies that an identity token has permission to access a message flow.

2. True or false: Security profiles apply to the SecurityPEP node and to security enabled input, output, and request nodes.

Figure 7-22. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.

WebSphere Education

Checkpoint answers

1. True or false: *Authentication* verifies that an identity token has permission to access a message flow.
   Answer: False. *Authentication* establishes the identity of a user or system and verifies that the identity is valid. *Authorization* verifies that an identity token has permission to access a message flow.

2. True or false: Security profiles apply to the SecurityPEP node and to security enabled input, output, and request nodes.
   Answer: True

Figure 7-23. Checkpoint answers WM6461.0

Notes:

Unit 8. Administering web services and web service security

What this unit is about

In this unit, you learn how to administer message flows that use HTTP and SOAP to provide web services. The unit also describes how to implement HTTPS and support Web Services Security (WS-Security).

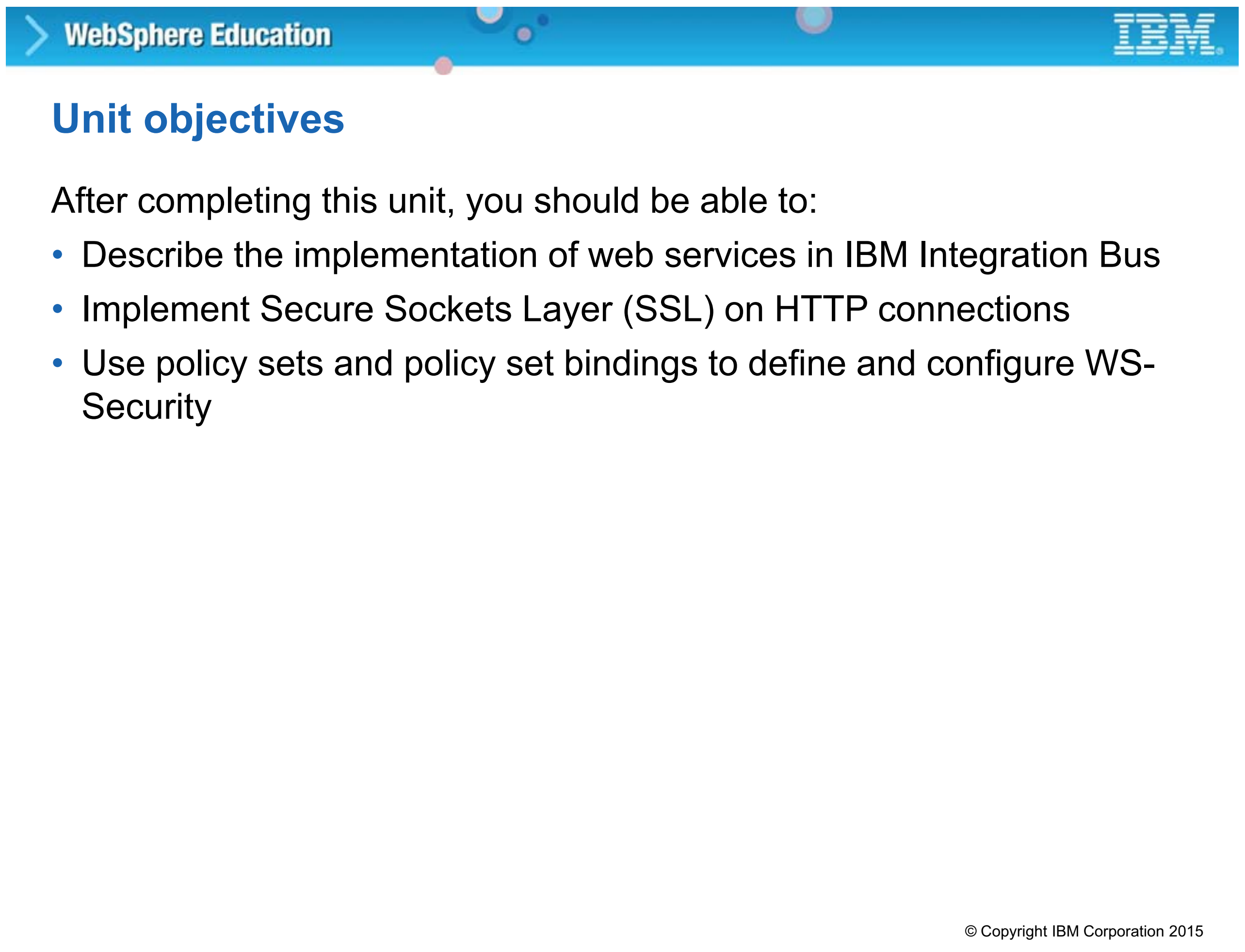
What you should be able to do

After completing this unit, you should be able to:

- Describe the implementation of web services in IBM Integration Bus
- Implement Secure Sockets Layer (SSL) on HTTP connections
- Use policy sets and policy set bindings to define and configure WS-Security

How you will check your progress

- Checkpoint questions
- Exercise: Implementing web services and web services security



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Describe the implementation of web services in IBM Integration Bus
- Implement Secure Sockets Layer (SSL) on HTTP connections
- Use policy sets and policy set bindings to define and configure WS-Security

© Copyright IBM Corporation 2015

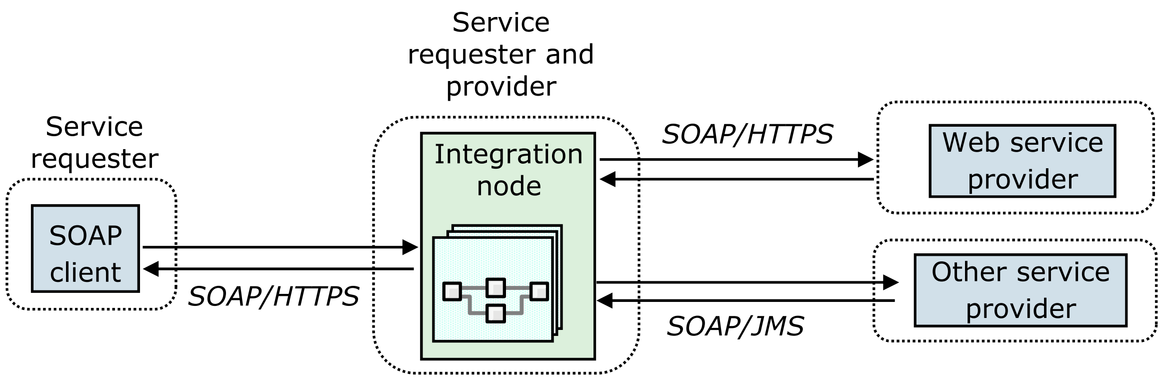
Figure 8-1. Unit objectives WM6461.0

Notes:

8.1. IBM Integration Bus in a web services environment

WebSphere Education IBM.

Integration Bus web services overview



- Integration Bus can:
  - Act as a front end to many service providers to allow for failover, which provides higher overall quality of service
  - Log messages for auditing and nonrepudiation
  - Map SOAP messages to other industry standard formats
  - Select service provider based on message context and content

Figure 8-2. Integration Bus web services overview WM6461.0

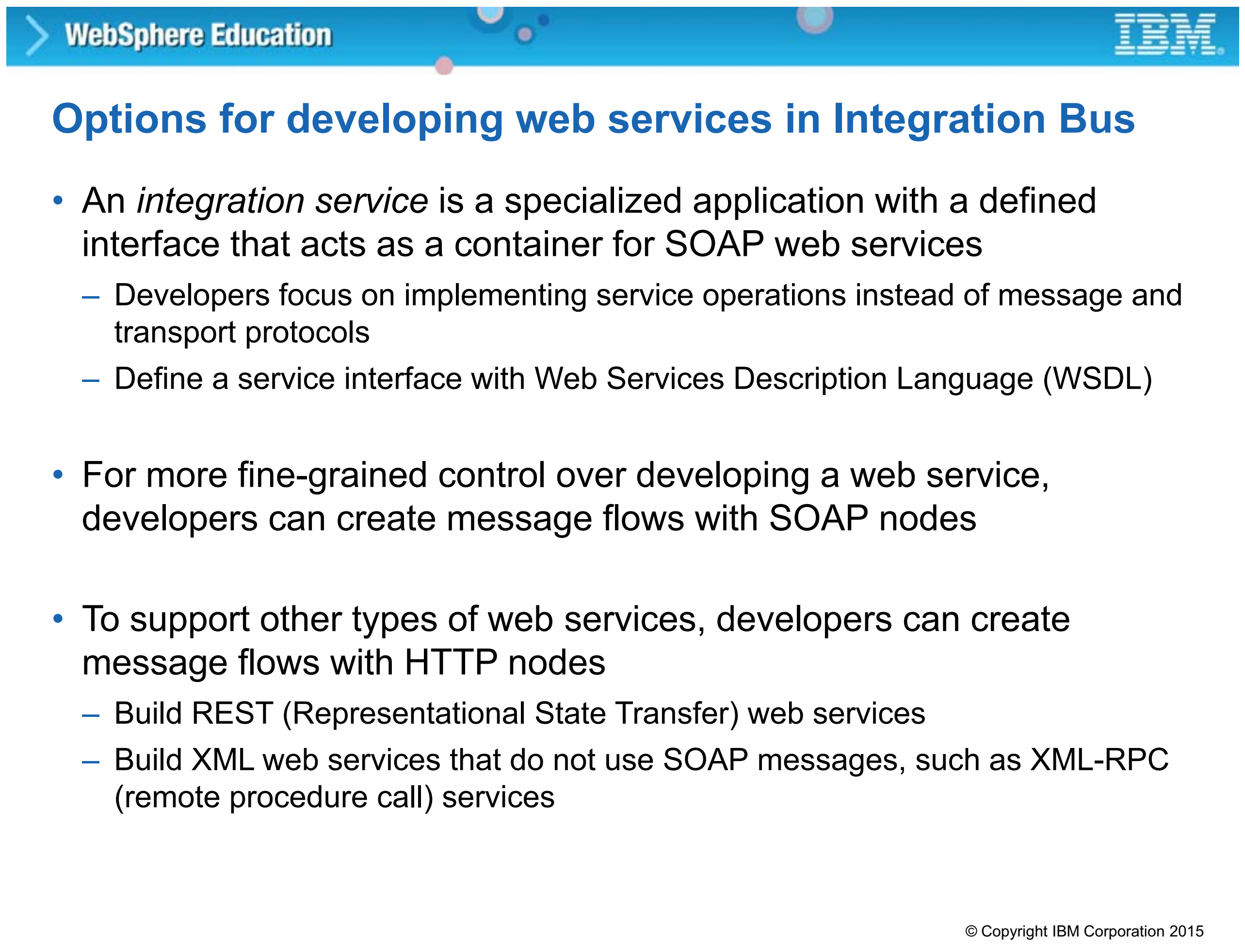
Notes:

Integration Bus is a convenient central point for web services brokering. You can wrap existing services or access web services from existing applications. The integration node is shown in the center of the diagram in the figure because it can serve as the interface between a client and a service provider.

Integration Bus can act as a SOAP intermediary, providing first point of contact functions like hiding or changing service implementation, transformation between WSDL definitions, monitoring, data warehousing, and auditing. Normal broker strengths are applied to and enhance web services.

A message flow application can be the requester or a client. Usually, an MQ Input node drives a flow that calls a web service (over HTTP or JMS bindings), the result of which is used to augment the message that is returned with an MQ Output node.

A message flow application can also be the service provider, allowing web service clients to call an existing message flow application.



Options for developing web services in Integration Bus

- An *integration service* is a specialized application with a defined interface that acts as a container for SOAP web services
  - Developers focus on implementing service operations instead of message and transport protocols
  - Define a service interface with Web Services Description Language (WSDL)
- For more fine-grained control over developing a web service, developers can create message flows with SOAP nodes
- To support other types of web services, developers can create message flows with HTTP nodes
  - Build REST (Representational State Transfer) web services
  - Build XML web services that do not use SOAP messages, such as XML-RPC (remote procedure call) services

Figure 8-3. Options for developing web services in Integration Bus WM6461.0

Notes:

The IBM Integration Toolkit supports three options to developing web services.

An *integration service* is a specialized application that acts as a container for SOAP web services that are defined with a WSDL document. Integration services provide a structured solution for quickly building a web service. Integration services provide a structured way to implement WSDL web service operations as subflows. The integration service container manages the SOAP message parsing and HTTP message transport options. With this approach, developers focus on implementing business logic, not managing the lower-level message serialization and de-serialization tasks.

For finer level of control in processing SOAP web services, developers can build message flows with SOAP nodes. This approach allows developers to create a SOAP web service that non-HTTP transport protocols, such as JMS.

Developers can build any type of web service with HTTP nodes. For example, RESTful web services and XML-RPC web services do not conform to the SOAP specification. The integration service container and SOAP nodes do not support these web service types.

WebSphere Education

Message flow as an HTTP/HTTPS client

- Clients that access web services
- Integration Bus is buffer between changing set of service providers (web service and other)
- Aggregation of web services

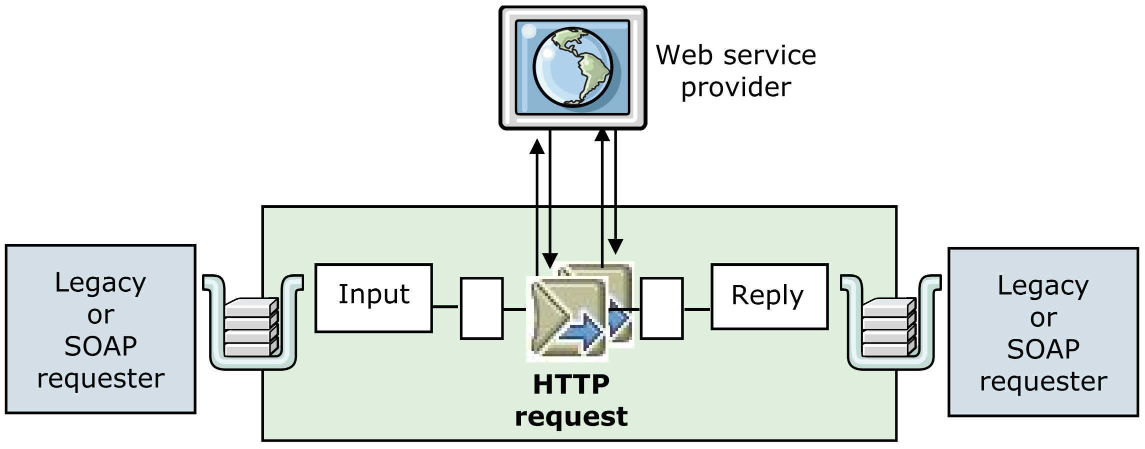


Figure 8-4. Message flow as an HTTP/HTTPS client WM6461.0

Notes:

The message flow in the figure uses the HTTP Request node to interact with a web service by using all or part of the input message as the request sent to that service. The node can also be configured to create an output message from the contents of the input message before the message is propagated to subsequent nodes in the message flow. The contents of the web service response augments the response.

Depending on the configuration, the HTTP Request node constructs an HTTP or an HTTP over SSL (HTTPS) request from the specified contents of the input message. The node sends the request to the web service and then receives the response from the web service and parses the response for inclusion in the output tree. The node generates HTTP headers if the configuration requires them.

The **Default Web service URL** property on the HTTP Request node determines the destination URL for a web service request.

WebSphere Education IBM.

Message flow as an HTTP/HTTPS service provider

- Integration Bus acts as an intermediary to the web service to provide routing
- Many more combinations

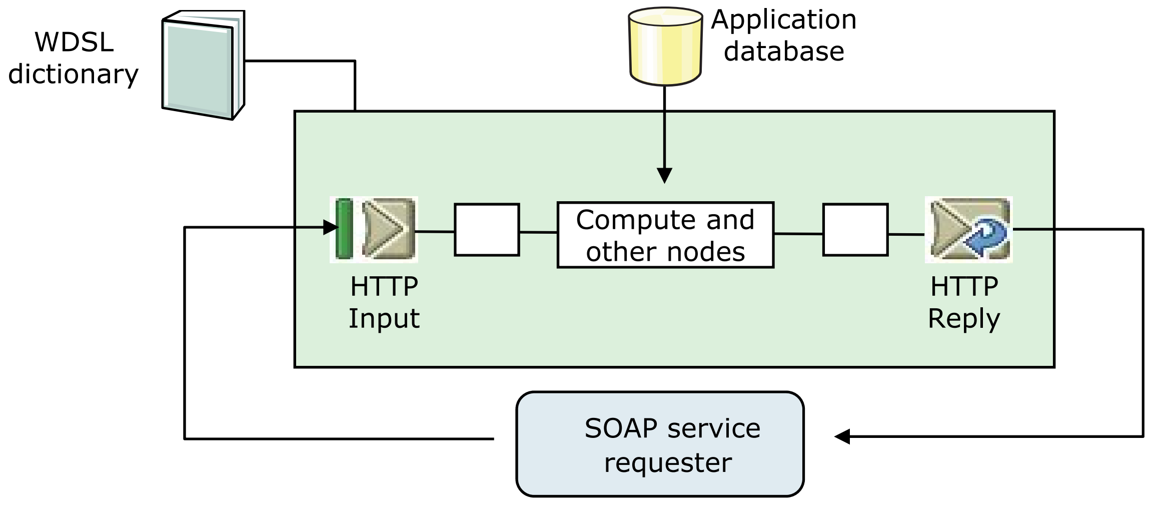


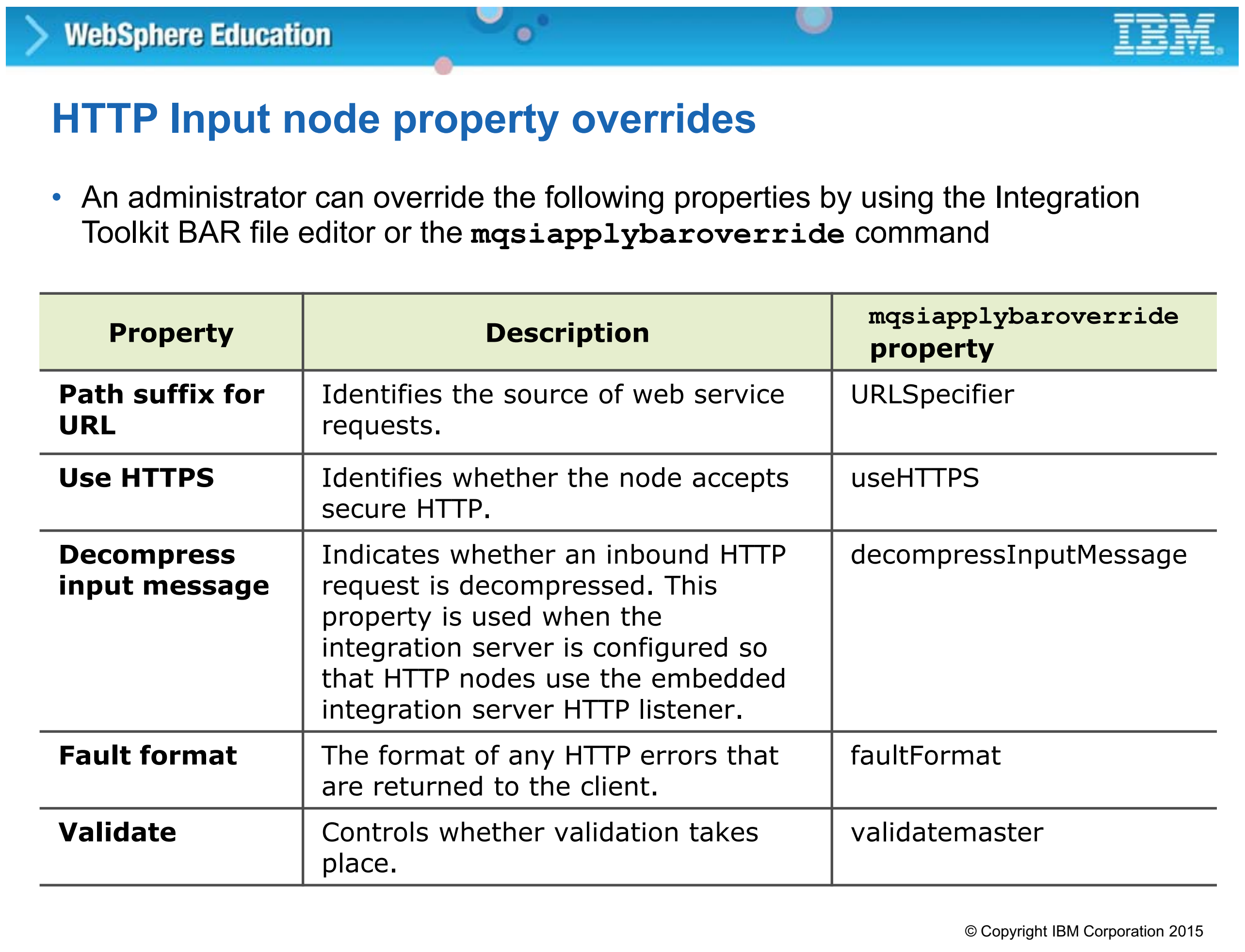
Figure 8-5. Message flow as an HTTP/HTTPS service provider WM6461.0

Notes:

Each integration node has a single TCP/IP port on which incoming HTTP requests are accepted. Client applications can post to a predefined URL or a URL that is already looked up. An attribute on each HTTP Input node qualifies the requests for which the node is responsible.

A message flow application can act as an intermediary and transform, route, or aggregate web service requests. A message flow application parses the request, and forwards it to one or more destinations according to the content, perhaps after transformation. Reply data from the destinations is returned to the original client.

For example, a message flow application can transform HTTP to IBM MQ. It parses the incoming HTTP request, and forwards the transformed request onto an IBM MQ application that does not handle HTTP. The reply is then transformed into the client format and returned.



HTTP Input node property overrides

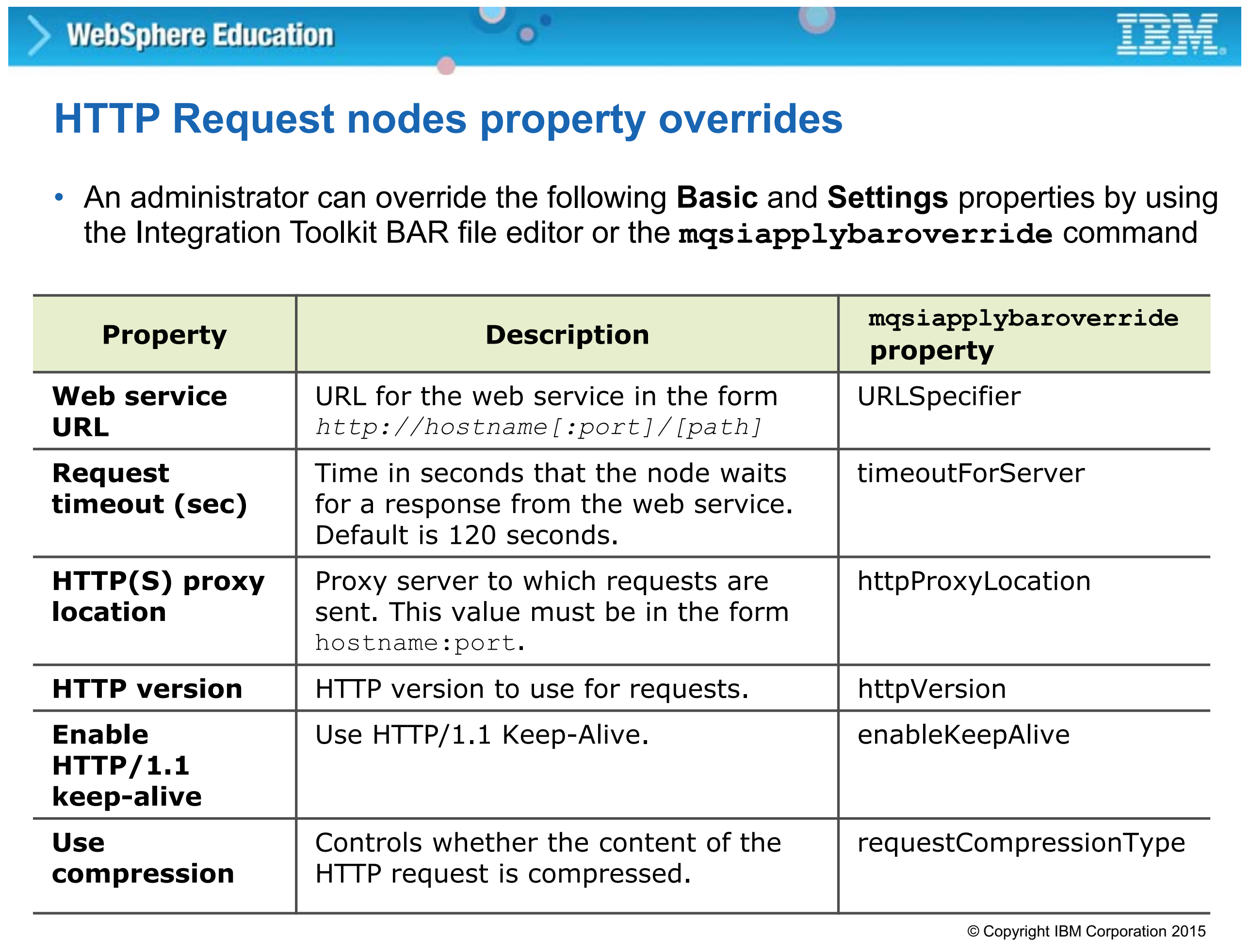
- An administrator can override the following properties by using the Integration Toolkit BAR file editor or the **`mqsiapplybaroverride`** command

Property	Description	**`mqsiapplybaroverride`** property
Path suffix for URL	Identifies the source of web service requests.	URLSpecifier
Use HTTPS	Identifies whether the node accepts secure HTTP.	useHTTPS
Decompress input message	Indicates whether an inbound HTTP request is decompressed. This property is used when the integration server is configured so that HTTP nodes use the embedded integration server HTTP listener.	decompressInputMessage
Fault format	The format of any HTTP errors that are returned to the client.	faultFormat
Validate	Controls whether validation takes place.	validatemaster

Figure 8-6. HTTP Input node property overrides WM6461.0

Notes:

As an administrator, only a few properties of an HTTP Input node can be altered in a BAR file before deploying or by using the `mqsiapplybaroverride` command. The figure lists those properties that the administrator can override.



WebSphere Education

HTTP Request nodes property overrides

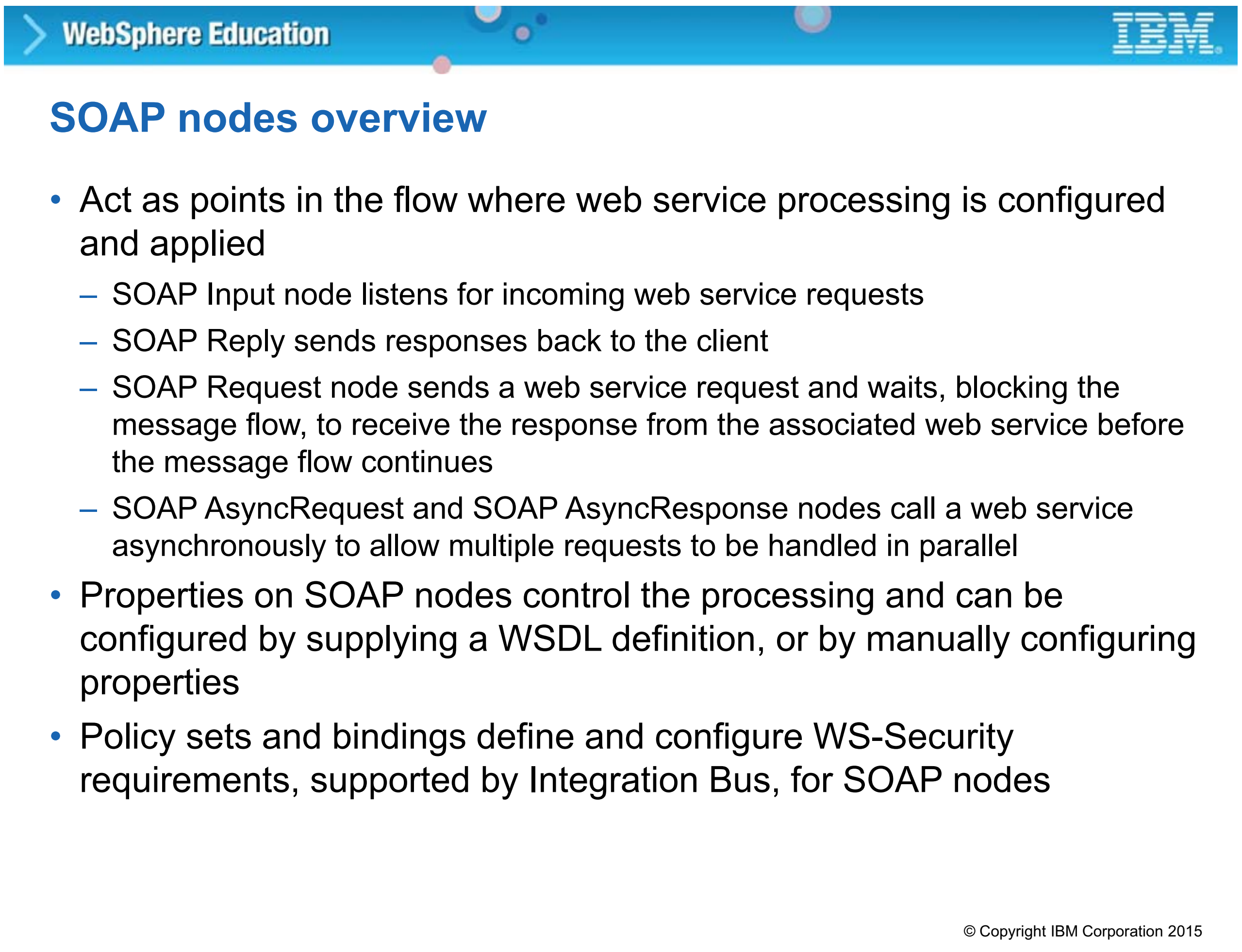
- An administrator can override the following **Basic** and **Settings** properties by using the Integration Toolkit BAR file editor or the **`mqsiapplybaroverride`** command

Property	Description	`mqsiapplybaroverride` property
Web service URL	URL for the web service in the form *`http://hostname[:port]/[path]`*	URLSpecifier
Request timeout (sec)	Time in seconds that the node waits for a response from the web service. Default is 120 seconds.	timeoutForServer
HTTP(S) proxy location	Proxy server to which requests are sent. This value must be in the form `hostname:port`.	httpProxyLocation
HTTP version	HTTP version to use for requests.	httpVersion
Enable HTTP/1.1 keep-alive	Use HTTP/1.1 Keep-Alive.	enableKeepAlive
Use compression	Controls whether the content of the HTTP request is compressed.	requestCompressionType

Figure 8-7. HTTP Request nodes property overrides WM6461.0

Notes:

The figure lists the HTTP Request node properties that an administrator can override in the BAR file or with the `mqsiapplybaroverride` command.



WebSphere Education IBM

SOAP nodes overview

- Act as points in the flow where web service processing is configured and applied
  – SOAP Input node listens for incoming web service requests
  – SOAP Reply sends responses back to the client
  – SOAP Request node sends a web service request and waits, blocking the message flow, to receive the response from the associated web service before the message flow continues
  – SOAP AsyncRequest and SOAP AsyncResponse nodes call a web service asynchronously to allow multiple requests to be handled in parallel
- Properties on SOAP nodes control the processing and can be configured by supplying a WSDL definition, or by manually configuring properties
- Policy sets and bindings define and configure WS-Security requirements, supported by Integration Bus, for SOAP nodes

© Copyright IBM Corporation 2015

Figure 8-8. SOAP nodes overview WM6461.0

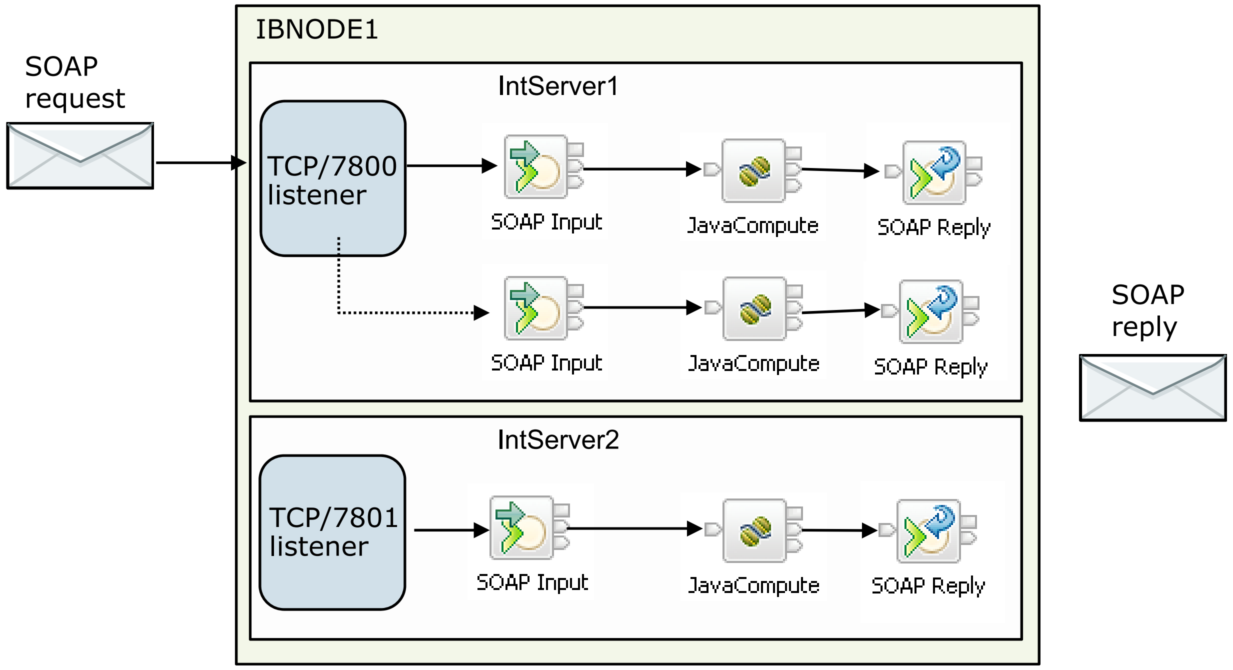
Notes:

HTTP and SOAP nodes can both be used to interact with web services. Typically, SOAP nodes are used when working with SOAP-based web services. The figure describes the SOAP nodes available for sending web service requests and receiving web service responses.

Calling a web service asynchronously means that the SOAP Async Request node sends a web service request. The request does not block the message flow by waiting for the associated web service response because the response is received at the SOAP Async Response node, which is in a separate flow. Therefore, multiple requests can be handled in parallel.

WebSphere Education

SOAP Input and SOAP Reply nodes example

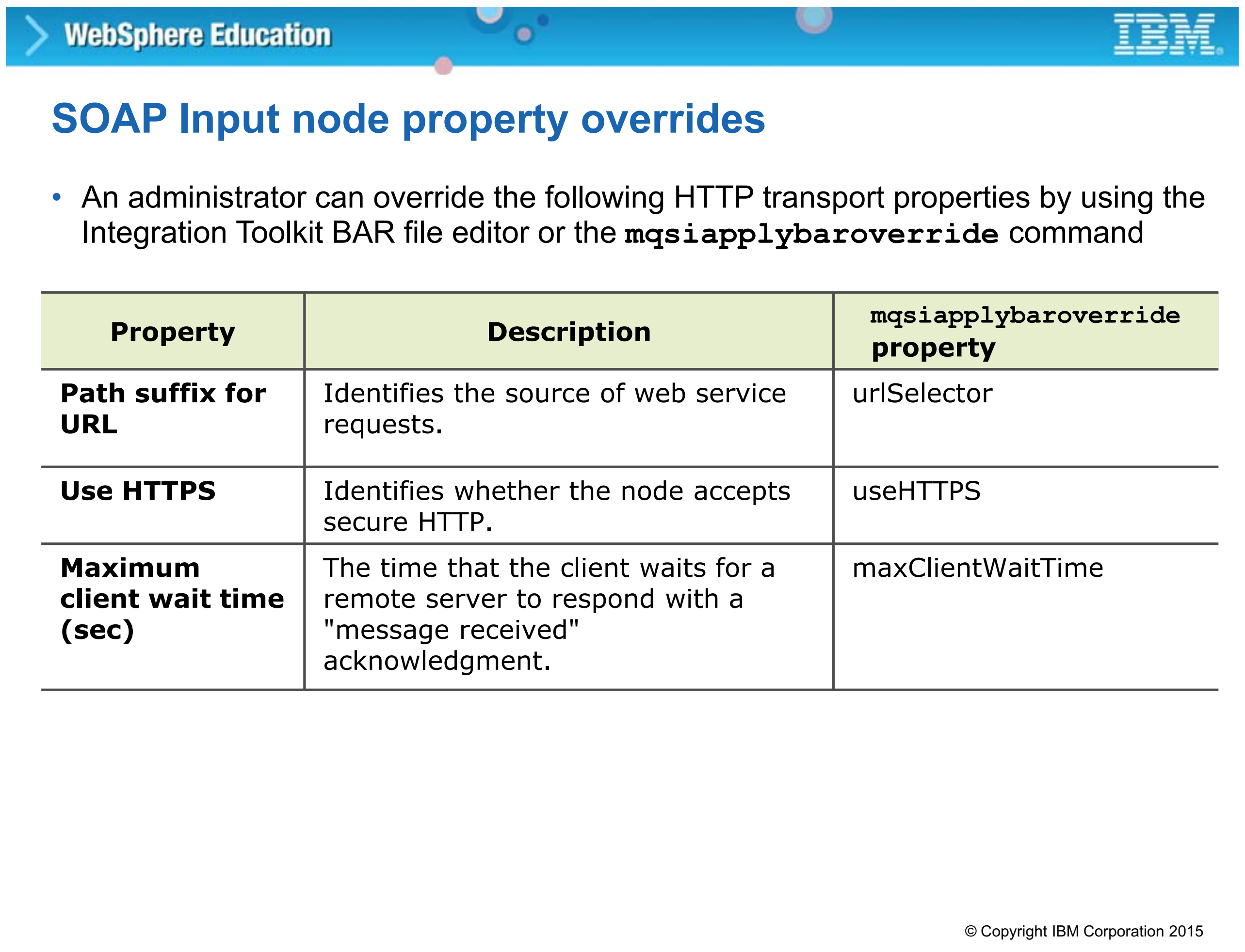


- Every integration server that contains a SOAP Input node is allocated a TCP/IP port; allows incoming HTTP requests to be accepted
- Default port range is configured at the integration node level

Figure 8-9. SOAP Input and SOAP Reply nodes example WM6461.0

Notes:

This figure shows an example of message flows that use SOAP Input nodes and SOAP Reply nodes. In this example, the integration server listener handles the SOAP requests. Each integration server on an integration node must use a unique listener port.


WebSphere Education
IBM

SOAP Input node property overrides

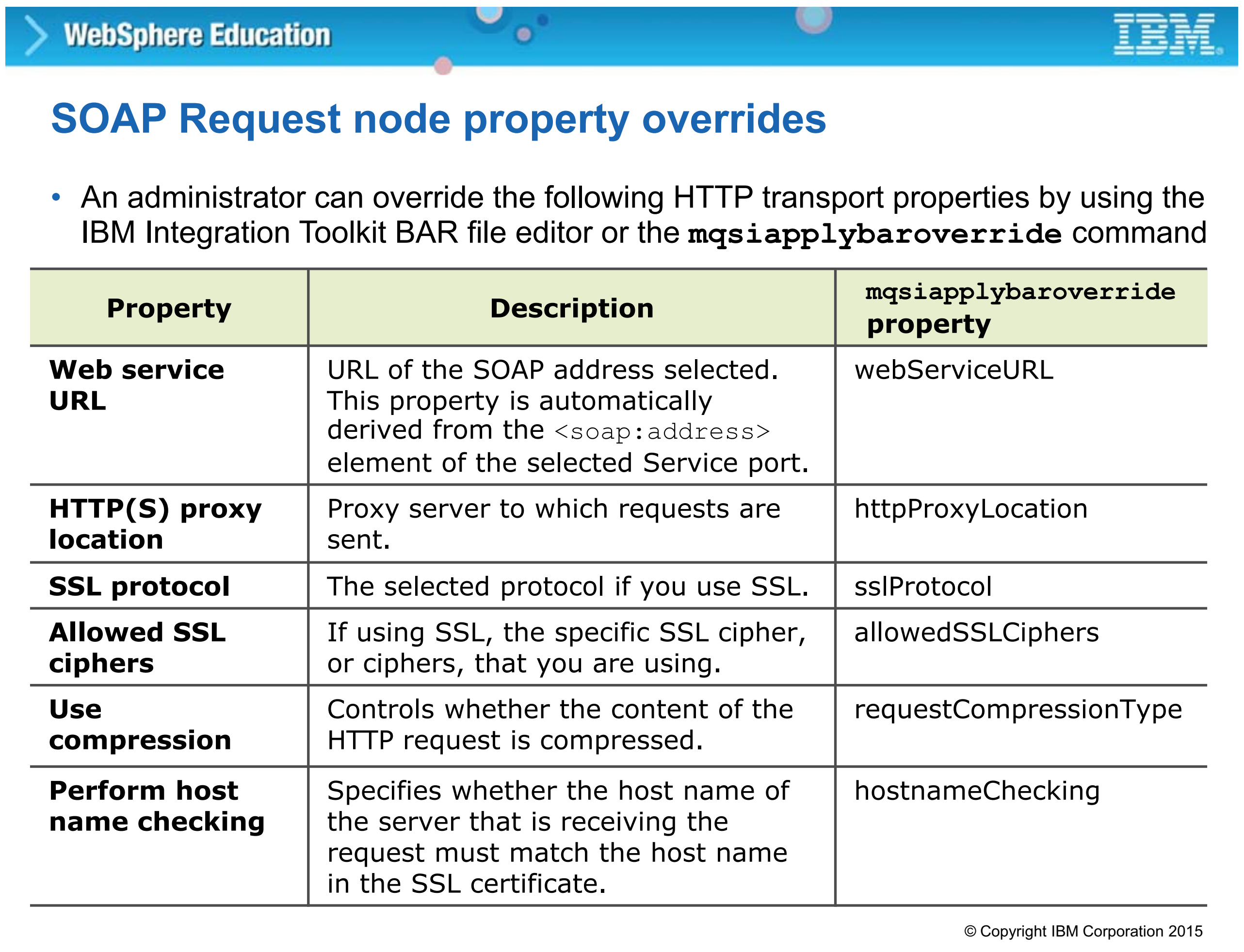
- An administrator can override the following HTTP transport properties by using the Integration Toolkit BAR file editor or the **`mqsiapplybaroverride`** command

Property	Description	**`mqsiapplybaroverride`** property
Path suffix for URL	Identifies the source of web service requests.	urlSelector
Use HTTPS	Identifies whether the node accepts secure HTTP.	useHTTPS
Maximum client wait time (sec)	The time that the client waits for a remote server to respond with a "message received" acknowledgment.	maxClientWaitTime

Figure 8-10. SOAP Input node property overrides WM6461.0

Notes:

The figure lists the SOAP Input node properties that an administrator can override in the BAR file or with the `mqsiapplybaroverride` command.



SOAP Request node property overrides

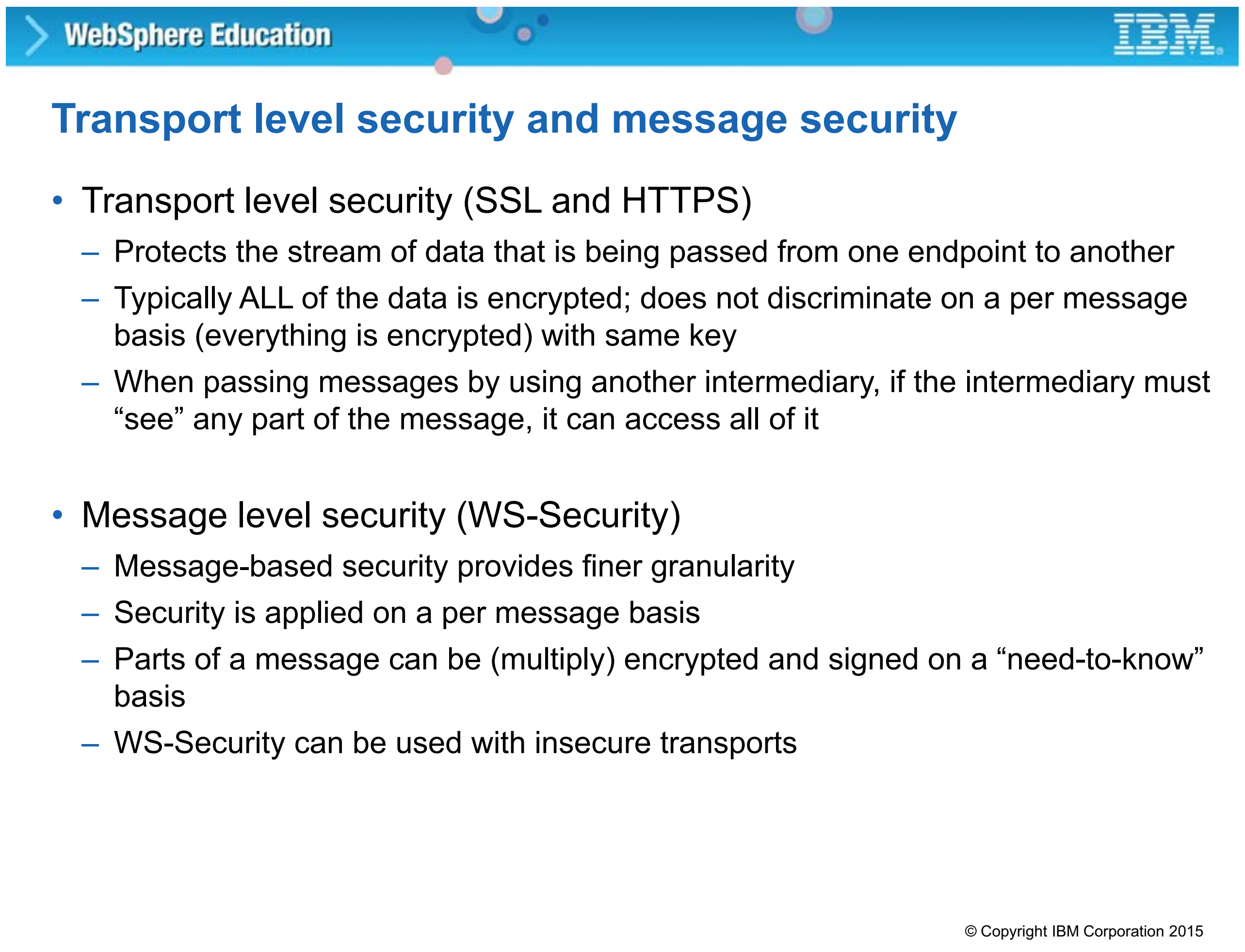
- An administrator can override the following HTTP transport properties by using the IBM Integration Toolkit BAR file editor or the **`mqsiapplybaroverride`** command

Property	Description	`mqsiapplybaroverride` property
Web service URL	URL of the SOAP address selected. This property is automatically derived from the `<soap:address>` element of the selected Service port.	webServiceURL
HTTP(S) proxy location	Proxy server to which requests are sent.	httpProxyLocation
SSL protocol	The selected protocol if you use SSL.	sslProtocol
Allowed SSL ciphers	If using SSL, the specific SSL cipher, or ciphers, that you are using.	allowedSSLCiphers
Use compression	Controls whether the content of the HTTP request is compressed.	requestCompressionType
Perform host name checking	Specifies whether the host name of the server that is receiving the request must match the host name in the SSL certificate.	hostnameChecking

Figure 8-11. SOAP Request node property overrides WM6461.0

Notes:

The figure lists the SOAP Request node properties that an administrator can override in the BAR file or with the `mqsiapplybaroverride` command.



Transport level security and message security

- Transport level security (SSL and HTTPS)
  - Protects the stream of data that is being passed from one endpoint to another
  - Typically ALL of the data is encrypted; does not discriminate on a per message basis (everything is encrypted) with same key
  - When passing messages by using another intermediary, if the intermediary must “see” any part of the message, it can access all of it
- Message level security (WS-Security)
  - Message-based security provides finer granularity
  - Security is applied on a per message basis
  - Parts of a message can be (multiply) encrypted and signed on a “need-to-know” basis
  - WS-Security can be used with insecure transports

Figure 8-12. Transport level security and message security WM6461.0

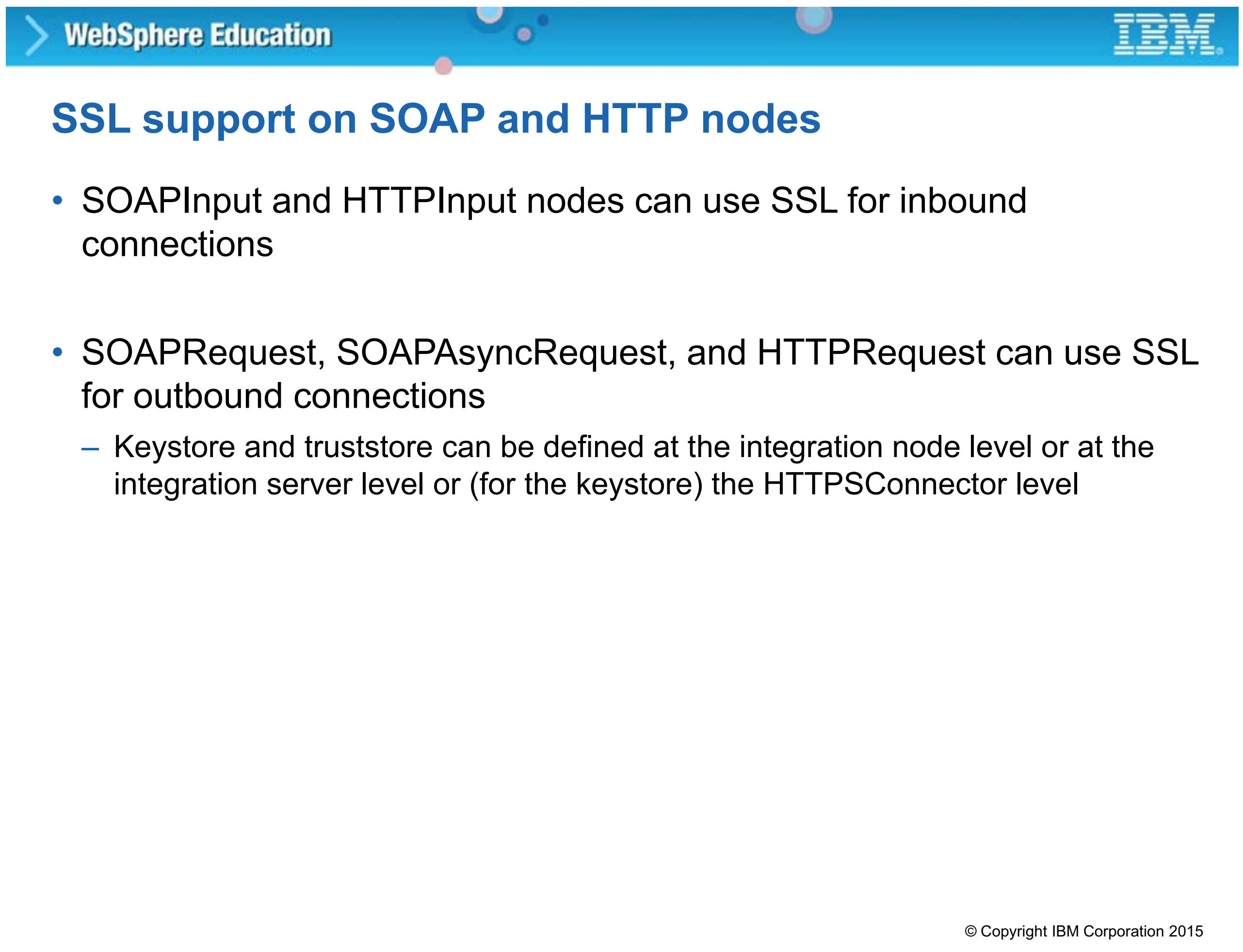
Notes:

There are many elements to security and SOAP messages. The task is to keep the message secure through intermediate systems until it reaches the ultimate recipient.

At the transport level, you can use SSL and HTTPS.

At the message level, you can use WS-Security. WS-Security is based on securing SOAP messages through an XML digital signature, confidentiality through XML encryption, and credential propagation through security tokens. The WS-Security specification defines the core facilities for protecting the integrity and confidentiality of a message and provides mechanisms for associating security-related claims with the message.

8.2. SSL on HTTP connections



WebSphere Education

SSL support on SOAP and HTTP nodes

- SOAPInput and HTTPInput nodes can use SSL for inbound connections
- SOAPRequest, SOAPAsyncRequest, and HTTPRequest can use SSL for outbound connections
  - Keystore and truststore can be defined at the integration node level or at the integration server level or (for the keystore) the HTTPSConnector level

© Copyright IBM Corporation 2015

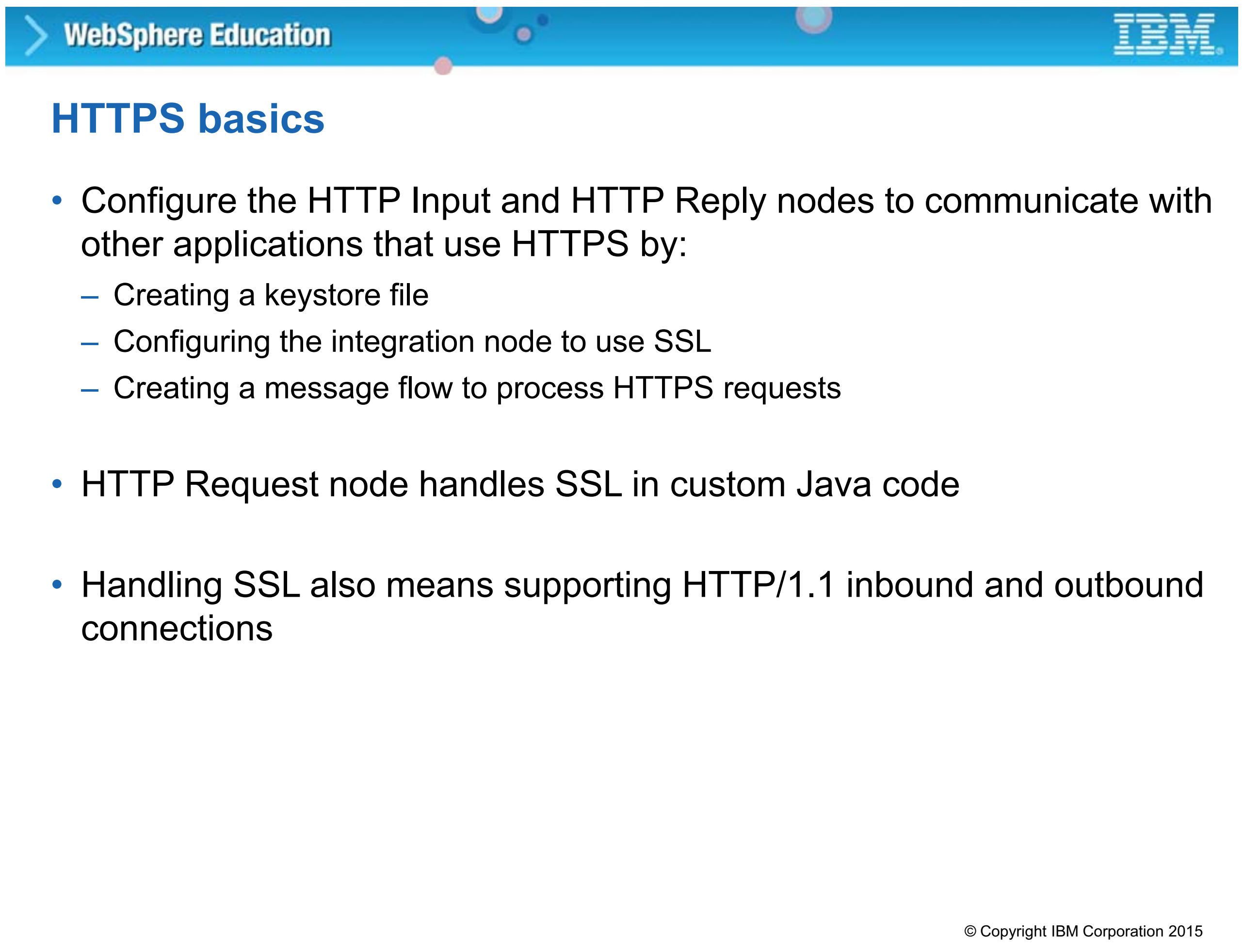
Figure 8-13. SSL support on SOAP and HTTP nodes WM6461.0

Notes:

Integration Bus can support SSL on inbound connections by using a SOAP Input node or HTTP Input node in a message flow.

Integration Bus can support SSL on output connections by using a SOAP Request node, SOAP AsyncRequestNode, or HTTP Request node in a message flow.

For outbound connections, the keystore and truststore can be defined at the integration node or integration server.



WebSphere Education IBM.

HTTPS basics

- Configure the HTTP Input and HTTP Reply nodes to communicate with other applications that use HTTPS by:
  - Creating a keystore file
  - Configuring the integration node to use SSL
  - Creating a message flow to process HTTPS requests
- HTTP Request node handles SSL in custom Java code
- Handling SSL also means supporting HTTP/1.1 inbound and outbound connections

© Copyright IBM Corporation 2015

Figure 8-14. HTTPS basics WM6461.0

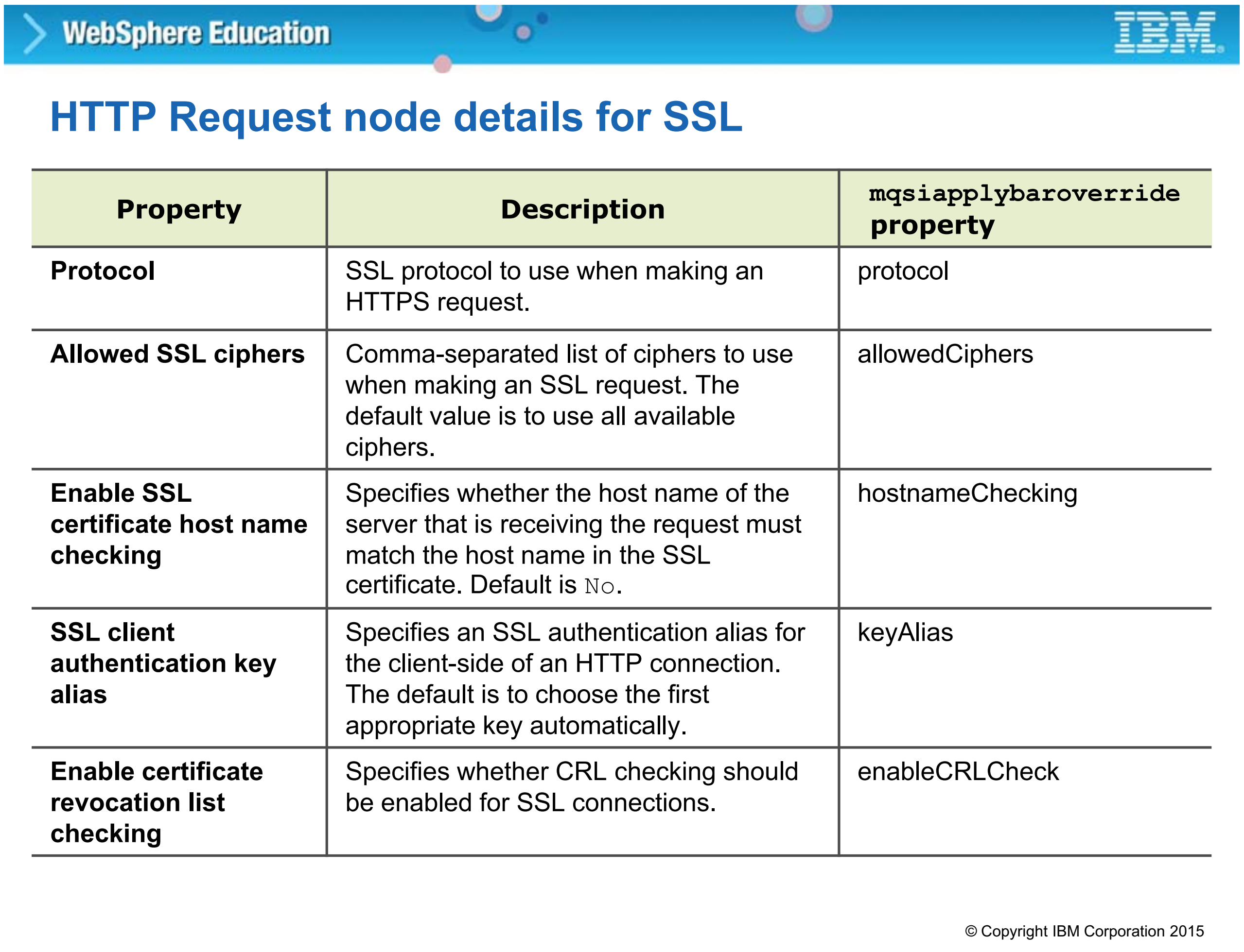
Notes:

The implementation of HTTPS support allows HTTP Request nodes to call web services over a secure HTTP connection and allows the HTTP Input node to receive its requests over a secure HTTP connection.

Before you start, you must set a public key infrastructure (PKI) at integration node. Then, configure the HTTP Input and HTTP Reply nodes to communicate with other applications that use HTTPS by completing the following steps:

- Create a keystore file
- Configure the integration node or integration server to use SSL
- Create a message flow to process HTTPS requests

If you are using the integration node listener, configure the integration node to use SSL. If you are using the integration server listener, configure the integration server to use SSL.



WebSphere Education

HTTP Request node details for SSL

Property	Description	`mqsiapplybaroverride` property
Protocol	SSL protocol to use when making an HTTPS request.	protocol
Allowed SSL ciphers	Comma-separated list of ciphers to use when making an SSL request. The default value is to use all available ciphers.	allowedCiphers
Enable SSL certificate host name checking	Specifies whether the host name of the server that is receiving the request must match the host name in the SSL certificate. Default is `No`.	hostnameChecking
SSL client authentication key alias	Specifies an SSL authentication alias for the client-side of an HTTP connection. The default is to choose the first appropriate key automatically.	keyAlias
Enable certificate revocation list checking	Specifies whether CRL checking should be enabled for SSL connections.	enableCRLCheck

© Copyright IBM Corporation 2015

Figure 8-15. HTTP Request node details for SSL WM6461.0

Notes:

Developers can set up their message flows to use SSL by setting the appropriate properties. However, it is also possible for the administrator to implement SSL by tailoring the BAR file or apply BAR file overrides from a command.

When implementing SSL, there are three possible values for the protocol. If SSL is not implemented, any value in this property is ignored.

- **SSL** (the default): This setting tries to connect by using the SSLv3 protocol first, but allows the handshake to fall back to the SSLv2 protocol where the underlying JSSE provider supports the SSLv2 protocol.
- **SSLv3**: This setting tries to connect with the SSLv3 protocol only. Fallback to SSLv2 is not allowed.
- **TLS**: This setting tries to connect with the TLS protocol only. Fallback to SSLv3 or SSLv2 is not allowed.

Both ends of an SSL connection must agree on the protocol to use. So, the selected protocol must be one that the remote server can accept.

WebSphere Education

HTTPS client key selection (1 of 2)

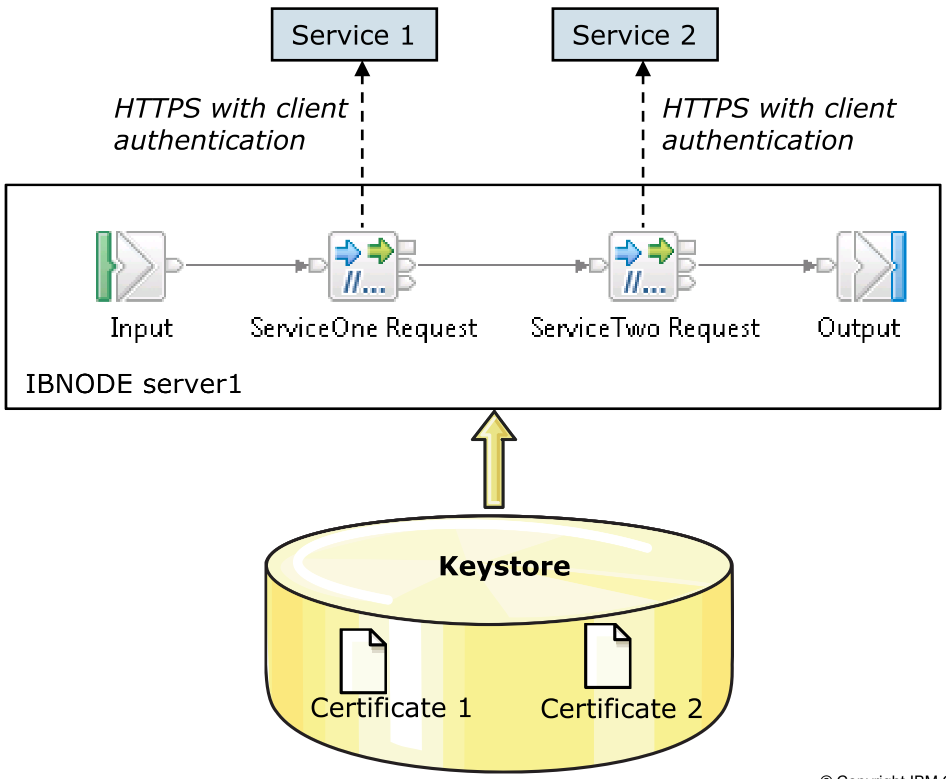


Figure 8-16. HTTPS client key selection (1 of 2) WM6461.0

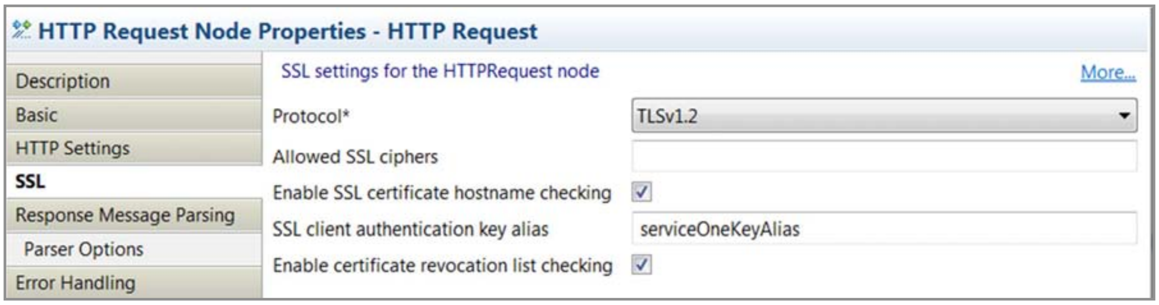
Notes:

In some applications, it might be necessary to use a different key for each service request from an HTTP Request node in a message flow. The developer can identify a certificate for each HTTP Request by using an SSL key alias.

WebSphere Education IBM

HTTPS client key selection (2 of 2)

- SSL key aliases
  - Allow SSL-based nodes to specify a “key alias” to identify the correct key for a connection
  - Allow integration node to communicate with remote servers by using different keys
  - Supported by all SSL-enabled message processing nodes (HTTP, SOAP, TCPIP, WSRR, LDAP, and JMS)
  - Supported for both client and server connections, by using either one-way or mutual authentication
  - Specify as node property or override in code by using local environment variable


HTTP Request Node Properties - HTTP Request

SSL settings for the HTTPRequest node

Protocol*	TLSv1.2
Allowed SSL ciphers	
Enable SSL certificate hostname checking	✓
SSL client authentication key alias	serviceOneKeyAlias
Enable certificate revocation list checking	✓

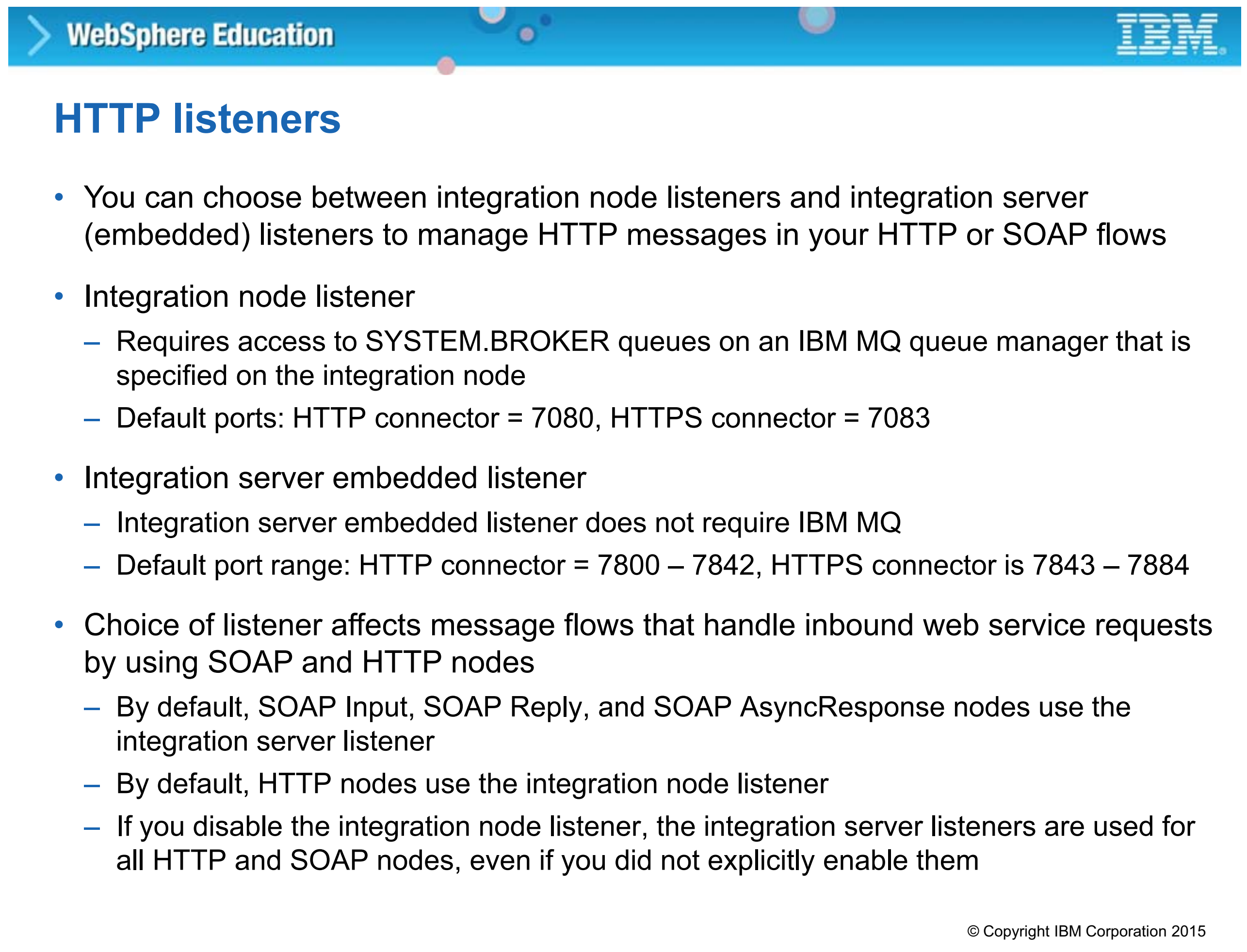
© Copyright IBM Corporation 2015

Figure 8-17. HTTPS client key selection (2 of 2) WM6461.0

Notes:

A key alias identifies the key that an SSL connection uses, if the keystore for your integration node or integration server contains more than one key.

The **SSL client authentication key alias** property specifies an SSL authentication alias for the client-side of an HTTP connection. Accepting the default value means that the first appropriate key is chosen for automatically. The alias value maps to the alias value in a policy set.



HTTP listeners

- You can choose between integration node listeners and integration server (embedded) listeners to manage HTTP messages in your HTTP or SOAP flows
- Integration node listener
  - Requires access to SYSTEM.BROKER queues on an IBM MQ queue manager that is specified on the integration node
  - Default ports: HTTP connector = 7080, HTTPS connector = 7083
- Integration server embedded listener
  - Integration server embedded listener does not require IBM MQ
  - Default port range: HTTP connector = 7800 – 7842, HTTPS connector is 7843 – 7884
- Choice of listener affects message flows that handle inbound web service requests by using SOAP and HTTP nodes
  - By default, SOAP Input, SOAP Reply, and SOAP AsyncResponse nodes use the integration server listener
  - By default, HTTP nodes use the integration node listener
  - If you disable the integration node listener, the integration server listeners are used for all HTTP and SOAP nodes, even if you did not explicitly enable them

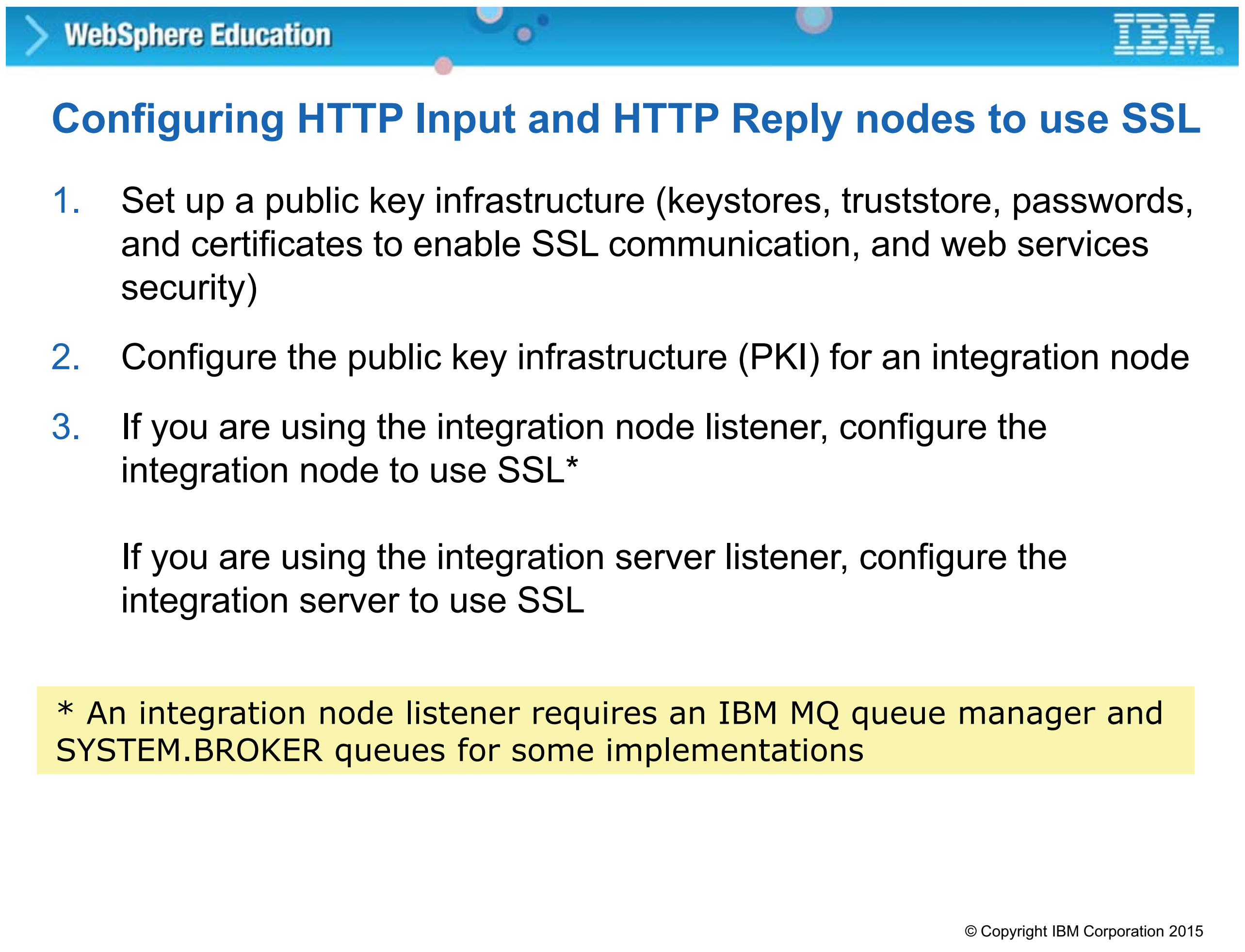
Figure 8-18. HTTP listeners WM6461.0

Notes:

In Integration Bus, you can choose between an integration node listener and an integration server listener to manage HTTP messages.

As an option, you can use the embedded listener at the integration server level. You can change your configuration such that some integration servers use the integration node listener for HTTP nodes, SOAP nodes, or both, and other integration servers use the embedded listener for HTTP nodes, SOAP nodes, or both.

However, if you disable the integration node listener, the integration server listeners are used for all HTTP and SOAP nodes, even if you did not explicitly enable support for them. Therefore, if you set all relevant integration node and integration server properties to false, the integration server listeners handle all HTTP messages.



WebSphere Education IBM

Configuring HTTP Input and HTTP Reply nodes to use SSL

1. Set up a public key infrastructure (keystores, truststore, passwords, and certificates to enable SSL communication, and web services security)
2. Configure the public key infrastructure (PKI) for an integration node
3. If you are using the integration node listener, configure the integration node to use SSL*

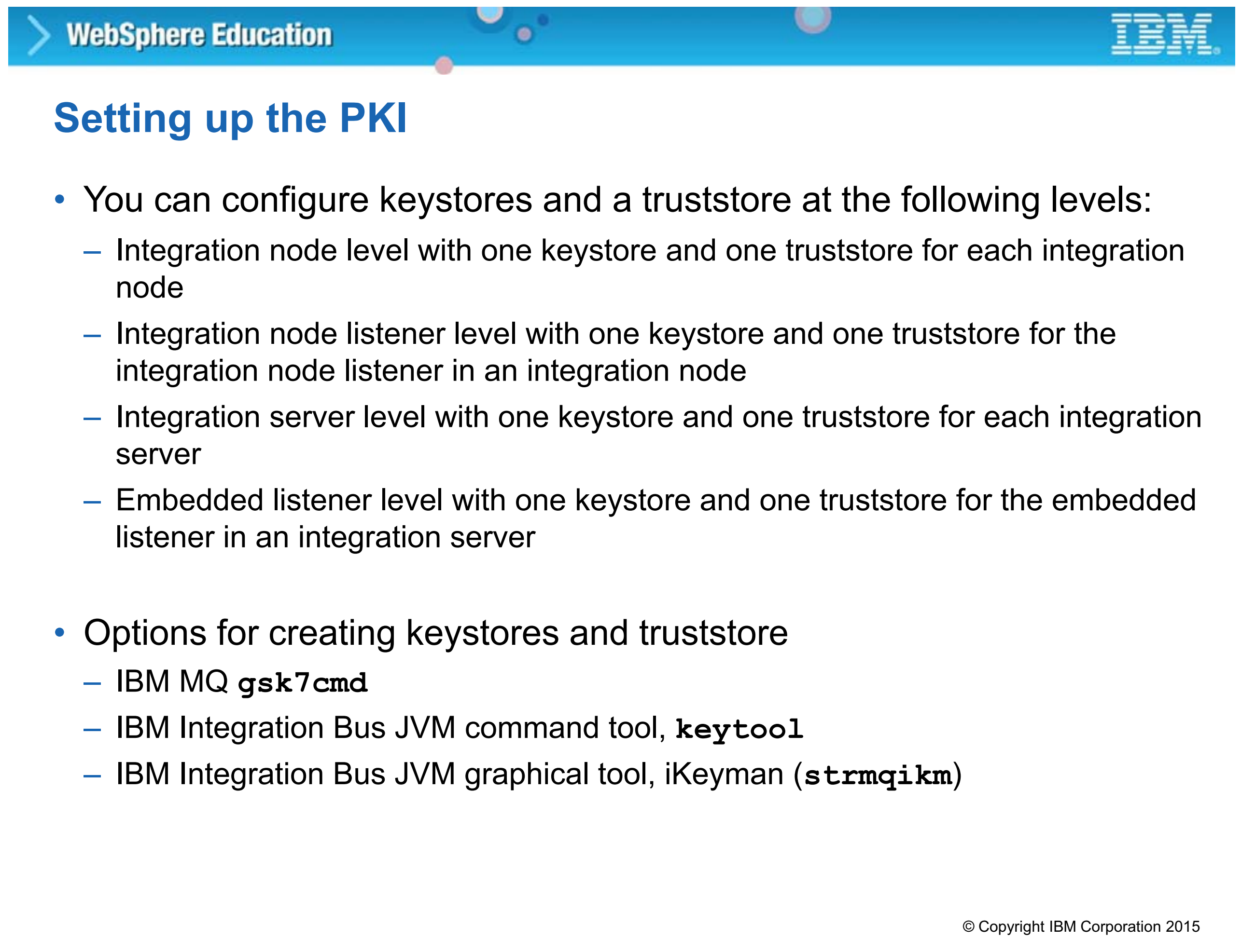
   If you are using the integration server listener, configure the integration server to use SSL

* An integration node listener requires an IBM MQ queue manager and SYSTEM.BROKER queues for some implementations

Figure 8-19. Configuring HTTP Input and HTTP Reply nodes to use SSL WM6461.0

Notes:

This figure summarizes the steps that you must complete to enable SSL for Integration Bus message flows. Each of these steps is described in detail next.



Setting up the PKI

- You can configure keystores and a truststore at the following levels:
  - Integration node level with one keystore and one truststore for each integration node
  - Integration node listener level with one keystore and one truststore for the integration node listener in an integration node
  - Integration server level with one keystore and one truststore for each integration server
  - Embedded listener level with one keystore and one truststore for the embedded listener in an integration server
- Options for creating keystores and truststore
  - IBM MQ **`gsk7cmd`**
  - IBM Integration Bus JVM command tool, **`keytool`**
  - IBM Integration Bus JVM graphical tool, iKeyman (**`strmqikm`**)

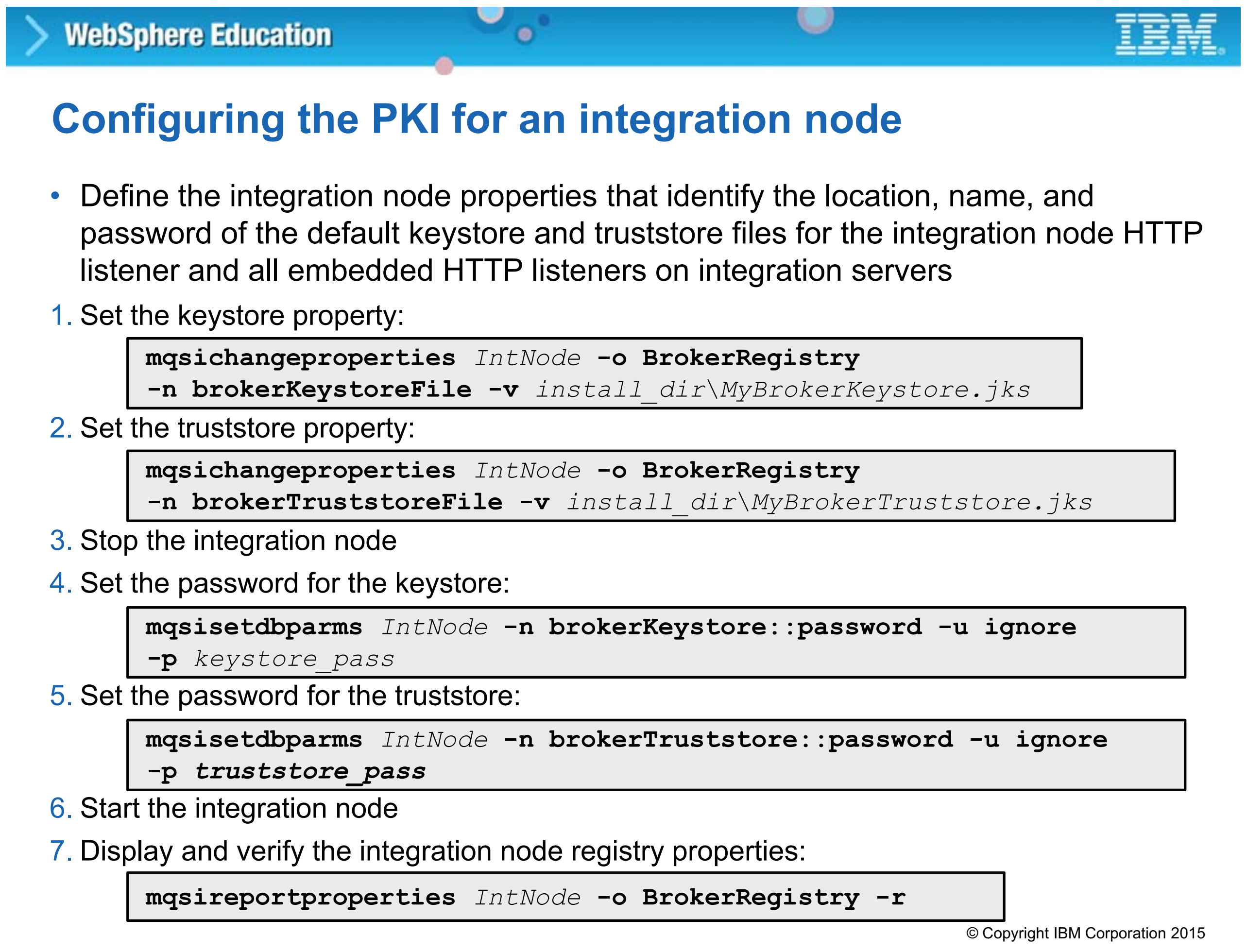
Figure 8-20. Setting up the PKI WM6461.0

Notes:

You can configure keystores and truststore at the following levels:

- Integration node level (one keystore, and one truststore for each integration node).
- Integration node listener level (one keystore and one truststore for the integration node listener in an integration node). Integration node listeners that do not have a PKI configured use the integration node level PKI configuration.
- Integration server level (one keystore and one truststore for each integration server). Integration servers that do not have a PKI configured use the integration node level PKI configuration.
- Embedded listener level (one keystore and one truststore for the embedded listener in an integration server). Embedded listeners that do not have a PKI configured use the integration server level PKI configuration. If there is no integration server level PKI configuration, then the embedded listener uses the integration node level PKI configuration.

You can use IBM MQ or Integration Bus tools to create the keystores and truststore. Keytool and iKeyman are in the *`install_dir`*`\IBM\IIB\10.0.0.0\common\jdk\jre\bin` directory on Windows.



WebSphere Education

Configuring the PKI for an integration node

- Define the integration node properties that identify the location, name, and password of the default keystore and truststore files for the integration node HTTP listener and all embedded HTTP listeners on integration servers

1. Set the keystore property:

```
mqsichangeproperties IntNode -o BrokerRegistry
-n brokerKeystoreFile -v install_dir\MyBrokerKeystore.jks
```

2. Set the truststore property:

```
mqsichangeproperties IntNode -o BrokerRegistry
-n brokerTruststoreFile -v install_dir\MyBrokerTruststore.jks
```

3. Stop the integration node
4. Set the password for the keystore:

```
mqsisetdbparms IntNode -n brokerKeystore::password -u ignore
-p keystore_pass
```

5. Set the password for the truststore:

```
mqsisetdbparms IntNode -n brokerTruststore::password -u ignore
-p truststore_pass
```

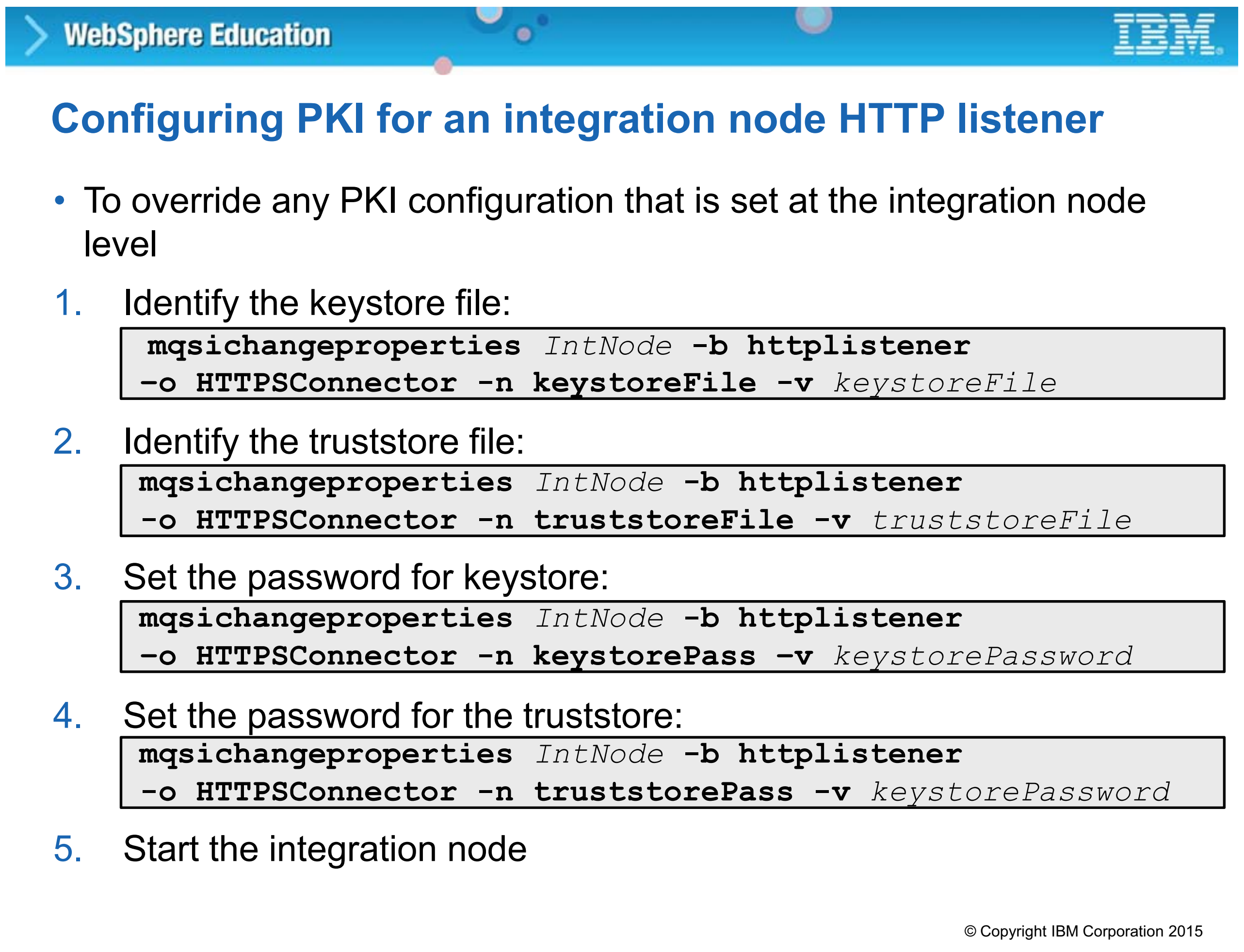
6. Start the integration node
7. Display and verify the integration node registry properties:

```
mqsireportproperties IntNode -o BrokerRegistry -r
```

Figure 8-21. Configuring the PKI for an integration node WM6461.0

Notes:

This figure shows the steps for configuring the PKI for an integration node.



WebSphere Education

Configuring PKI for an integration node HTTP listener

- To override any PKI configuration that is set at the integration node level

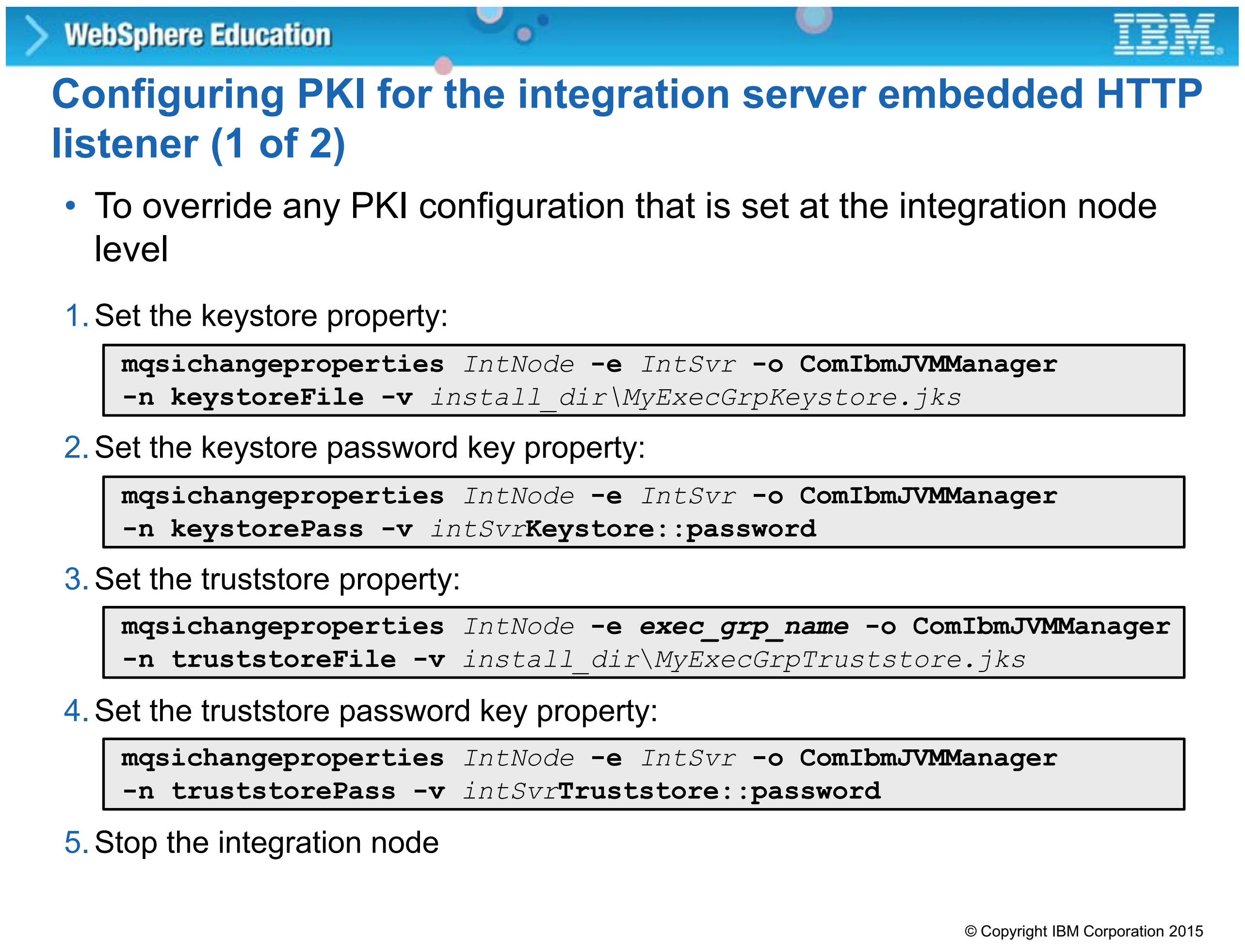
1. Identify the keystore file:
```
mqsichangeproperties IntNode -b httplistener
-o HTTPSConnector -n keystoreFile -v keystoreFile
```
2. Identify the truststore file:
```
mqsichangeproperties IntNode -b httplistener
-o HTTPSConnector -n truststoreFile -v truststoreFile
```
3. Set the password for keystore:
```
mqsichangeproperties IntNode -b httplistener
-o HTTPSConnector -n keystorePass -v keystorePassword
```
4. Set the password for the truststore:
```
mqsichangeproperties IntNode -b httplistener
-o HTTPSConnector -n truststorePass -v keystorePassword
```
5. Start the integration node

Figure 8-22. Configuring PKI for an integration node HTTP listener WM6461.0

Notes:

This figure lists the steps for configuring the PKI for integration node HTTP listener. These steps assume that you already created the keystore and truststore files.

In these steps, you define the integration node properties that identify the location, name, and password of the keystore and truststore files.



WebSphere Education

Configuring PKI for the integration server embedded HTTP listener (1 of 2)

- To override any PKI configuration that is set at the integration node level

1. Set the keystore property:

```
mqsichangeproperties IntNode -e IntSvr -o ComIbmJVMManager
-n keystoreFile -v install_dir\MyExecGrpKeystore.jks
```

2. Set the keystore password key property:

```
mqsichangeproperties IntNode -e IntSvr -o ComIbmJVMManager
-n keystorePass -v intSvrKeystore::password
```

3. Set the truststore property:

```
mqsichangeproperties IntNode -e exec_grp_name -o ComIbmJVMManager
-n truststoreFile -v install_dir\MyExecGrpTruststore.jks
```

4. Set the truststore password key property:

```
mqsichangeproperties IntNode -e IntSvr -o ComIbmJVMManager
-n truststorePass -v intSvrTruststore::password
```

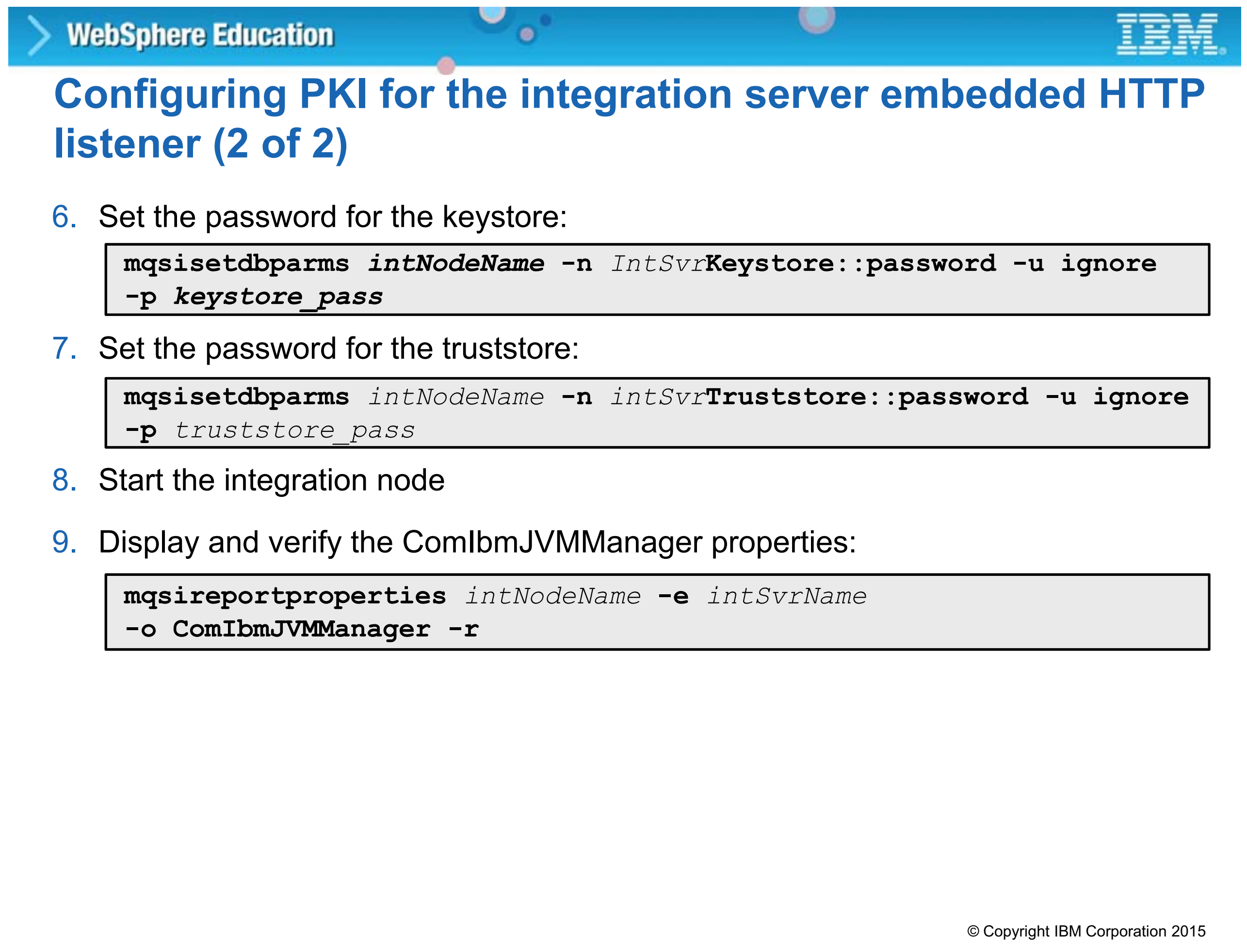
5. Stop the integration node

© Copyright IBM Corporation 2015

Figure 8-23. Configuring PKI for the integration server embedded HTTP listener (1 of 2) WM6461.0

Notes:

As an option, you can configure PKI for the integration server. When you configure the PKI on an integration server, it overrides the configuration that is set at the integration node level.



WebSphere Education

Configuring PKI for the integration server embedded HTTP listener (2 of 2)

6. Set the password for the keystore:

```
mqsisetdbparms intNodeName -n IntSvrKeystore::password -u ignore
-p keystore_pass
```

7. Set the password for the truststore:

```
mqsisetdbparms intNodeName -n intSvrTruststore::password -u ignore
-p truststore_pass
```

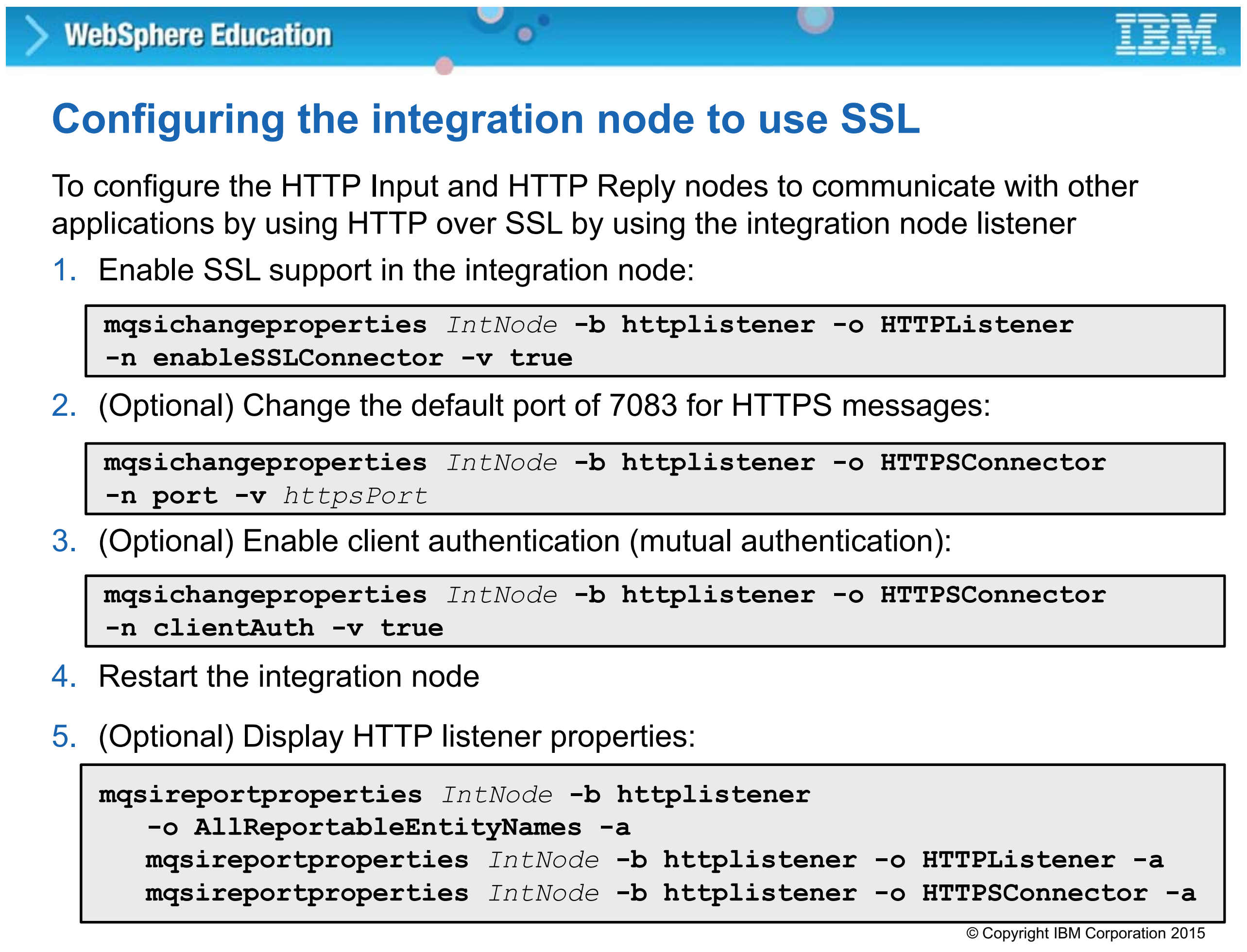
8. Start the integration node

9. Display and verify the ComIbmJVMManager properties:

```
mqsireportproperties intNodeName -e intSvrName
-o ComIbmJVMManager -r
```

Figure 8-24. Configuring PKI for the integration server embedded HTTP listener (2 of 2) WM6461.0

Notes:



WebSphere Education

Configuring the integration node to use SSL

To configure the HTTP Input and HTTP Reply nodes to communicate with other applications by using HTTP over SSL by using the integration node listener

1. Enable SSL support in the integration node:

```
mqsichangeproperties IntNode -b httplistener -o HTTPListener
-n enableSSLConnector -v true
```

2. (Optional) Change the default port of 7083 for HTTPS messages:

```
mqsichangeproperties IntNode -b httplistener -o HTTPSConnector
-n port -v httpsPort
```

3. (Optional) Enable client authentication (mutual authentication):

```
mqsichangeproperties IntNode -b httplistener -o HTTPSConnector
-n clientAuth -v true
```

4. Restart the integration node

5. (Optional) Display HTTP listener properties:

```
mqsireportproperties IntNode -b httplistener
   -o AllReportableEntityNames -a
   mqsireportproperties IntNode -b httplistener -o HTTPListener -a
   mqsireportproperties IntNode -b httplistener -o HTTPSConnector -a
```

© Copyright IBM Corporation 2015

Figure 8-25. Configuring the integration node to use SSL WM6461.0

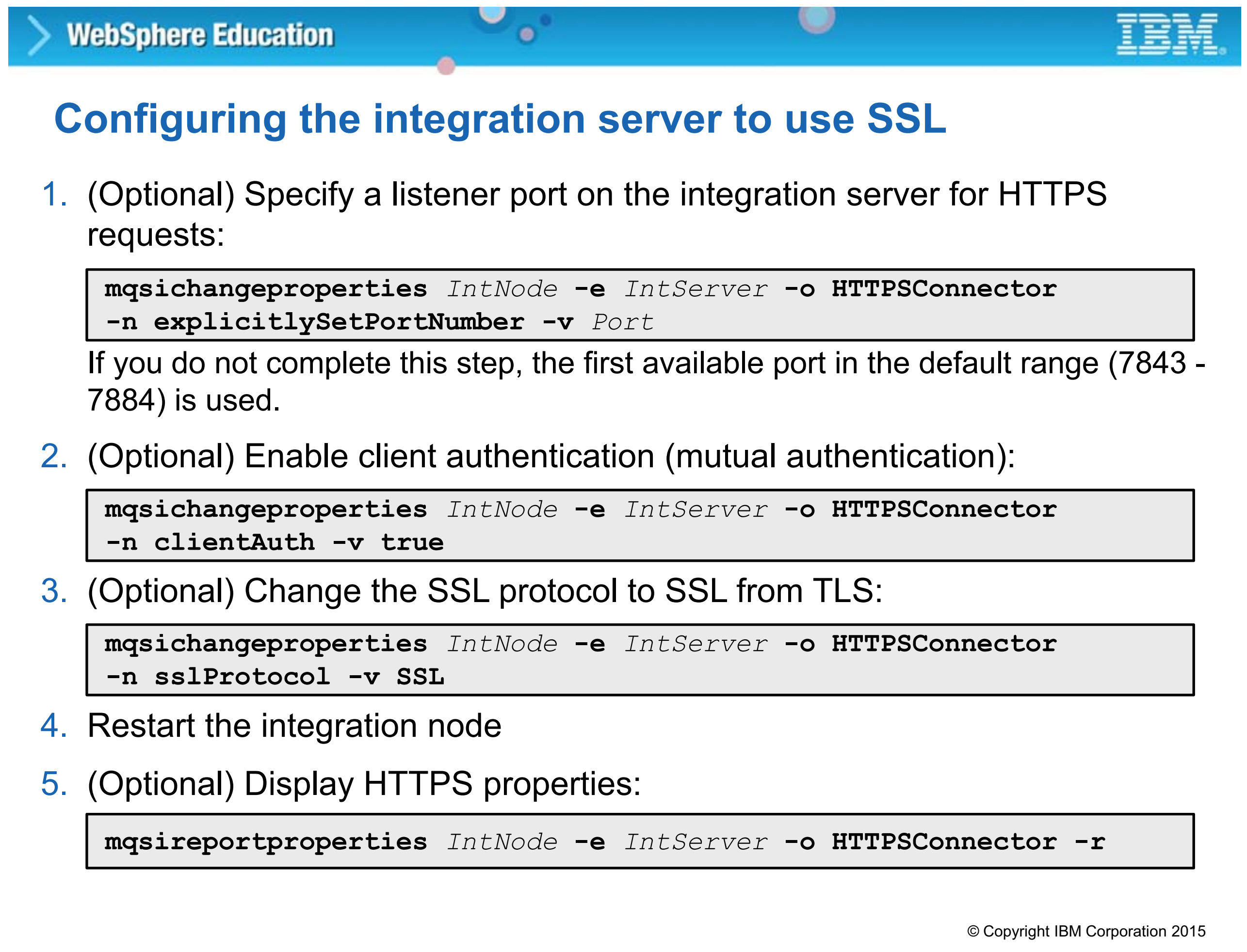
Notes:

The figure lists the steps for configuring the integration node to use SSL.

The keytool utility is used to create the necessary file (and certificates) for use in implementing the SSL support for the nodes. If the path to the keystore file changes, you must be sure to indicate this change by using the **mqsichangeproperties** command.

UNIX

On UNIX systems, only processes that run under a privileged user account (in most cases, root) can bind to ports lower than 1024. For the integration node to listen on these ports, the user ID under which the integration node is started must be “root”.



Configuring the integration server to use SSL

1. (Optional) Specify a listener port on the integration server for HTTPS requests:

```
mqsichangeproperties IntNode -e IntServer -o HTTPSConnector
-n explicitlySetPortNumber -v Port
```

   If you do not complete this step, the first available port in the default range (7843 - 7884) is used.

2. (Optional) Enable client authentication (mutual authentication):

```
mqsichangeproperties IntNode -e IntServer -o HTTPSConnector
-n clientAuth -v true
```

3. (Optional) Change the SSL protocol to SSL from TLS:

```
mqsichangeproperties IntNode -e IntServer -o HTTPSConnector
-n sslProtocol -v SSL
```

4. Restart the integration node

5. (Optional) Display HTTPS properties:

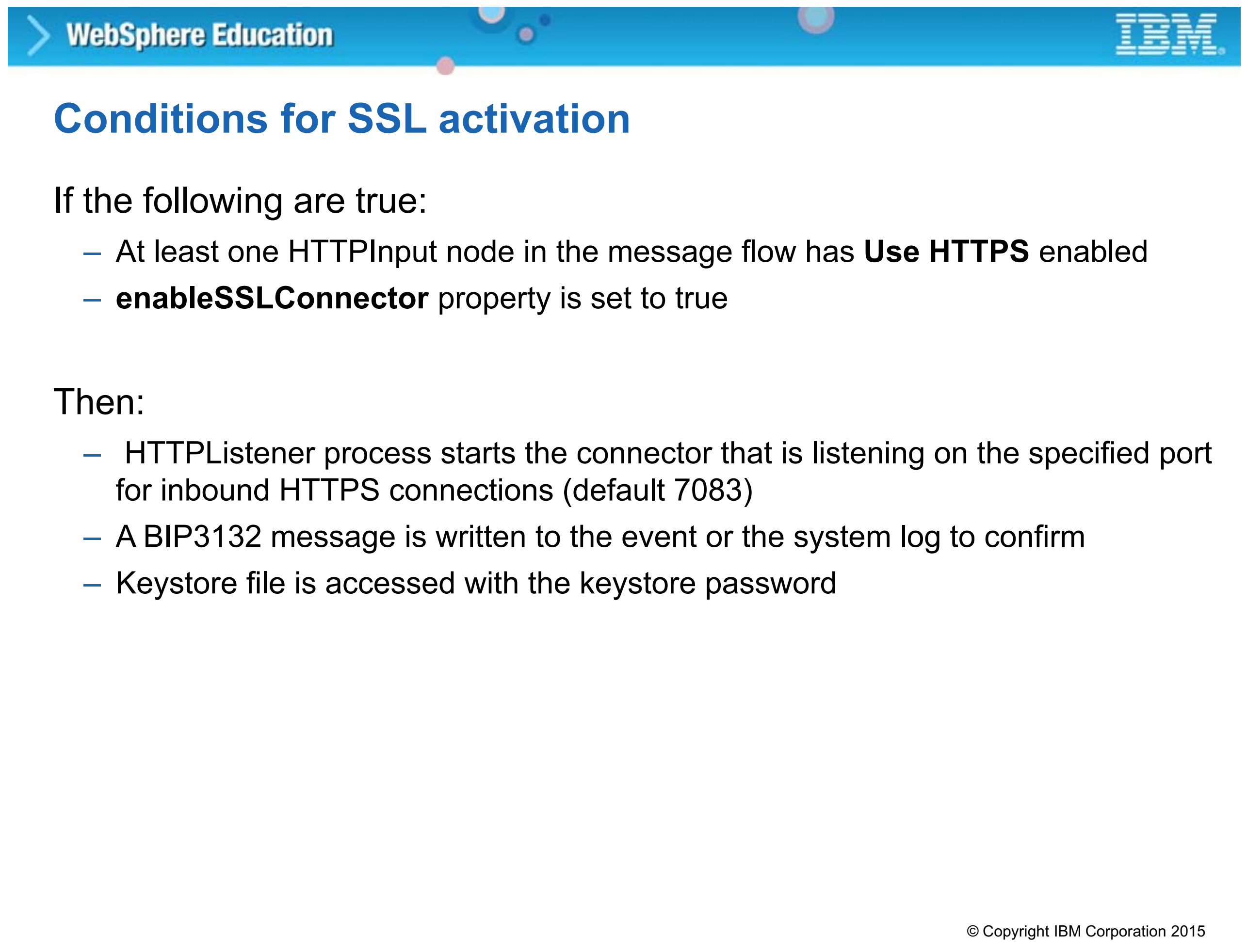
```
mqsireportproperties IntNode -e IntServer -o HTTPSConnector -r
```

Figure 8-26. Configuring the integration server to use SSL WM6461.0

Notes:

Each integration server has an embedded listener. The listener is associated with an HTTPConnector object and an HTTPSConnector object. The HTTPConnector object controls the runtime properties that affect the handling of HTTP messages.

If you are using the integration server listener, follow the steps in the figure to configure the integration server to use SSL.



WebSphere Education

Conditions for SSL activation

If the following are true:

- At least one HTTPInput node in the message flow has **Use HTTPS** enabled
- **enableSSLConnector** property is set to true

Then:

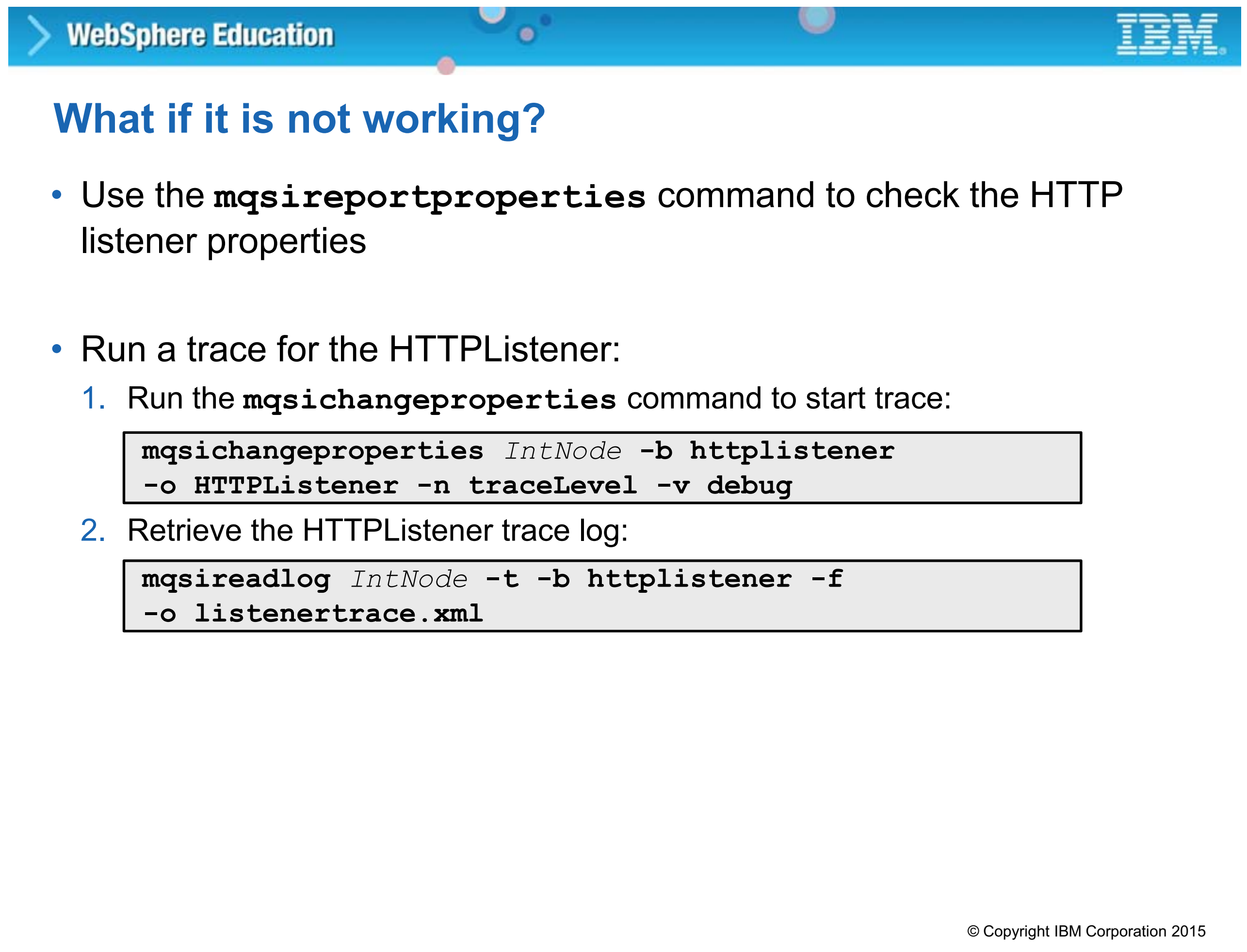
- HTTPListener process starts the connector that is listening on the specified port for inbound HTTPS connections (default 7083)
- A BIP3132 message is written to the event or the system log to confirm
- Keystore file is accessed with the keystore password

Figure 8-27. Conditions for SSL activation WM6461.0

Notes:

The HTTP listener is started for the SSL port if all the conditions are met, the setup is complete, and the first HTTPS message flow is successfully deployed.

It is important that the keystore is properly set up and available before SSL activation, as the keystore file is accessed when the HTTP listener is started.



WebSphere Education

What if it is not working?

- Use the **`mqsireportproperties`** command to check the HTTP listener properties
- Run a trace for the HTTPListener:
  1. Run the **`mqsichangeproperties`** command to start trace:

```
mqsichangeproperties IntNode -b httplistener
-o HTTPListener -n traceLevel -v debug
```

  2. Retrieve the HTTPListener trace log:

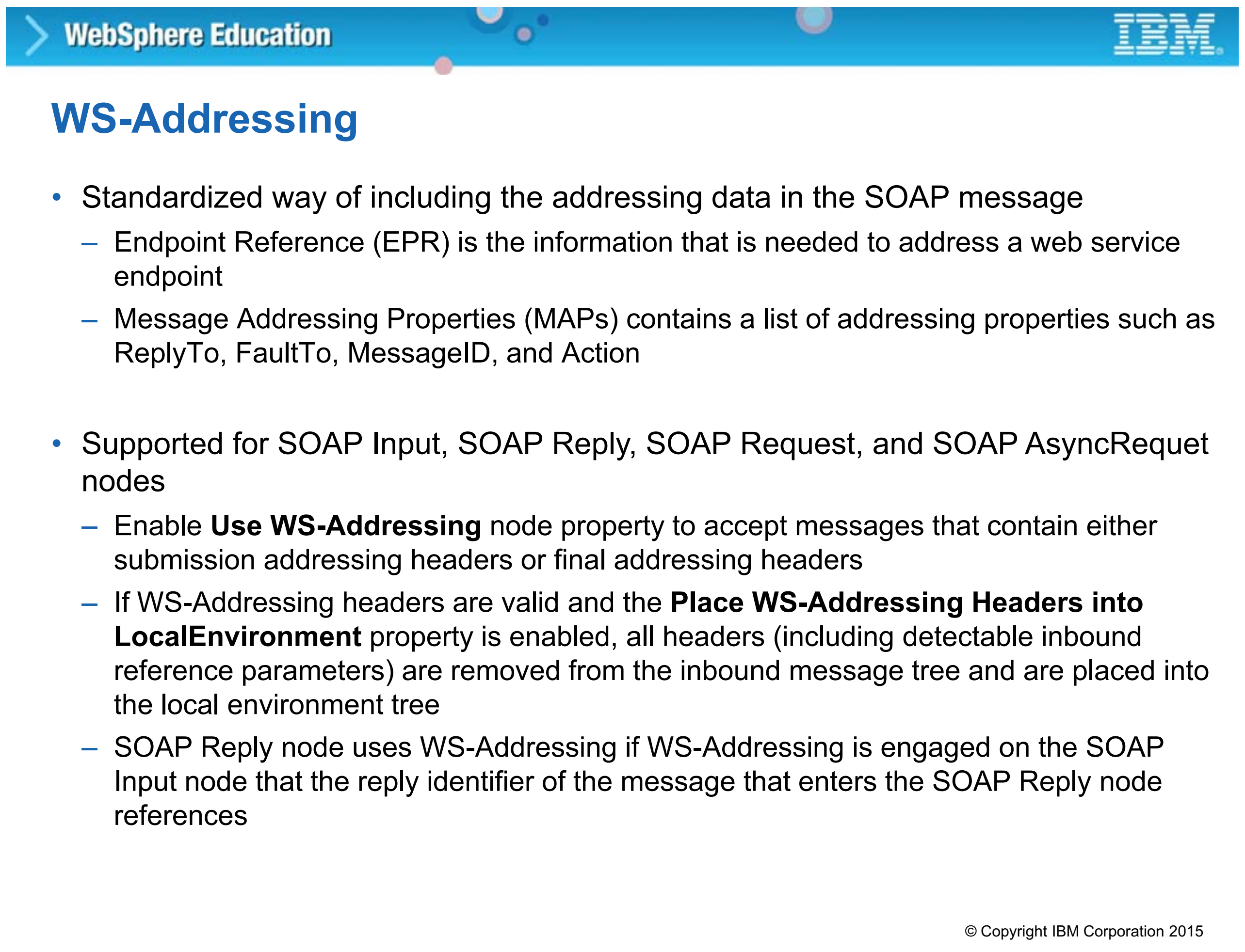
```
mqsireadlog IntNode -t -b httplistener -f
-o listenertrace.xml
```

Figure 8-28. What if it is not working? WM6461.0

Notes:

A service trace of the HTTPListener and the integration server provides some information if the listener does not activate or if the message flows do not seem to be working.

8.3. WS-Addressing and WS-Security



WebSphere Education IBM

WS-Addressing

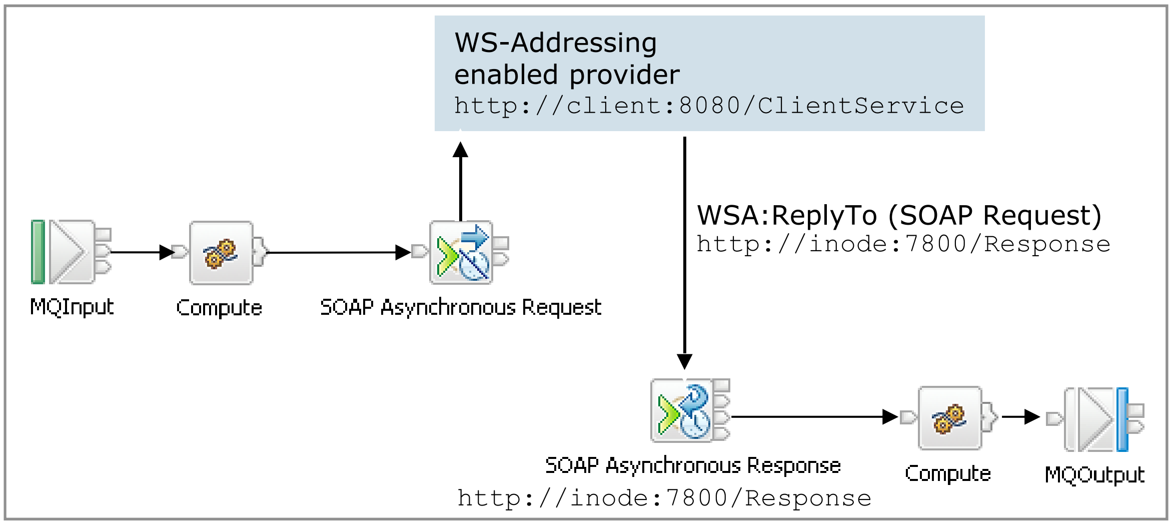
- Standardized way of including the addressing data in the SOAP message
  - Endpoint Reference (EPR) is the information that is needed to address a web service endpoint
  - Message Addressing Properties (MAPs) contains a list of addressing properties such as ReplyTo, FaultTo, MessageID, and Action
- Supported for SOAP Input, SOAP Reply, SOAP Request, and SOAP AsyncRequet nodes
  - Enable **Use WS-Addressing** node property to accept messages that contain either submission addressing headers or final addressing headers
  - If WS-Addressing headers are valid and the **Place WS-Addressing Headers into LocalEnvironment** property is enabled, all headers (including detectable inbound reference parameters) are removed from the inbound message tree and are placed into the local environment tree
  - SOAP Reply node uses WS-Addressing if WS-Addressing is engaged on the SOAP Input node that the reply identifier of the message that enters the SOAP Reply node references

Figure 8-29. WS-Addressing WM6461.0

Notes:

WebSphere Education IBM

WS-Addressing asynchronous consumer scenario

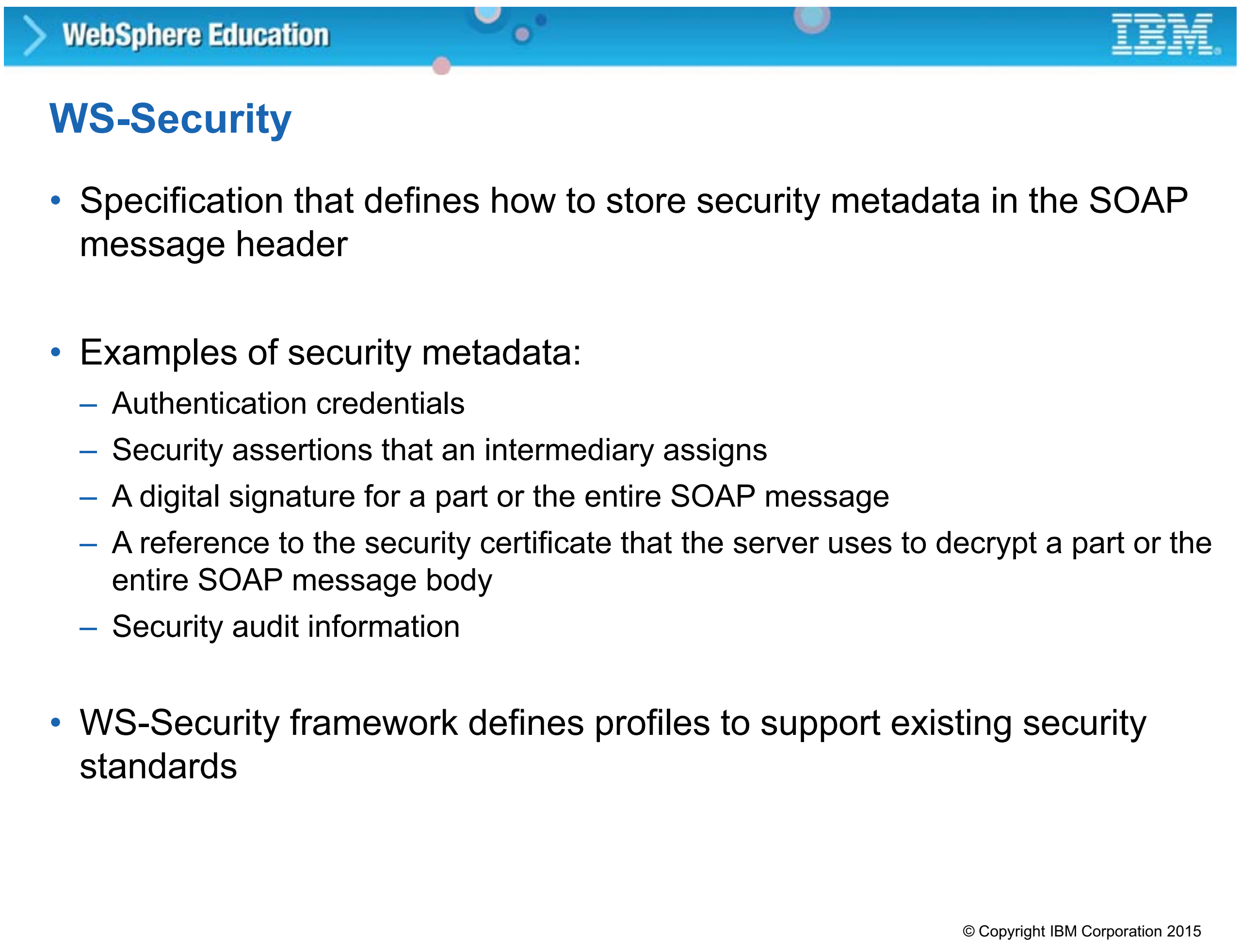

WS-Addressing
enabled provider
http://client:8080/ClientService
WSA:ReplyTo (SOAP Request)
http://inode:7800/Response
MQInput
Compute
SOAP Asynchronous Request
SOAP Asynchronous Response
http://inode:7800/Response
Compute
MQOutput

- SOAP Asynchronous Request node places the return address of the associated SOAP Asynchronous Response node into the **WSA:ReplyTo** fields

Figure 8-30. WS-Addressing asynchronous consumer scenario WM6461.0

Notes:

This figure shows an example of a SOAP request/response scenario that uses WS-Addressing.



WS-Security

- Specification that defines how to store security metadata in the SOAP message header
- Examples of security metadata:
  - Authentication credentials
  - Security assertions that an intermediary assigns
  - A digital signature for a part or the entire SOAP message
  - A reference to the security certificate that the server uses to decrypt a part or the entire SOAP message body
  - Security audit information
- WS-Security framework defines profiles to support existing security standards

Figure 8-31. WS-Security WM6461.0

Notes:

WS-Security describes how to encode binary security tokens and attach them to SOAP messages. Specifically, the WS-Security profile specifications describe how to encode user name Tokens and X.509 Tokens. With WS-Security, the domain of these mechanisms can be extended by carrying authentication information in web services requests. WS-Security also includes extensibility mechanisms that can be used to further describe the credentials that are included with a message. WS-Security is a building block that can be used with other web service protocols to address a range of application security requirements.

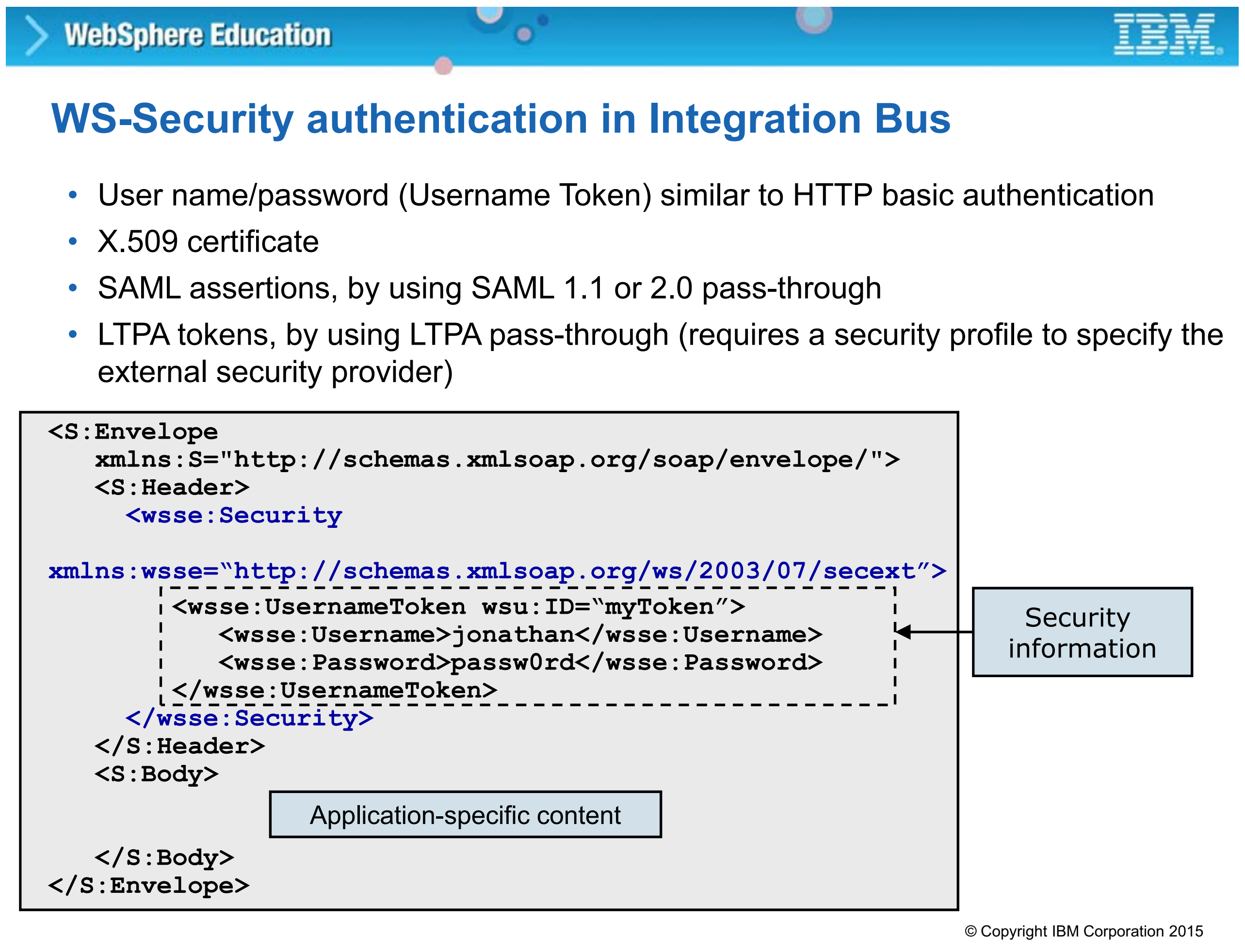
There are numerous advantages to using WS-Security.

- Different parts of a message can be secured in various ways. For example, you can use integrity on the security token (user ID and password) and confidentiality on the SOAP message body.
- Intermediaries can be used and end-to-end message-level security can be provided through any number of intermediaries.
- WS-Security works across multiple transports and is independent of the underlying transport protocol.

- Authentication of both individual users and multiple party identities is possible.

Integration Bus focuses on the part of the WS-Security specification that is known as WS-Security Policy. This policy is implemented by using the properties of the SOAP Input or SOAP Request nodes.

Key areas that the Integration Bus encompasses include: authentication; message-level protection through digital signature, encryption or both; and message part protection through digital signature, encryption, or both.



WS-Security authentication in Integration Bus

- User name/password (Username Token) similar to HTTP basic authentication
- X.509 certificate
- SAML assertions, by using SAML 1.1 or 2.0 pass-through
- LTPA tokens, by using LTPA pass-through (requires a security profile to specify the external security provider)

```
<S:Envelope
   xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
     <wsse:Security

xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/07/secext">
       <wsse:UsernameToken wsu:ID="myToken">
          <wsse:Username>jonathan</wsse:Username>
          <wsse:Password>passw0rd</wsse:Password>
       </wsse:UsernameToken>
     </wsse:Security>
   </S:Header>
   <S:Body>

   </S:Body>
</S:Envelope>
```

Figure 8-32. WS-Security authentication in Integration Bus WM6461.0

Notes:

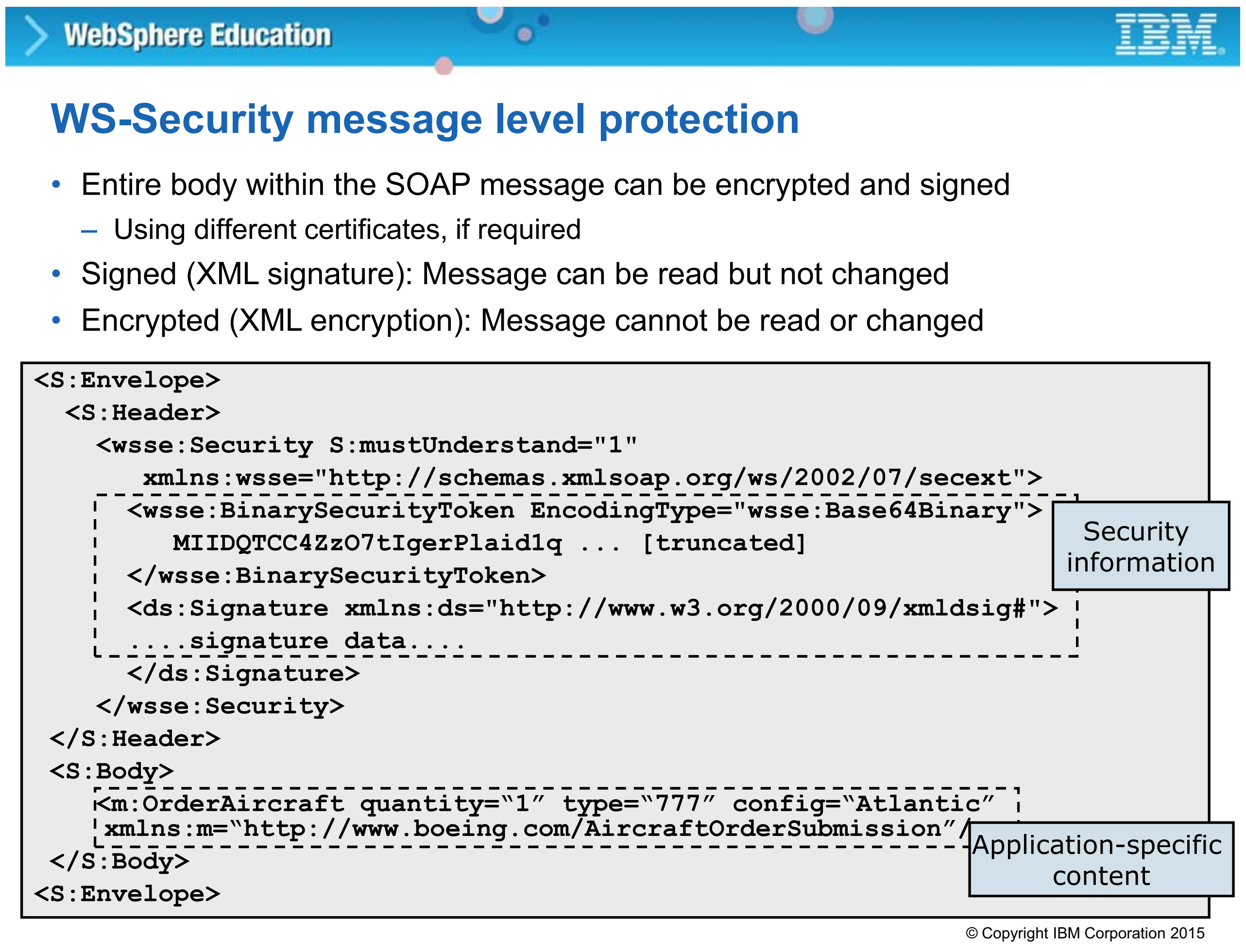
WS-Security defines two types of security tokens: a *user name* token and *binary security* token.

A user name token consists of a user name and optionally, password information. You can include a user name token directly as an element in the Security header within the SOAP message, rather than the HTTP header.

Binary tokens, such as X.509 certificates, Kerberos tickets, Lightweight Third-Party Authentication (LTPA) tokens, or other non-XML formats, require a special encoding for inclusion. The web services security specification describes how to encode binary security tokens such as X.509 certificates and Kerberos tickets, and it also describes how to include opaque encrypted keys. The specification also includes extensibility mechanisms that you can use to further describe the characteristics of the credentials that are included with a message.

In this example, a user name token is specified. The user name and password are specified in clear text. This text can be encrypted by using message part encryption, which is described later.

The Integration Bus administrator must enable security and specify a security profile. If these steps are not done, all user names and passwords are accepted.



WS-Security message level protection

- Entire body within the SOAP message can be encrypted and signed
  - Using different certificates, if required
- Signed (XML signature): Message can be read but not changed
- Encrypted (XML encryption): Message cannot be read or changed

```
<S:Envelope>
  <S:Header>
    <wsse:Security S:mustUnderstand="1"
       xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/07/secext">
      <wsse:BinarySecurityToken EncodingType="wsse:Base64Binary">
         MIIDQTCC4Zz07tIgerPlaid1q ... [truncated]
      </wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      ....signature data....
      </ds:Signature>
    </wsse:Security>
 </S:Header>
 <S:Body>
    <m:OrderAircraft quantity=“1” type=“777” config=“Atlantic”
     xmlns:m=“http://www.boeing.com/AircraftOrderSubmission”/>
 </S:Body>
<S:Envelope>
```

© Copyright IBM Corporation 2015

Figure 8-33. WS-Security message level protection WM6461.0

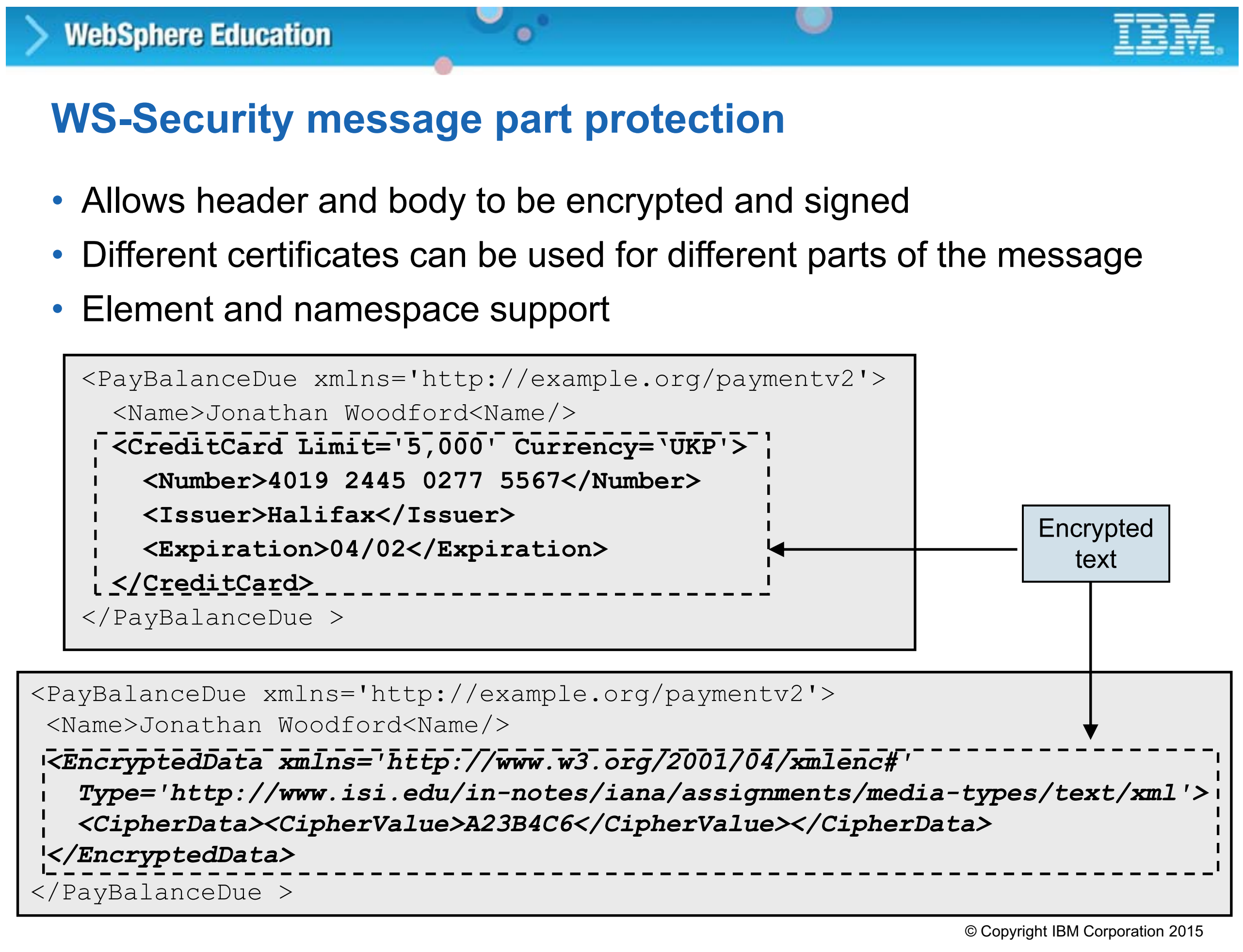
Notes:

Integration Bus also supports message protection. Either the entire message or part of the message can be protected.

The message can be encrypted, digitally signed, or both.

These different security functions can be used through different security certificates, and can be used separately from each other.

The example shows a payload that is digitally signed. However, it is not encrypted, and is still readable by intermediaries. By using this technique, you can allow intermediaries to access parts of the message, but inhibit access to other parts of the message.



WS-Security message part protection

- Allows header and body to be encrypted and signed
- Different certificates can be used for different parts of the message
- Element and namespace support

```
<PayBalanceDue xmlns='http://example.org/paymentv2'>
  <Name>Jonathan Woodford<Name/>
  <CreditCard Limit='5,000' Currency='UKP'>
    <Number>4019 2445 0277 5567</Number>
    <Issuer>Halifax</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PayBalanceDue >
```

```
<PayBalanceDue xmlns='http://example.org/paymentv2'>
 <Name>Jonathan Woodford<Name/>
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
  Type='http://www.isi.edu/in-notes/iana/assignments/media-types/text/xml'>
  <CipherData><CipherValue>A23B4C6</CipherValue></CipherData>
</EncryptedData>
</PayBalanceDue >
```

Figure 8-34. WS-Security message part protection WM6461.0

Notes:

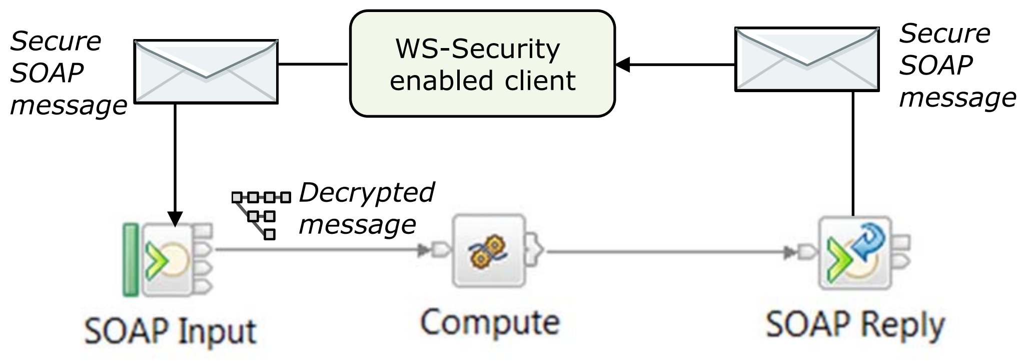
As was described previously, it is possible to protect certain parts of the SOAP message, rather than the entire message. You protect the message parts by specifying elements and namespaces within the header or body of the SOAP message that you want to sign or encrypt.

Because message part protection can specify the header, it can be used with user name tokens to protect the user name and password. The example shows a credit card transaction, where the credit card details are encrypted. The section at the bottom of the figure contains the message that is sent over the transmission channel. The sensitive credit card data is shown as encrypted.

The encrypted information in this example is truncated.

WebSphere Education

WS-Security on SOAP nodes



- SOAP nodes in the message flow handle the WS-Security function
  - SOAP Input node authenticates, decrypts, and verifies the signature of the message and message parts
  - Message passes decrypted through the message flow
  - SOAP Reply node encrypts and signs the appropriate parts of message

Figure 8-35. WS-Security on SOAP nodes WM6461.0

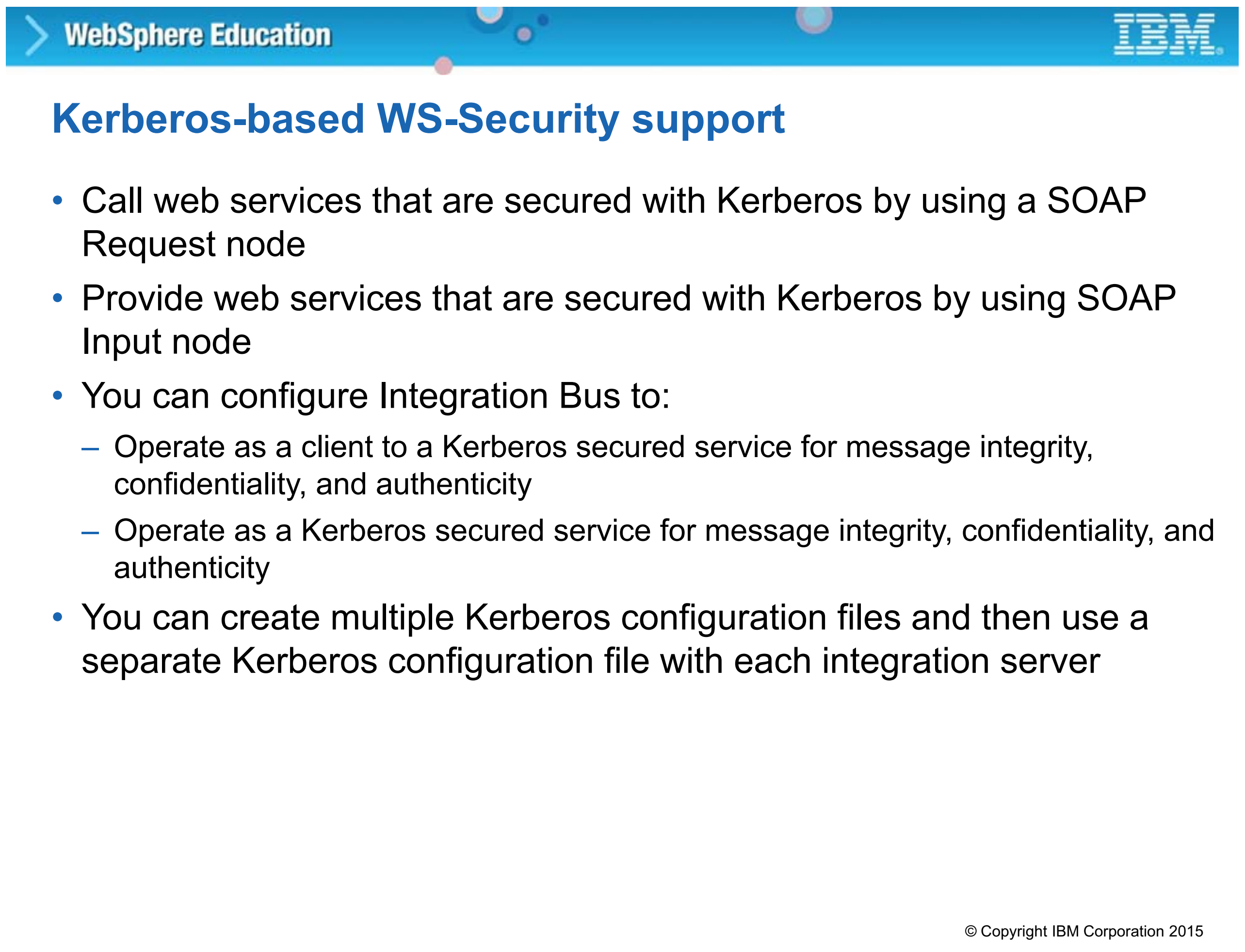
Notes:

This figure shows a scenario where the message flow represents the web service. A web service client starts the message flow by sending it a SOAP message. In this case, the SOAP message is secured by using WS-Security. The SOAP nodes handle the WS-Security function. No logic or action is required in the rest of the flow.

The role of the SOAP Input node is to verify all signed parts and decrypt the message as defined within the configuration. It does so by using the security configuration that it is given. The role of the SOAP Reply node is to apply any required WS-Security function. If the WS-Security configuration is applied to the whole message, then the entire message is decrypted for use by the message flow.

If the WS-Security configuration defines parts of the message, then the SOAP input node decrypts these parts for use by the message flow. Any parts of the message that Integration Bus does not have permission to access, because it does not contain the correct certificates, remain encrypted.

When the message reaches the SOAP Reply node, the message must be signed and encrypted, according to the definition of the WS-Security configuration for the message flow.



WebSphere Education

Kerberos-based WS-Security support

- Call web services that are secured with Kerberos by using a SOAP Request node
- Provide web services that are secured with Kerberos by using SOAP Input node
- You can configure Integration Bus to:
  - Operate as a client to a Kerberos secured service for message integrity, confidentiality, and authenticity
  - Operate as a Kerberos secured service for message integrity, confidentiality, and authenticity
- You can create multiple Kerberos configuration files and then use a separate Kerberos configuration file with each integration server

© Copyright IBM Corporation 2015

Figure 8-36. Kerberos-based WS-Security support WM6461.0

Notes:

Kerberos is a network authentication protocol that enables mutual authentication with symmetric keys. Users and services on a network authenticate with each other through a Key Distribution Center (KDC), as a trusted third party. Integration Bus supports Kerberos either as a service or as a client.

You can use message flows to call web services that are secured with Kerberos by using a SOAP Request node. You can also provide web services that are secured with Kerberos by using SOAP Input Nodes. The WS-Security header passes Kerberos tokens. You can sign and encrypt either parts or all of a SOAP message by using Kerberos tokens. Signing and encrypting messages provides message integrity, confidentiality, and authenticity.

WebSphere Education IBM

Policy sets

- *Policy set* is a container for the WS-Security policy type
- Policy set *binding* is associated with a policy set and contains information that is specific to the environment and operating system, such as information about keys
- Use policy sets and bindings to define authentication and encryption on SOAP messages
- Administered from IBM Integration Toolkit
  - Defined at the integration node level
  - Associated with a message flow, a node, or both in the BAR File editor
  - Any changes are saved directly to the associated integration node
  - Restart the message flow for the new configuration information to take effect

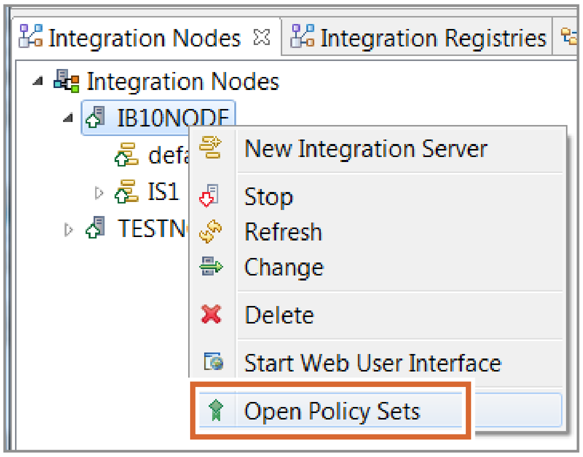


Figure 8-37. Policy sets WM6461.0

Notes:

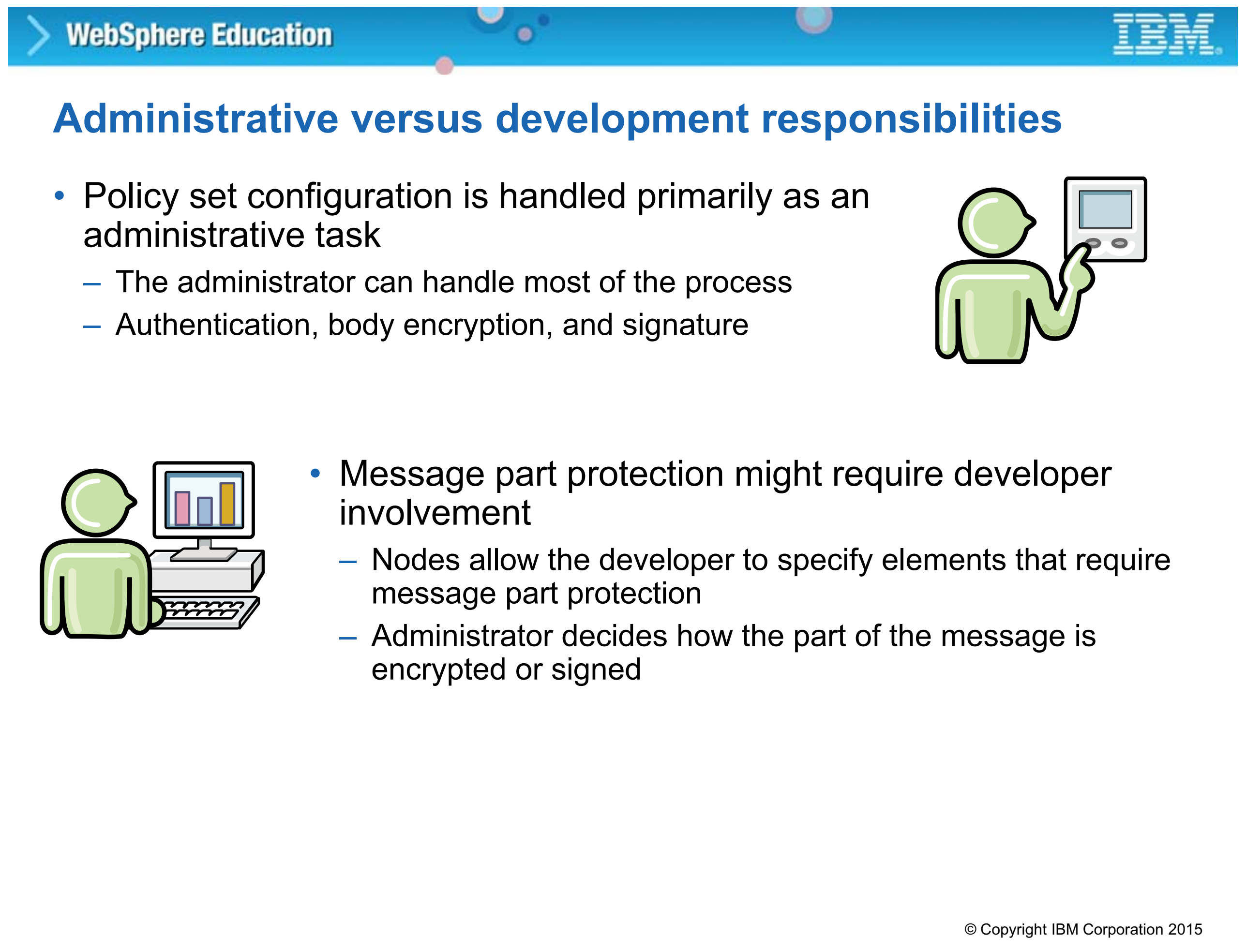
Policy sets and bindings define and configure WS-Security requirements, supported by Integration Bus, for the SOAP Input, SOAP Reply, SOAP Request, SOAP AsyncRequest, and SOAP AsyncResponse nodes.

A *policy set* is a container for the WS-Security policy type.

A *policy set binding* is associated with a policy set and contains information that is specific to the environment and operating system, such as information about keys.

Either the whole SOAP message body, or specific parts of the SOAP message header and body can be encrypted and signed.

You administer policy sets and bindings from the Integration Toolkit or by using the `mqsicreateconfigurableservice` command. In the Integration Toolkit, you can add, delete, display, and edit policy sets and bindings. Any changes to policy sets or bindings are saved directly to the associated integration node. You must stop and then restart the message flow for the new configuration information to take effect.



WebSphere Education

Administrative versus development responsibilities

- Policy set configuration is handled primarily as an administrative task
  - The administrator can handle most of the process
  - Authentication, body encryption, and signature
- Message part protection might require developer involvement
  - Nodes allow the developer to specify elements that require message part protection
  - Administrator decides how the part of the message is encrypted or signed

© Copyright IBM Corporation 2015

Figure 8-38. Administrative versus development responsibilities WM6461.0

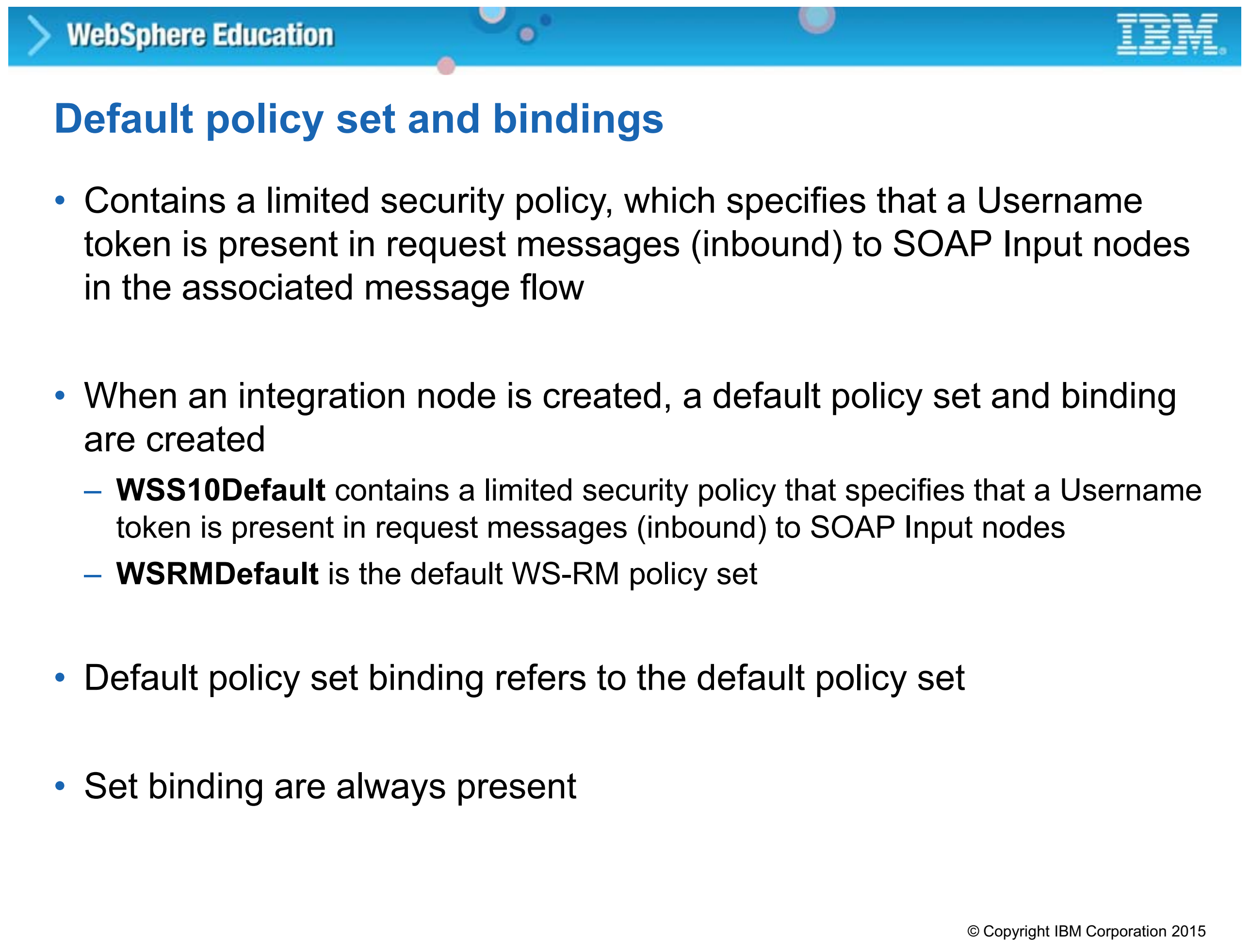
Notes:

The implementation of the security requirements for a particular application is split between the application developer and the system administrator.

Most of the implementation is done as an administration task. For message-level protection, the administrator can do all the necessary tasks to implement WS-Security. However, administrators might not be familiar with the message tree contained within the SOAP header and SOAP body. If you need to implement message part protection, you can specify which parts of the message should be protected.

With the SOAP nodes, a developer can specify elements that require the message part protection to be applied. The policy set editor allows the administrator to specify how each of these message parts is to be encrypted or signed.

Message processing nodes allow the developer to specify elements that require message part protection to be applied. The administrator decides how the message parts are to be encrypted or signed.



WebSphere Education IBM

Default policy set and bindings

- Contains a limited security policy, which specifies that a Username token is present in request messages (inbound) to SOAP Input nodes in the associated message flow
- When an integration node is created, a default policy set and binding are created
  - **WSS10Default** contains a limited security policy that specifies that a Username token is present in request messages (inbound) to SOAP Input nodes
  - **WSRMDefault** is the default WS-RM policy set
- Default policy set binding refers to the default policy set
- Set binding are always present

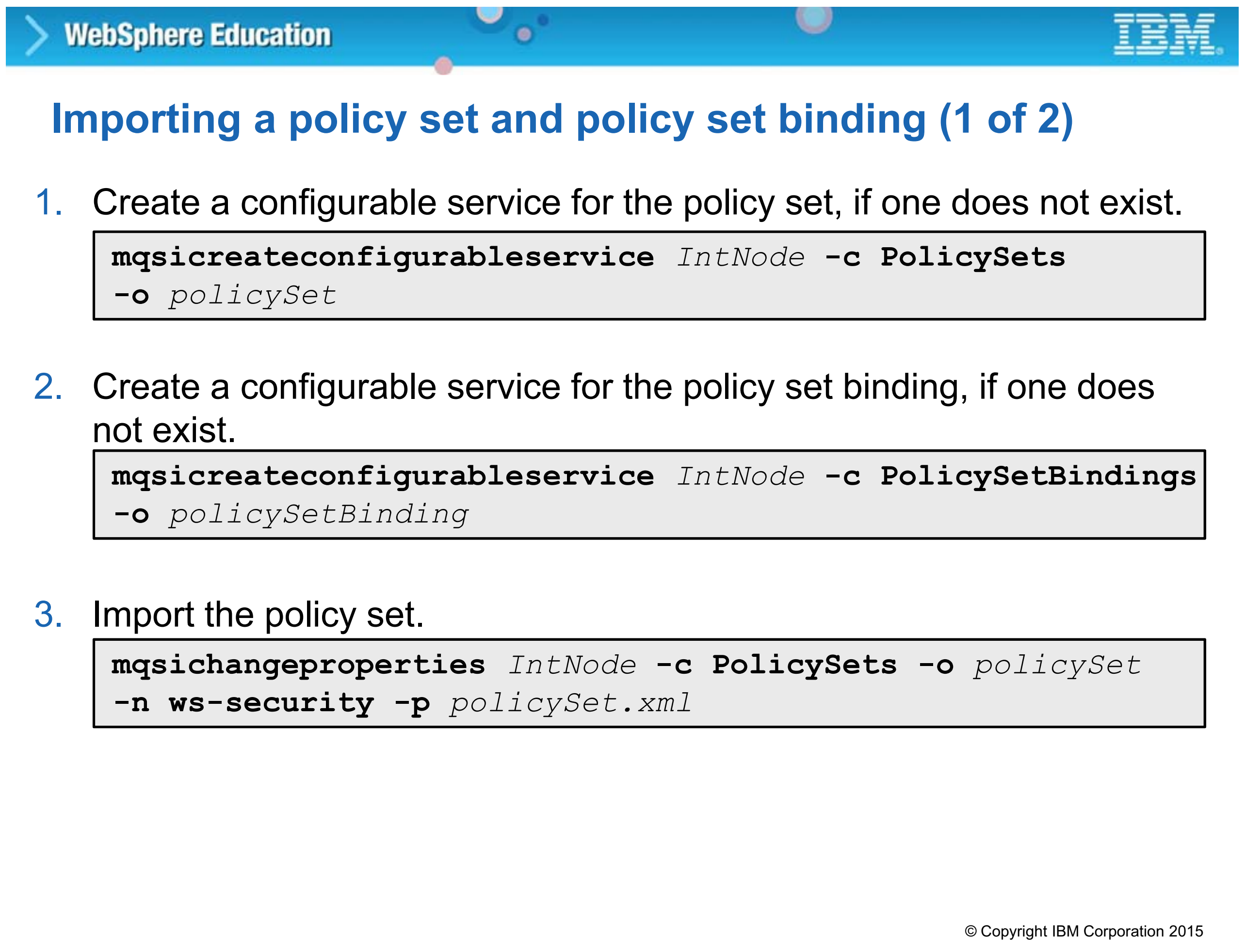
© Copyright IBM Corporation 2015

Figure 8-39. Default policy set and bindings WM6461.0

Notes:

IBM Integration Bus includes a default policy set and policy set binding, named WSS10Default, that contains a limited security policy.

The WSS10Default policy set and policy set binding cannot be modified.



WebSphere Education

Importing a policy set and policy set binding (1 of 2)

1. Create a configurable service for the policy set, if one does not exist.

```
mqsicreateconfigurableservice IntNode -c PolicySets
-o policySet
```

2. Create a configurable service for the policy set binding, if one does not exist.

```
mqsicreateconfigurableservice IntNode -c PolicySetBindings
-o policySetBinding
```

3. Import the policy set.

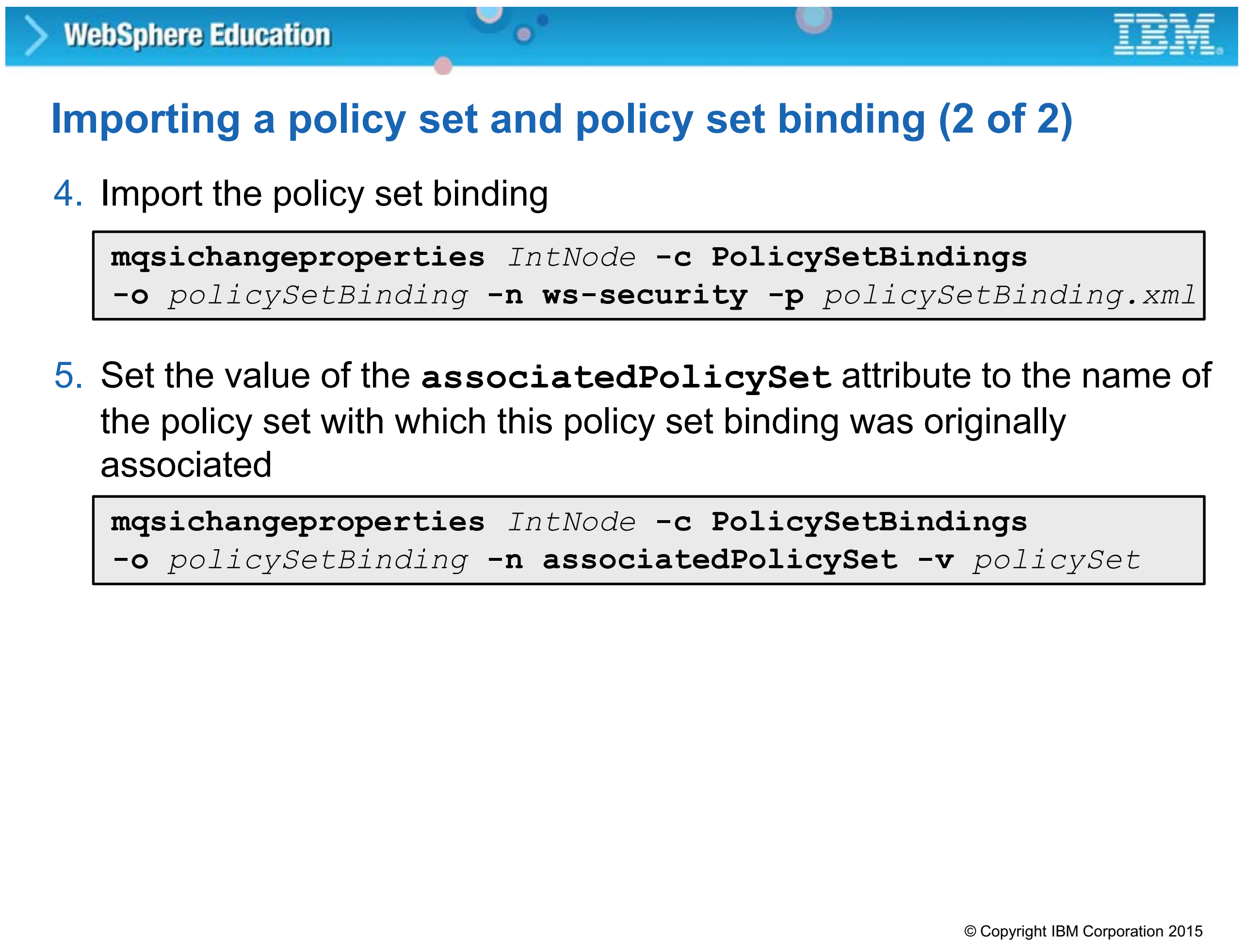
```
mqsichangeproperties IntNode -c PolicySets -o policySet
-n ws-security -p policySet.xml
```

Figure 8-40. Importing a policy set and policy set binding (1 of 2) WM6461.0

Notes:

This figure and the next show the steps for importing an existing policy set and policy set binding from the command.

The examples show how to import the policy set `policySet` to the integration node `inodeName` from a file called `policySet.xml`. The associated binding is `PolicySetBinding`, which is imported from `PolicySetBinding.xml`.



WebSphere Education

Importing a policy set and policy set binding (2 of 2)

4. Import the policy set binding

```
mqsichangeproperties IntNode -c PolicySetBindings
-o policySetBinding -n ws-security -p policySetBinding.xml
```

5. Set the value of the **associatedPolicySet** attribute to the name of the policy set with which this policy set binding was originally associated

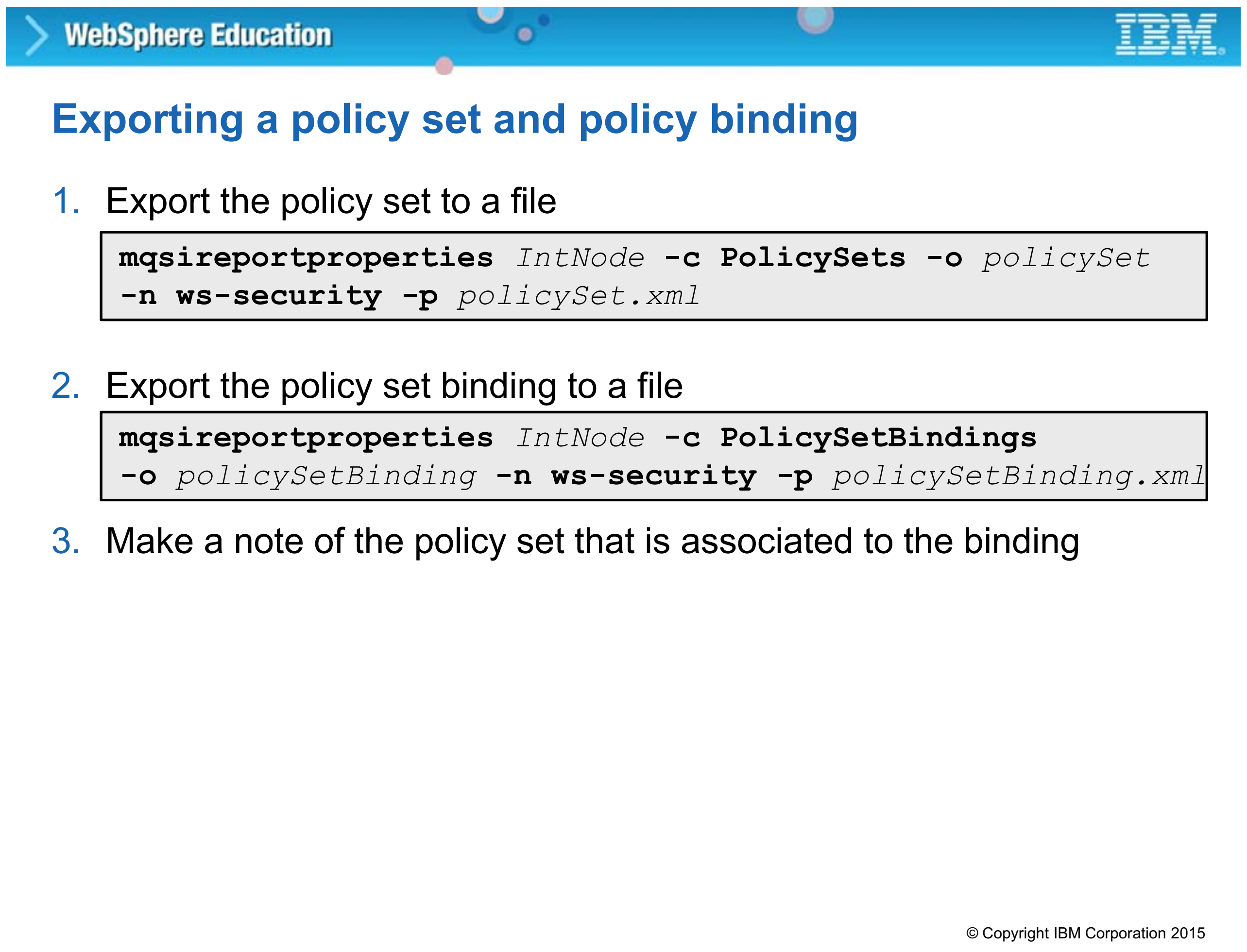
```
mqsichangeproperties IntNode -c PolicySetBindings
-o policySetBinding -n associatedPolicySet -v policySet
```

© Copyright IBM Corporation 2015

Figure 8-41. Importing a policy set and policy set binding (2 of 2) WM6461.0

Notes:

The next steps are to import the policy set binding and then associate the binding with the imported policy set.



WebSphere Education IBM.

Exporting a policy set and policy binding

1. Export the policy set to a file

```
mqsireportproperties IntNode -c PolicySets -o policySet
-n ws-security -p policySet.xml
```

2. Export the policy set binding to a file

```
mqsireportproperties IntNode -c PolicySetBindings
-o policySetBinding -n ws-security -p policySetBinding.xml
```

3. Make a note of the policy set that is associated to the binding

© Copyright IBM Corporation 2015

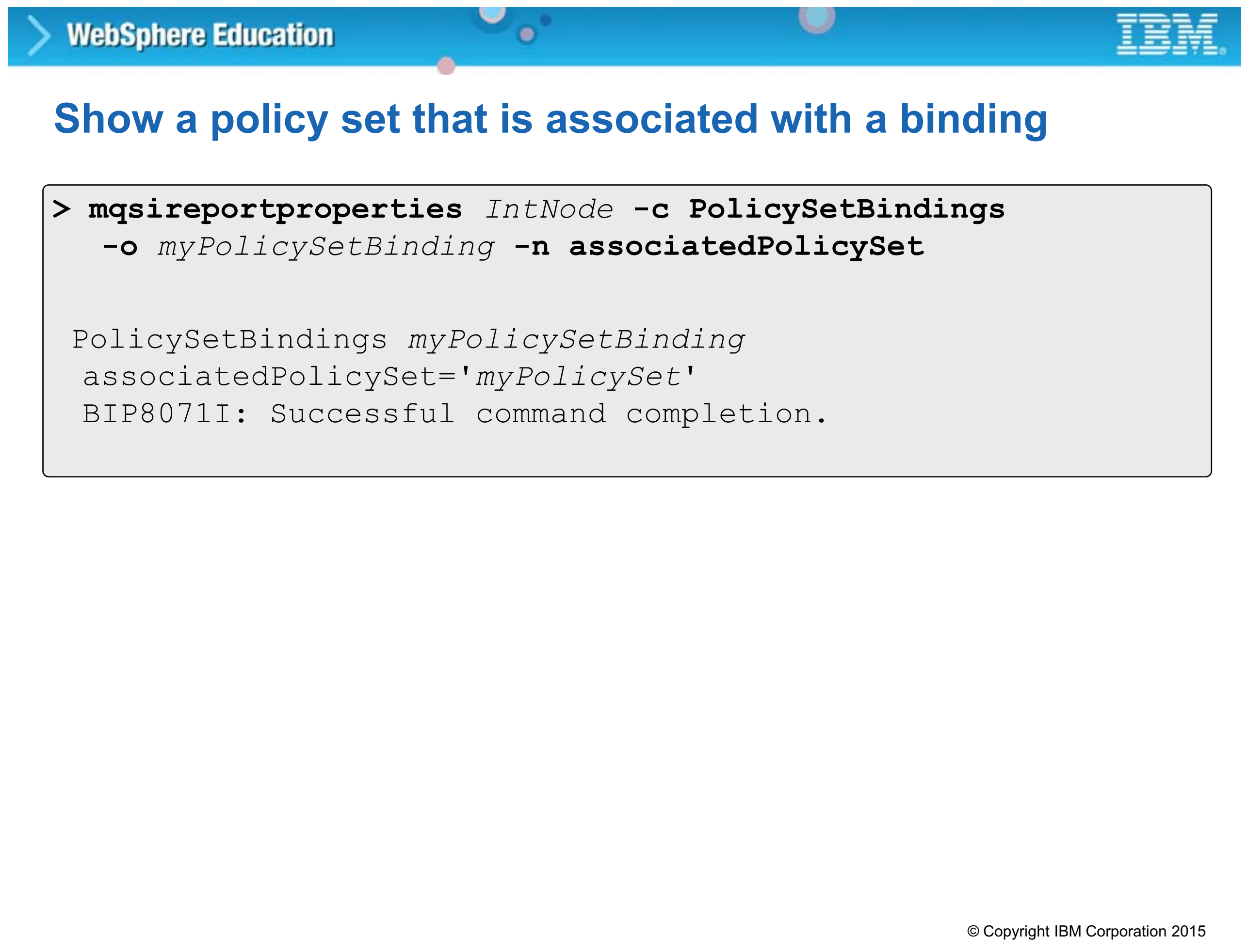
Figure 8-42. Exporting a policy set and policy binding WM6461.0

Notes:

Use the `mqsireportproperties` command to export a policy set and associated binding to a file.

The examples in the figure show how to export policy set `policySet` from an integration node `inodeName` to a file that is called `policySet.xml`. The associated binding is `policySetBinding`, which you export to `policySetBinding.xml`.

The final step is to make a note of the policy set that is associated with the binding. You need this information when you import the policy set and binding.



Show a policy set that is associated with a binding

```
> mqsireportproperties IntNode -c PolicySetBindings
   -o myPolicySetBinding -n associatedPolicySet


 PolicySetBindings myPolicySetBinding
  associatedPolicySet='myPolicySet'
  BIP8071I: Successful command completion.
```

Figure 8-43. Show a policy set that is associated with a binding WM6461.0

Notes:

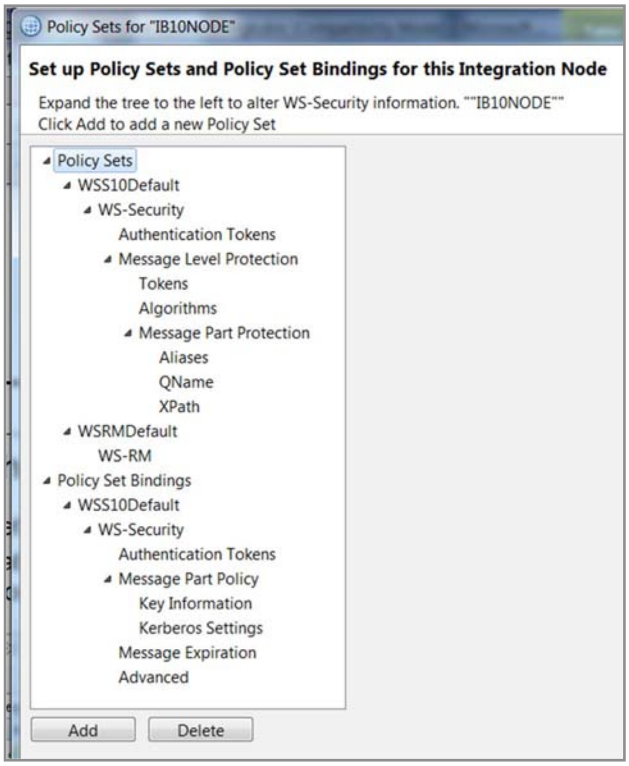
This `mqsireportproperties` command displays the policy set that is associated with a binding.

The example in the figure shows how to display the associated policy set for the policy set binding that is named `myPolicySetBinding` from the integration node named `myBroker`.

WebSphere Education IBM

Defining policy sets in IBM Integration Toolkit

- Policy Sets and Policy Set Bindings editor
  - For creating and editing a policy set or binding
  - Can define all the WS-Security policies for a single node, or a set of nodes
  - Associate the policy set with a message flow or a node in the BAR file editor



Policy Sets for "IB10NODE"

Set up Policy Sets and Policy Set Bindings for this Integration Node

Expand the tree to the left to alter WS-Security information. ""IB10NODE""
Click Add to add a new Policy Set

Figure 8-44. Defining policy sets in IBM Integration Toolkit WM6461.0

Notes:

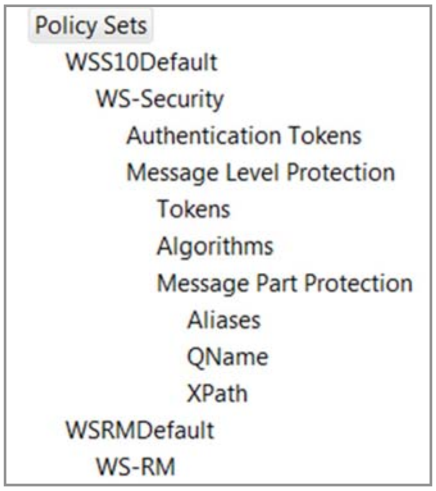
The Policy Sets and Policy Set Bindings editor is the default editor for editing, saving, importing, and exporting a policy set or binding. You can define all the WS-Security policies for a single node, or a set of nodes. The administrator can associate the policy set with either a message flow or a node.

Note

You must connect to an integration node before you can edit policy sets and bindings.

WebSphere Education IBM

Policy set editor



- **Authentication Tokens** to create Username, X.509, SAML, and LTPA authentication tokens
- **Message Level Protection** to apply signatures and encryption to the whole message, whether inbound or outbound
- **Message Part Protection** to define the parts of a message that encryption and signature apply to
  – **Aliases** to refer to an alias identified in a SOAPInput, SOAPRequest, or SOAPAsyncRequest node
  – **XPath** to define an expression that refers to an element in the message to which encryption or signature applies

Figure 8-45. Policy set editor WM6461.0

Notes:

The Policy set editor contains panes for defining authentication tokens, message level protection, and message part protection.

To create a policy set, select **Policy Sets** and then click **Add**. A policy set is added with a default name, which can be changed.

Policy set binding editor

Associate a policy set binding with a policy set

- **Authentication Tokens** to further configure any X.509 authentication tokens that are defined in the associated policy set
- **Message Part Policy** to further configure any message part protection tokens that are defined in the associated policy set
- **Key Information** to further configure any message level protection tokens that are defined in the associated policy set
- **Kerberos Settings** to further configure any Kerberos tokens that are defined in the associated policy set
- **Message Expiration** to define settings to expire message after the specified time interval

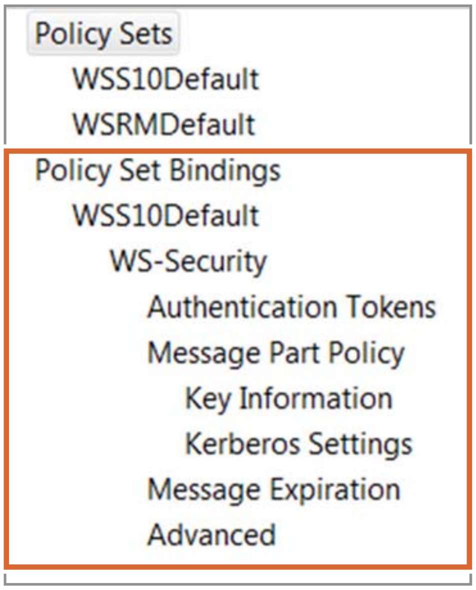


Figure 8-46. Policy set binding editor WM6461.0

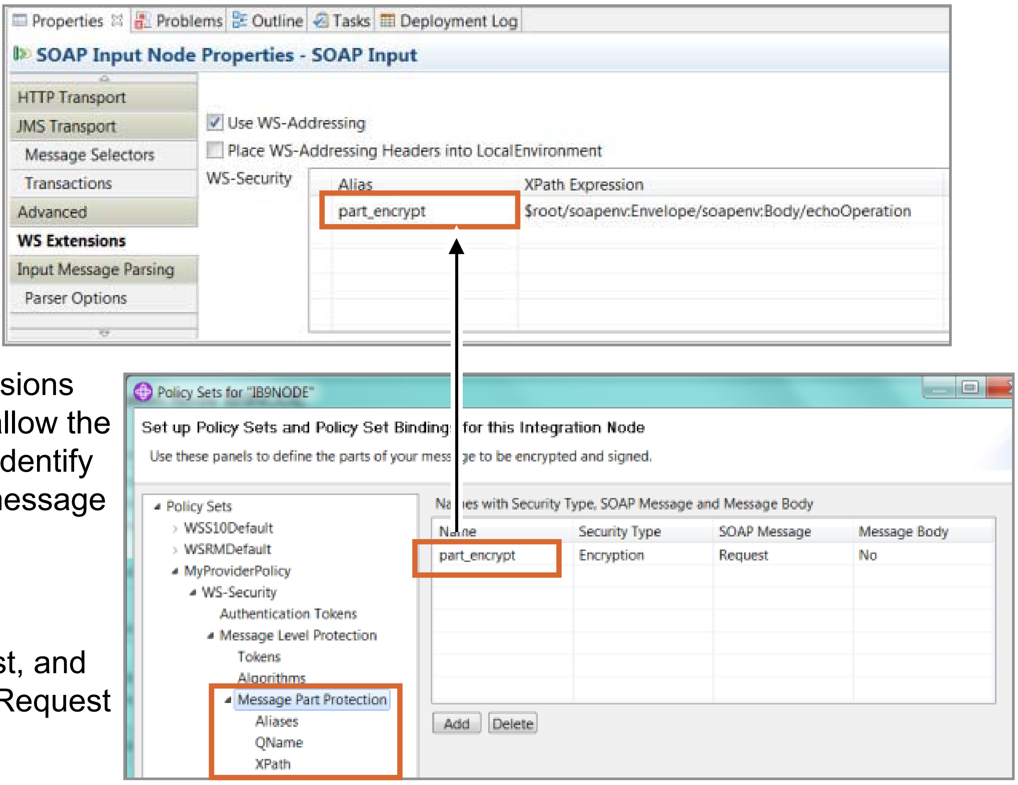
Notes:

The Policy Set Binding editor is used to create and maintain policy set bindings. Part of the definition includes associating a policy set binding with a policy set when the policy set binding is added. Information about the binding is then displayed on the main pane of the policy set binding.

You cannot change the policy set to which a binding is associated. You can delete the policy set binding independently of the policy set. However, deleting a policy set also deletes the associated policy set binding.

To create a policy set binding, select **Policy Set Binding** and then click **Add**. A policy set binding is added with a default name, which can be changed.

Defining message part protection on a SOAP node



[✓] Use WS-Addressing

[ ] Place WS-Addressing Headers into LocalEnvironment

WS-Security

Alias	XPath Expression
part_encrypt	$root/soapenv:Envelope/soapenv:Body/echoOperation

Set up Policy Sets and Policy Set Bindings for this Integration Node

Use these panels to define the parts of your message to be encrypted and signed.

Names with Security Type, SOAP Message and Message Body

Name	Security Type	SOAP Message	Message Body
part_encrypt	Encryption	Request	No

- XPath expressions on the node allow the developer to identify parts of the message to protect
- Provided on SOAPInput, SOAPRequest, and SOAPAsync-Request nodes

Figure 8-47. Defining message part protection on a SOAP node WM6461.0

Notes:

This figure shows how the properties of the SOAP nodes are used to specify security on parts of the SOAP message. On the SOAP node **WS Extensions** property tab, an alias is added for each message part that is subject to security. This alias can be the contained in the header of the message or the body of the message. You use an XPath expression to specify the parts of the message that must be protected.

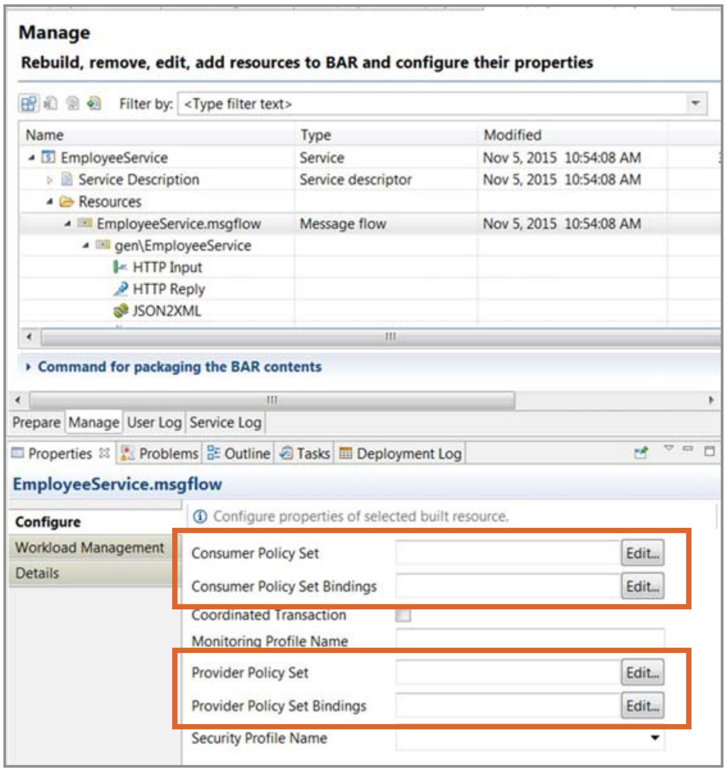
The alias provides a link between the configuration in the SOAP nodes and the runtime components that the system administrator specifies. The Integration Bus administrator defines the Policy Set, which resolves the alias to either encrypt or sign the part of the message that the XPath expression references. Specifying the policy set is an administrative function. You cannot define or reference it with the SOAP node properties.

You can also use the LocalEnvironment to specify parts of the message that must be signed or encrypted. This capability can be used to change the security requirement dynamically (based on the message, for example), for different types of messages. This technique still uses the alias to link the XPath expression that is defined with the policy set.

WebSphere Education

Assigning policy sets and bindings in the BAR file

- You can set the policy set and bindings at the message flow level and the integration node level
  - Select the message flow on **Manage** tab in the BAR file editor, to set the consumer and provider policy sets and bindings
  - Select a SOAP Input, SOAP Request, or SOAP AsyncRequest node on the **Manage** tab to assign the policy set and binding
  - Click **Edit** to select a policy set and policy set binding that you defined by using the Policy Set editor and the Policy Set Binding editor



Manage

Rebuild, remove, edit, add resources to BAR and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
EmployeeService	Service	Nov 5, 2015 10:54:08 AM
Service Description	Service descriptor	Nov 5, 2015 10:54:08 AM
Resources		
EmployeeService.msgflow	Message flow	Nov 5, 2015 10:54:08 AM
gen\EmployeeService		
HTTP Input		
HTTP Reply		
JSON2XML		

Command for packaging the BAR contents

Prepare Manage User Log Service Log
Properties Problems Outline Tasks Deployment Log

EmployeeService.msgflow

Configure | Workload Management | Details

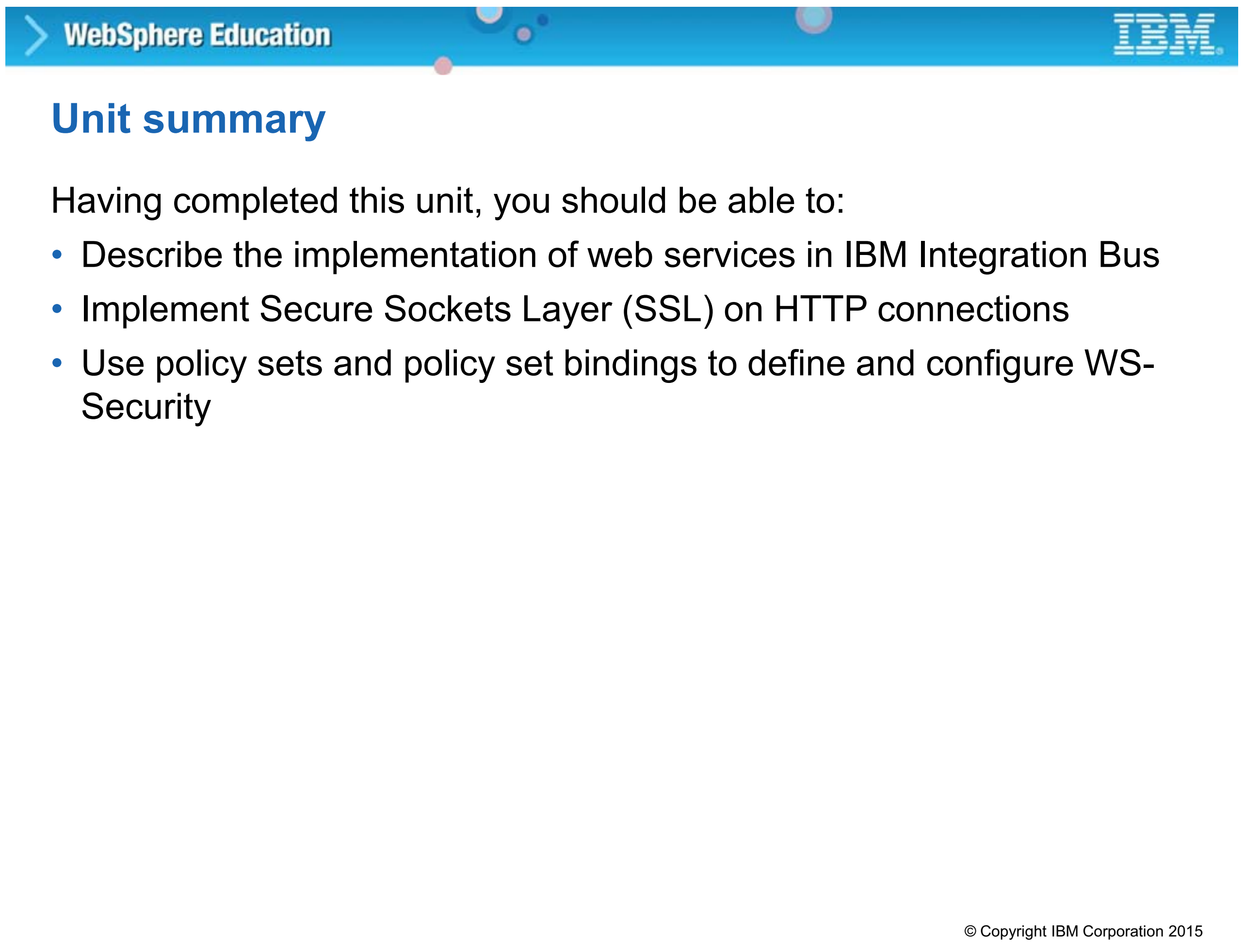
Configure properties of selected built resource.

Consumer Policy Set [Edit...]
Consumer Policy Set Bindings [Edit...]
Coordinated Transaction
Monitoring Profile Name
Provider Policy Set [Edit...]
Provider Policy Set Bindings [Edit...]
Security Profile Name

© Copyright IBM Corporation 2015

Figure 8-48. Assigning policy sets and bindings in the BAR file WM6461.0

Notes:



WebSphere Education

Unit summary

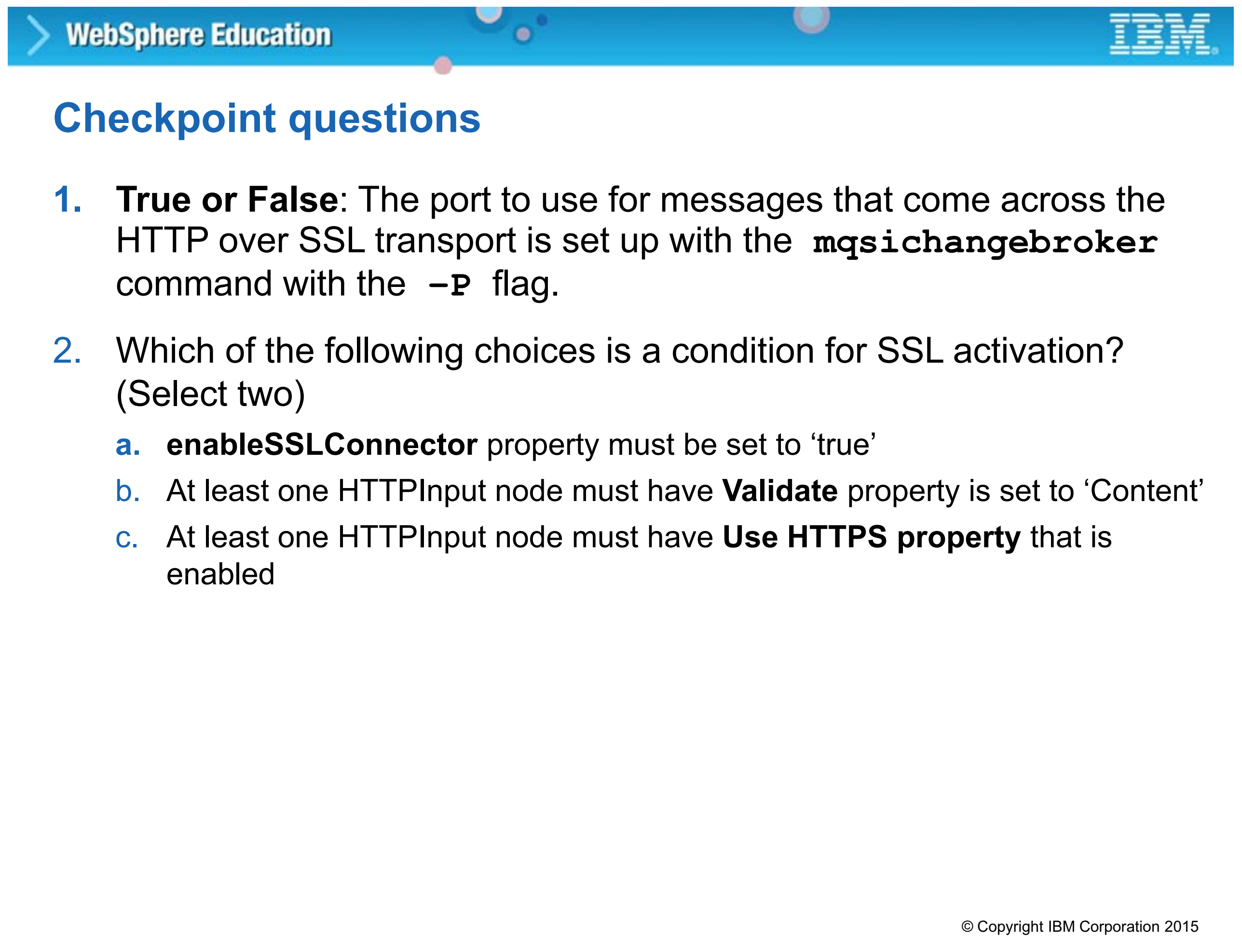
Having completed this unit, you should be able to:

- Describe the implementation of web services in IBM Integration Bus
- Implement Secure Sockets Layer (SSL) on HTTP connections
- Use policy sets and policy set bindings to define and configure WS-Security

© Copyright IBM Corporation 2015

Figure 8-49. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. **True or False**: The port to use for messages that come across the HTTP over SSL transport is set up with the `mqsichangebroker` command with the `–P` flag.
2. Which of the following choices is a condition for SSL activation? (Select two)
   a. **enableSSLConnector** property must be set to ‘true’
   b. At least one HTTPInput node must have **Validate** property is set to ‘Content’
   c. At least one HTTPInput node must have **Use HTTPS property** that is enabled

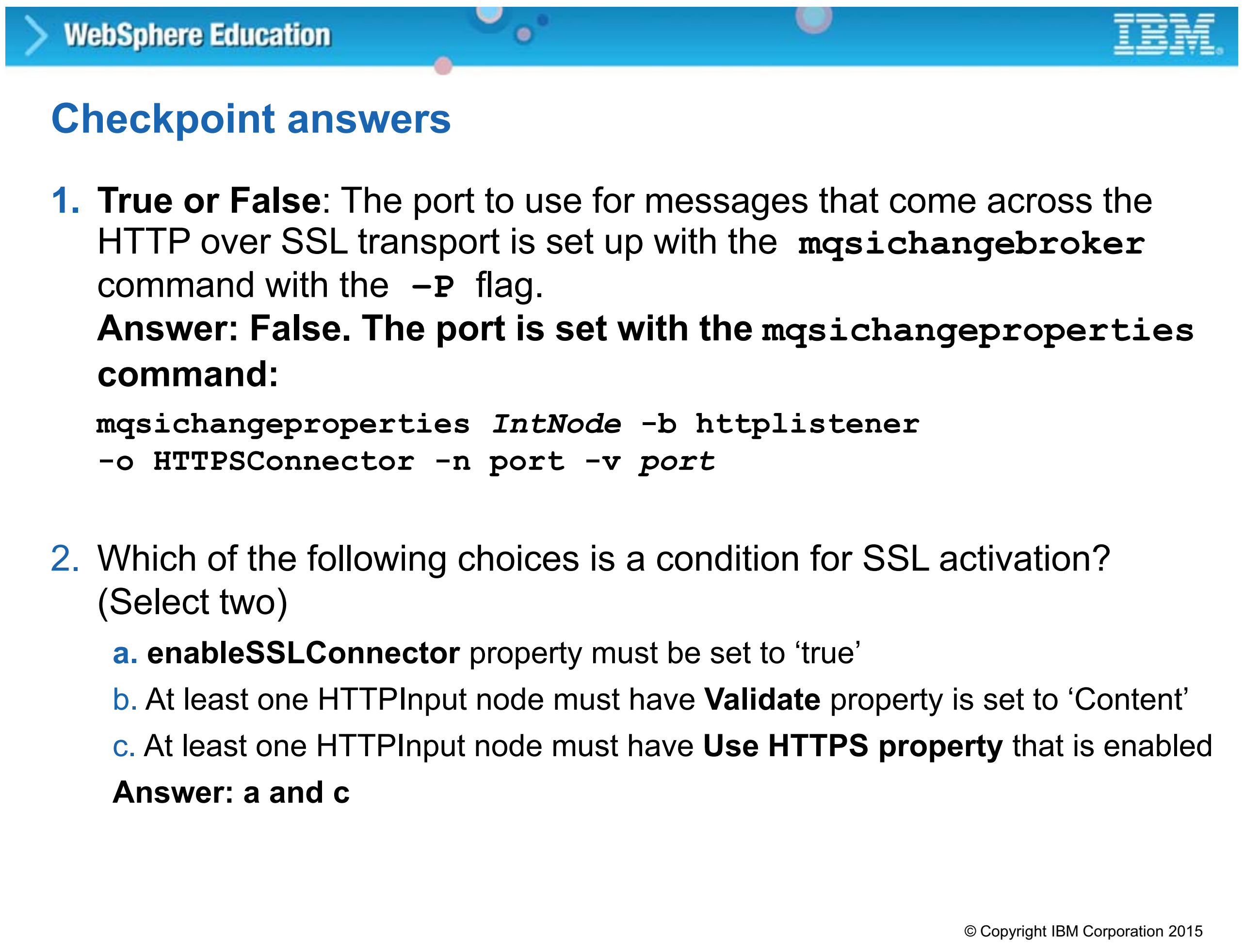
Figure 8-50. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education

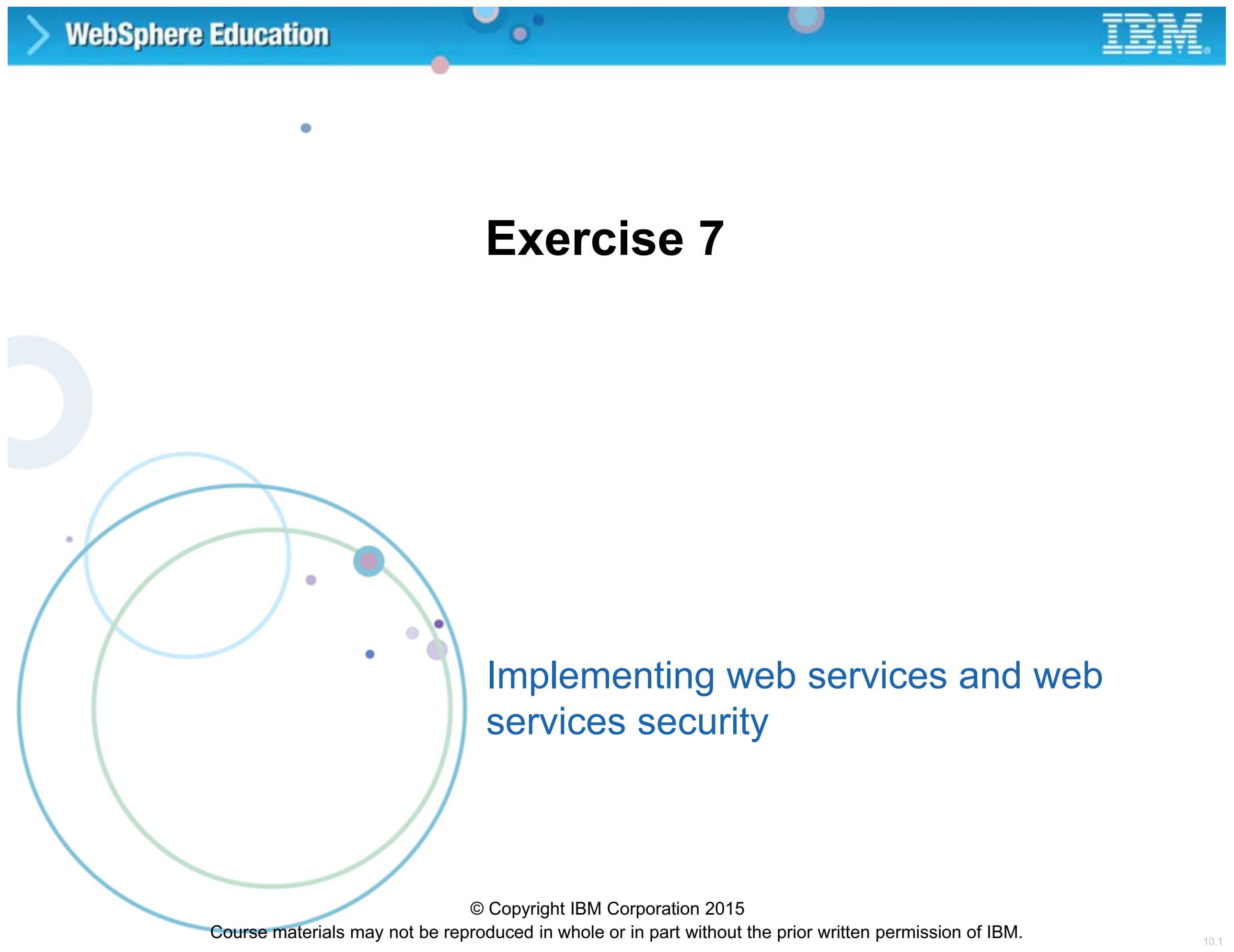
Checkpoint answers

1. **True or False**: The port to use for messages that come across the HTTP over SSL transport is set up with the **`mqsichangebroker`** command with the **`–P`** flag.
   Answer: False. The port is set with the `mqsichangeproperties` command:
   ```
   mqsichangeproperties IntNode -b httplistener
   -o HTTPSConnector -n port -v port
   ```

2. Which of the following choices is a condition for SSL activation? (Select two)
   a. **enableSSLConnector** property must be set to 'true'
   b. At least one HTTPInput node must have **Validate** property is set to 'Content'
   c. At least one HTTPInput node must have **Use HTTPS property** that is enabled
   Answer: a and c

Figure 8-51. Checkpoint answers WM6461.0

Notes:



WebSphere Education

IBM

Exercise 7

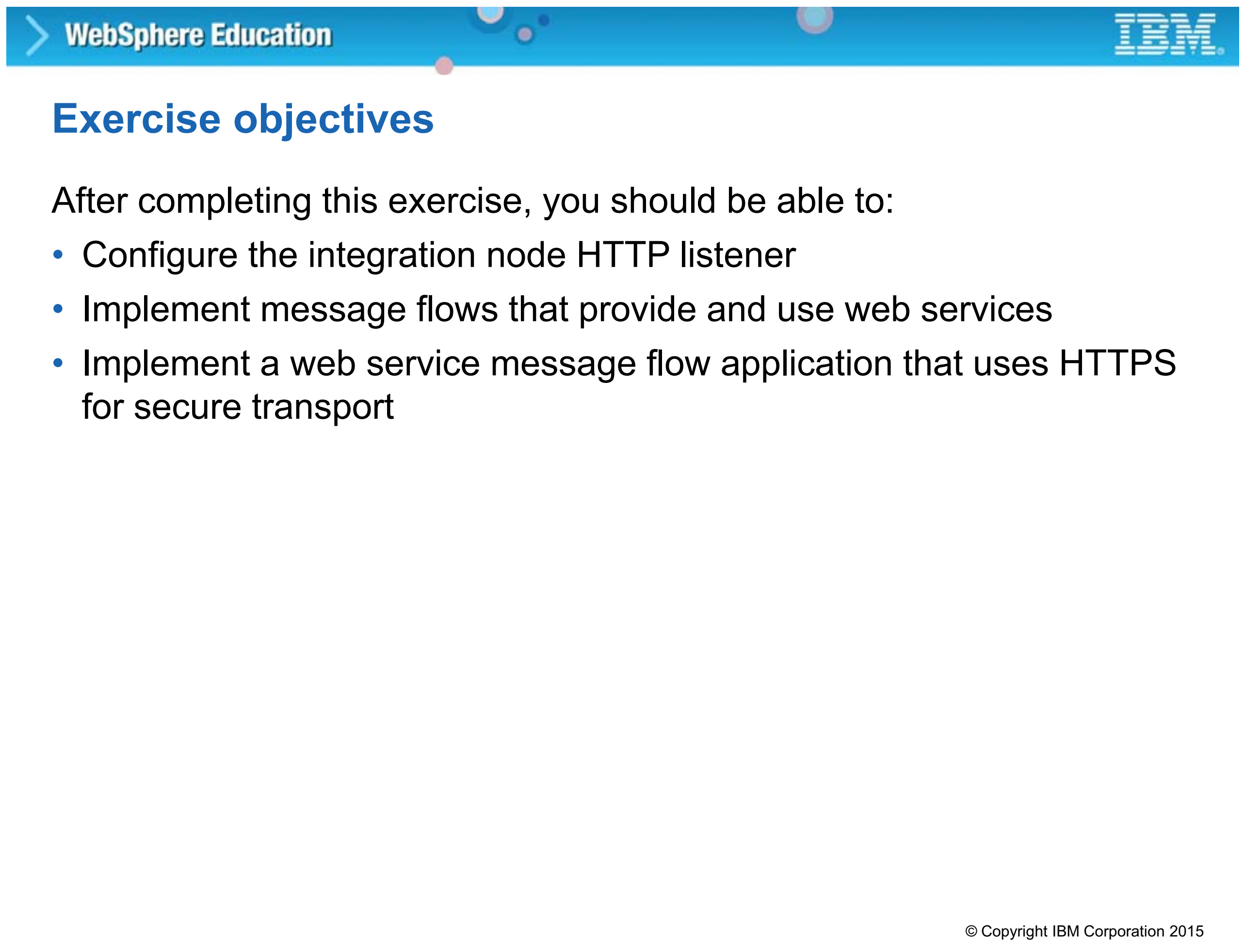
Implementing web services and web services security

10.1

Figure 8-52. Exercise 7 WM6461.0

Notes:

In this exercise, you configure the integration node runtime environment to support web services. You use the supplied tools to set up a web service that can accept information from, or deliver it to, a transport with secure HTTP (HTTPS).



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Configure the integration node HTTP listener
- Implement message flows that provide and use web services
- Implement a web service message flow application that uses HTTPS for secure transport

Figure 8-53. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 9. Diagnosing problems

What this unit is about

This unit introduces the tools and approaches you use for problem diagnosis in IBM Integration Bus.

What you should be able to do

After completing this unit, you should be able to:

- Predict the location of the message if a runtime error is encountered during message flow processing
- Configure message flows for transactional behavior
- Set the Flow Debug Port for an integration server
- Enable and disable a user trace
- Enable and disable a service trace
- Enable and disable Trace nodes in a message flow
- Diagnose common problems in the IBM Integration Bus environment

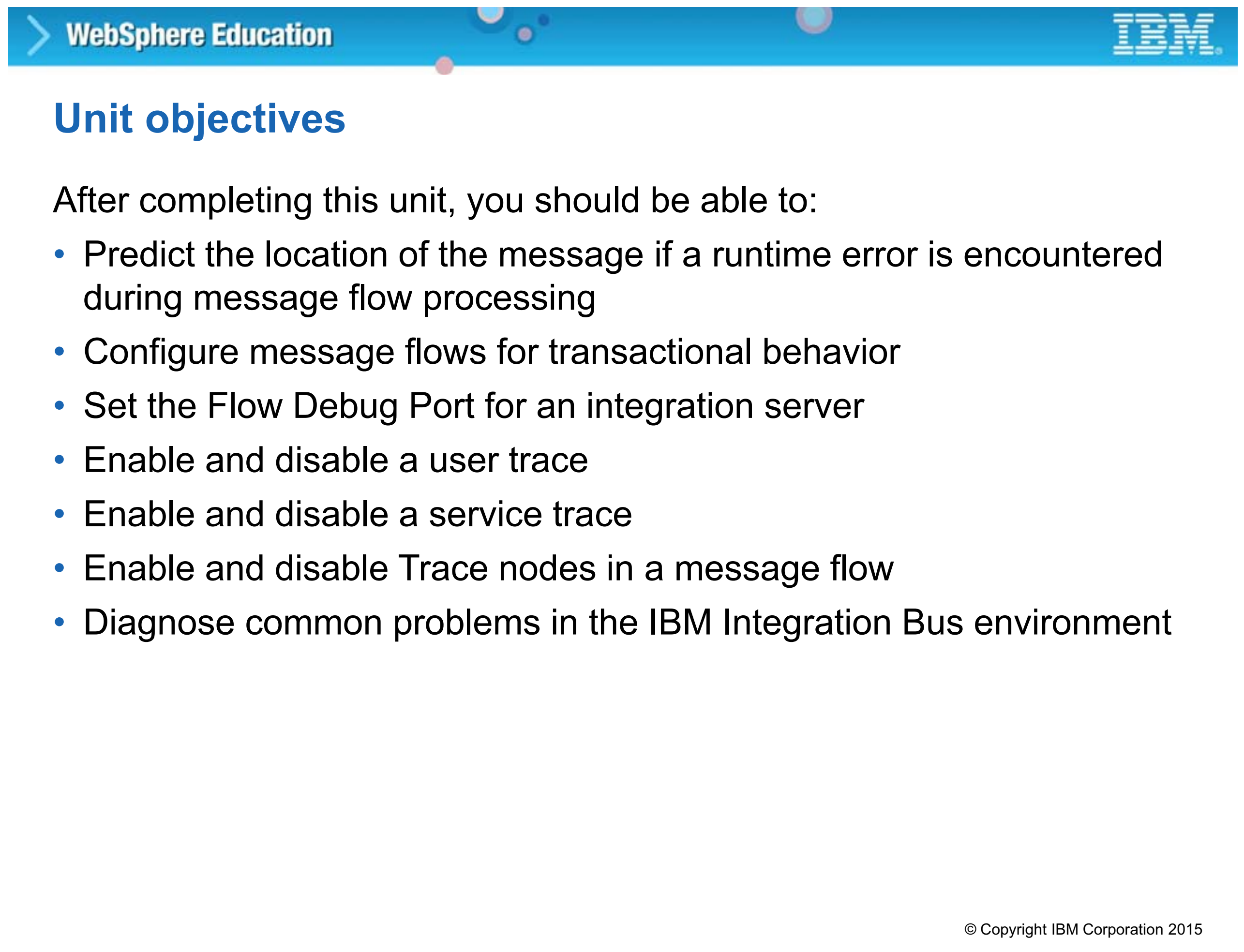
How you will check your progress

Checkpoints

Lab exercises

References

IBM Knowledge Center for IBM Integration Bus V10



Unit objectives

After completing this unit, you should be able to:

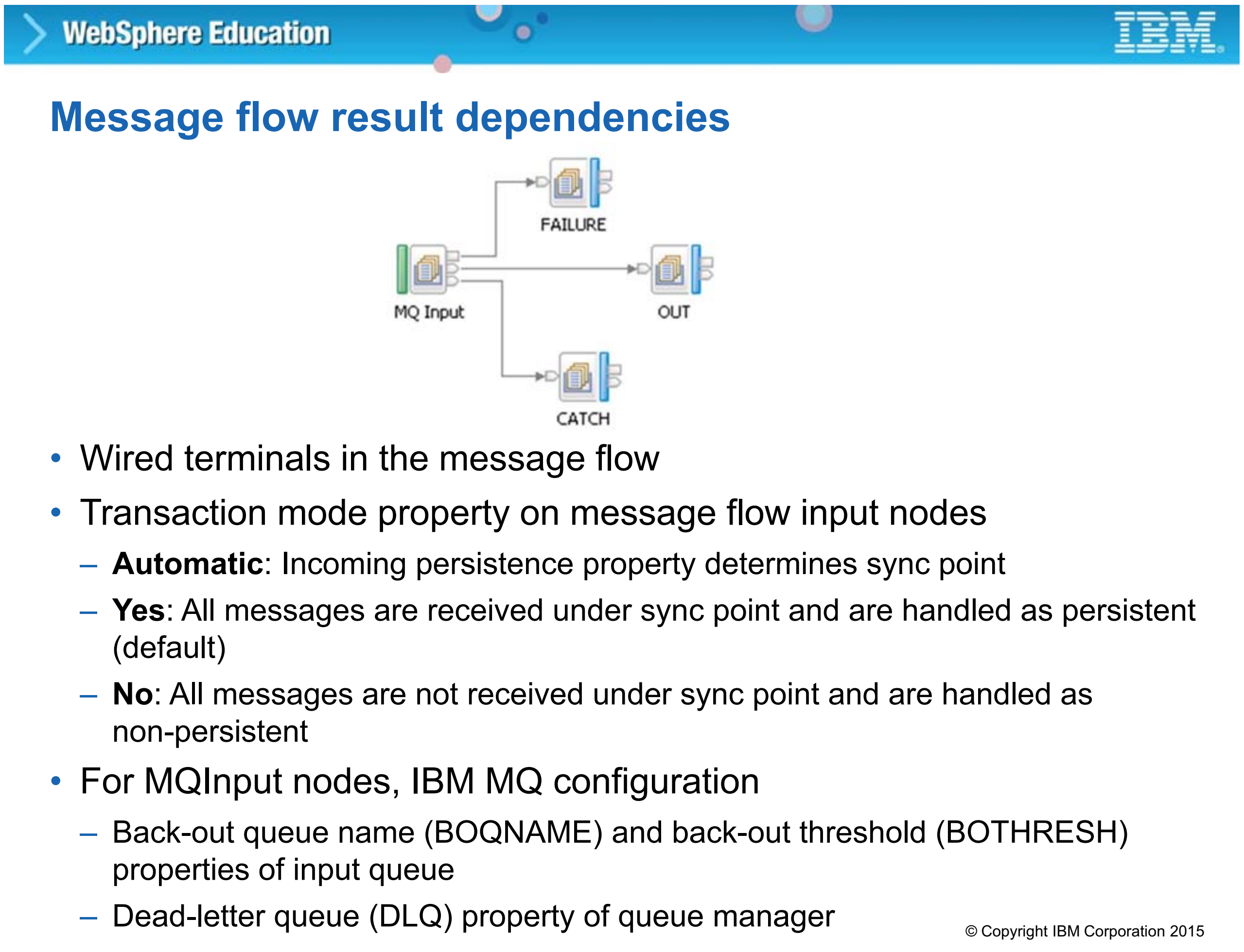
- Predict the location of the message if a runtime error is encountered during message flow processing
- Configure message flows for transactional behavior
- Set the Flow Debug Port for an integration server
- Enable and disable a user trace
- Enable and disable a service trace
- Enable and disable Trace nodes in a message flow
- Diagnose common problems in the IBM Integration Bus environment

© Copyright IBM Corporation 2015

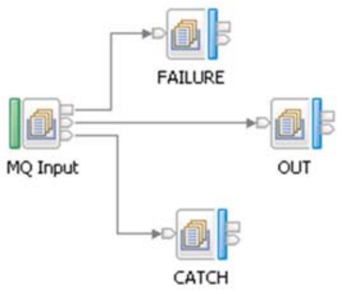
Figure 9-1. Unit objectives WM6461.0

Notes:

9.1. Integration node default behavior in case of runtime errors



Message flow result dependencies



- Wired terminals in the message flow
- Transaction mode property on message flow input nodes
  - **Automatic**: Incoming persistence property determines sync point
  - **Yes**: All messages are received under sync point and are handled as persistent (default)
  - **No**: All messages are not received under sync point and are handled as non-persistent
- For MQInput nodes, IBM MQ configuration
  - Back-out queue name (BOQNAME) and back-out threshold (BOTHRESH) properties of input queue
  - Dead-letter queue (DLQ) property of queue manager

Figure 9-2. Message flow result dependencies WM6461.0

Notes:

If a message flow completes successfully, messages almost always end up on the output as designed. Many factors can affect the path that is taken if a runtime error occurs.

The first factor is the wiring of the node terminals in the message flow. If there is a runtime error and other terminals, such as the **Catch** and **Failure** terminal, are connected, the message has an opportunity to go to many different destinations.

Most input nodes, such as the MQInput node, include a property for setting the transaction mode.

- If the transaction mode is set to **No**, and an error is encountered in the message flow, even a persistent message might be discarded.
- With transaction mode set to **Yes**, all messages are rolled back if an error occurs. This rollback processing might be inefficient, given the volatile nature of messages that are not persistent. This mode is the default mode.
- The **Automatic** setting affords transaction protection for persistent messages while denying the same for messages that are not persistent.

The final factor that affects the message path is the type of input nodes in the message flow. For example, if the message flow contains an MQInput node, the properties of the input queue and configuration of the queue manager affect the message path.

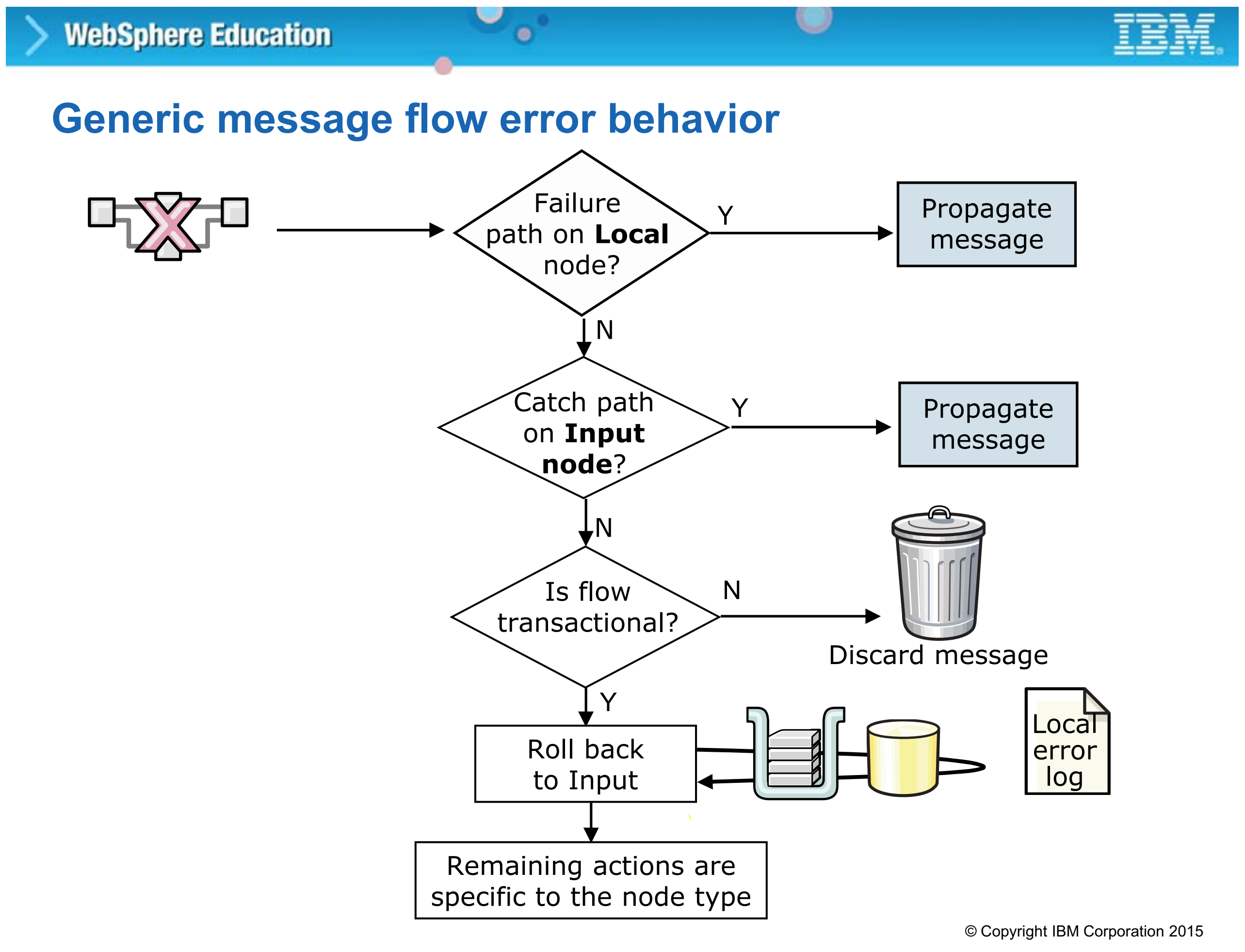

WebSphere Education
IBM
Generic message flow error behavior
Failure path on **Local** node?
Y
Propagate message
N
Catch path on **Input node**?
Y
Propagate message
N
Is flow transactional?
N
Discard message
Y
Roll back to Input
Local error log
Remaining actions are specific to the node type
© Copyright IBM Corporation 2015

Figure 9-3. Generic message flow error behavior WM6461.0

Notes:

The figure shows the events that occur when an error occurs in a message flow processing node.

If an error causes a message to roll back, the message is first directed to the **Failure** terminal of the node where the exception occurred. If the local **Failure** terminal is not connected, the message rolls back to the message flow input node.

At the message flow input node, the message is first directed to the input node **Catch** terminal for disposition. If another error occurs, the rollback continues, and the message is directed to the input node **Failure** terminal for disposition. If the **Catch** terminal is not wired, the flow transaction is checked.

Transactional messages are rolled back. The remaining actions are specific to the input node type.

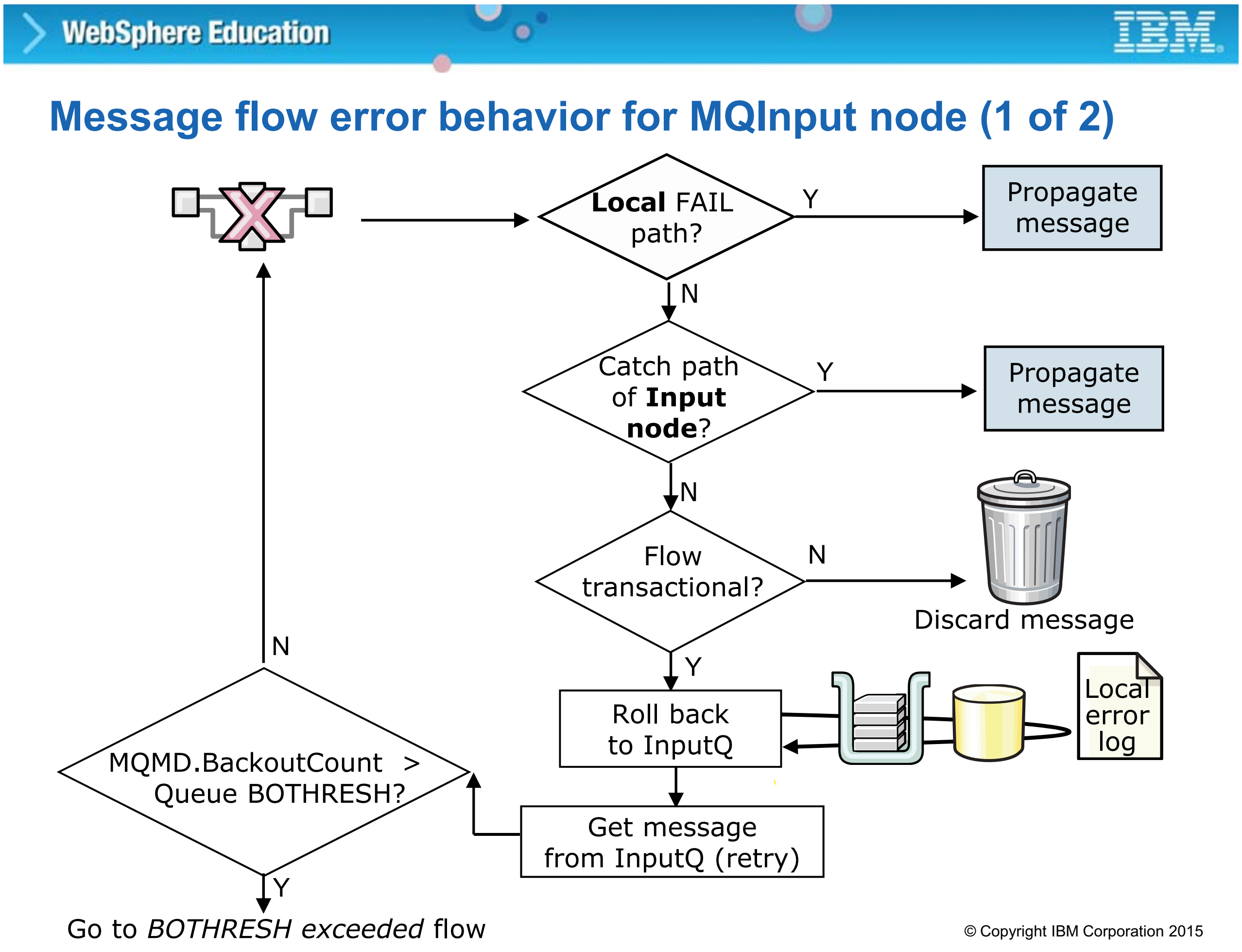

WebSphere Education
IBM
Message flow error behavior for MQInput node (1 of 2)
Local FAIL path?
Y
Propagate message
N
Catch path of Input node?
Y
Propagate message
N
Flow transactional?
N
Discard message
Y
Roll back to InputQ
Local error log
Get message from InputQ (retry)
MQMD.BackoutCount > Queue BOTHRESH?
N
Y
Go to BOTHRESH exceeded flow
© Copyright IBM Corporation 2015

Figure 9-4. Message flow error behavior for MQInput node (1 of 2) WM6461.0

Notes:

The figure shows the events that occur when an error occurs in a message flow processing node and the input node is an MQInput node.

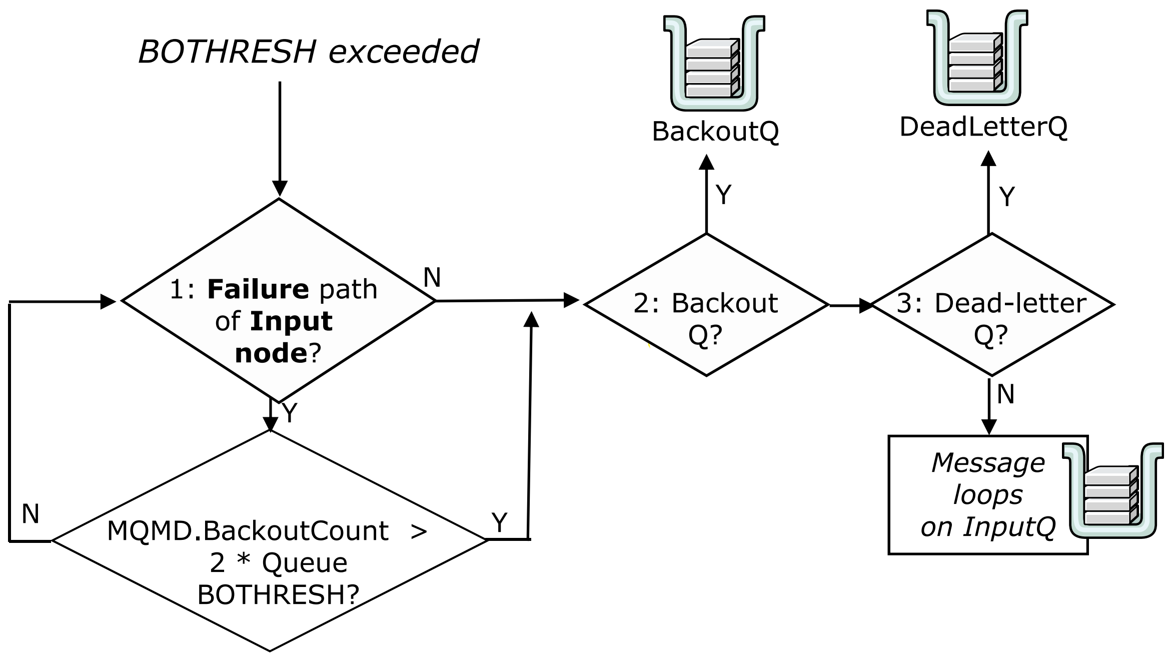
If an error causes a message to roll back, the message is first directed to the **Failure** terminal of the node where the exception occurred. If the local **Failure** terminal is not connected, the message rolls back to the message flow input node.

At the message flow input node, the message is first directed to the input node **Catch** terminal for disposition. If another error occurs, the rollback continues, and the message is directed to the input node **Failure** terminal for disposition. If the **Catch** terminal is not wired, the flow transaction is checked.

Transactional messages are rolled back to the input queue, and the message flow restarts. The retry attempt might not work and, if so, would end with another backout.

A backout threshold (BOTHRESH) property on the input queue limits the number of retry attempts. The next page shows the behavior that occurs when the BOTHRESH is exceeded.

Message flow error behavior for MQInput node (2 of 2)



© Copyright IBM Corporation 2015

Figure 9-5. Message flow error behavior for MQInput node (2 of 2) WM6461.0

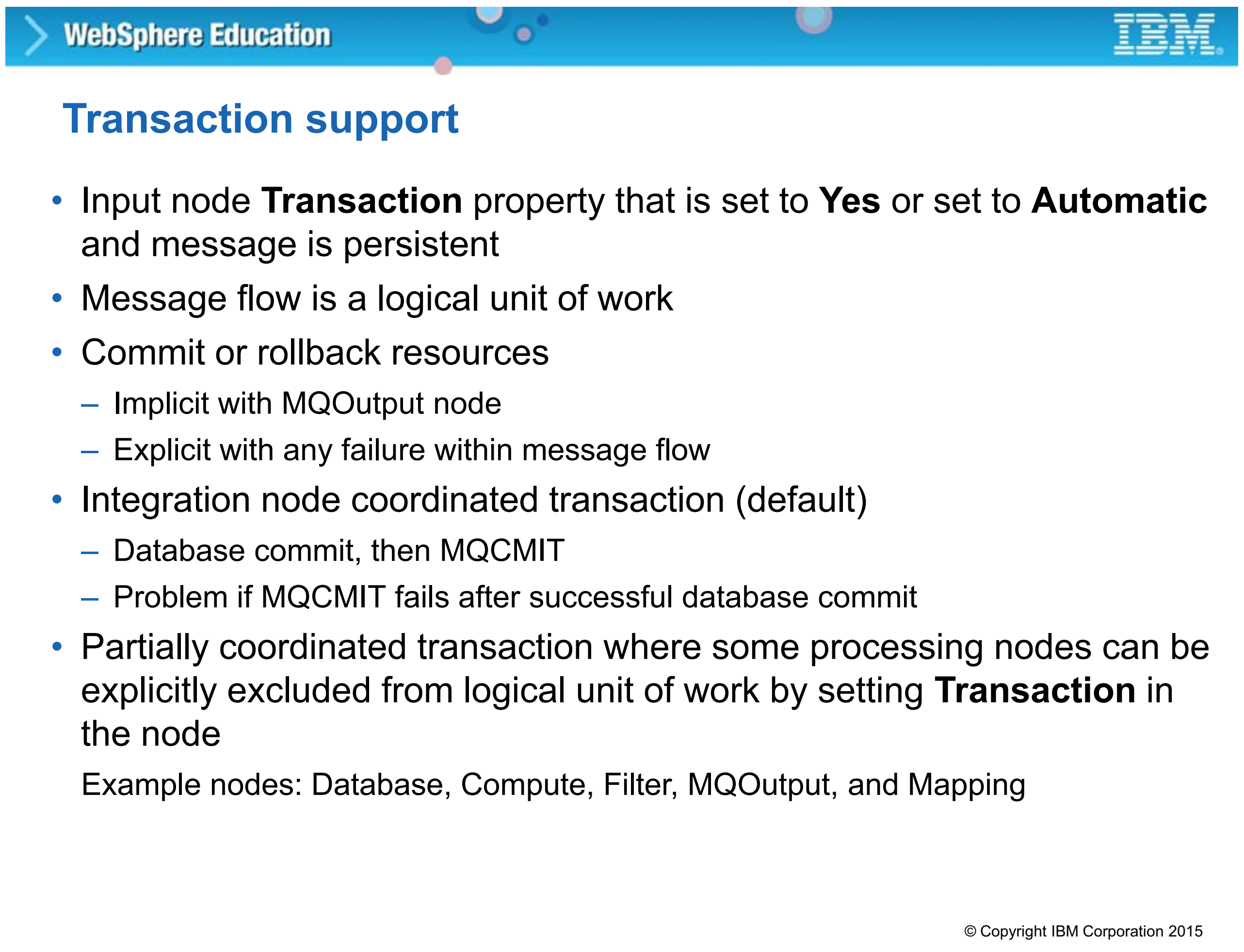
Notes:

When the input queue backout threshold value is reached, the message destination depends on several factors:

- The existence of the **Failure** path of an input node
- The backout queue (BOQ) of the input queue
- The dead-letter queue of the queue manager

These paths are checked in exactly this order.

If a failure occurs beyond the **Failure** terminal, further attempts to process the message occur until the backout count in the MQMD is twice the backout threshold set for the input queue. When this limit is reached, the node attempts to put the message to either the backout queue or the dead-letter queue. If neither of these queues exist, or if the message cannot be written to them, the message loops on the input queue, and prevents any new messages from being processed. In these cases, the error condition must be corrected manually.



WebSphere Education

Transaction support

- Input node **Transaction** property that is set to **Yes** or set to **Automatic** and message is persistent
- Message flow is a logical unit of work
- Commit or rollback resources
  - Implicit with MQOutput node
  - Explicit with any failure within message flow
- Integration node coordinated transaction (default)
  - Database commit, then MQCMIT
  - Problem if MQCMIT fails after successful database commit
- Partially coordinated transaction where some processing nodes can be explicitly excluded from logical unit of work by setting **Transaction** in the node

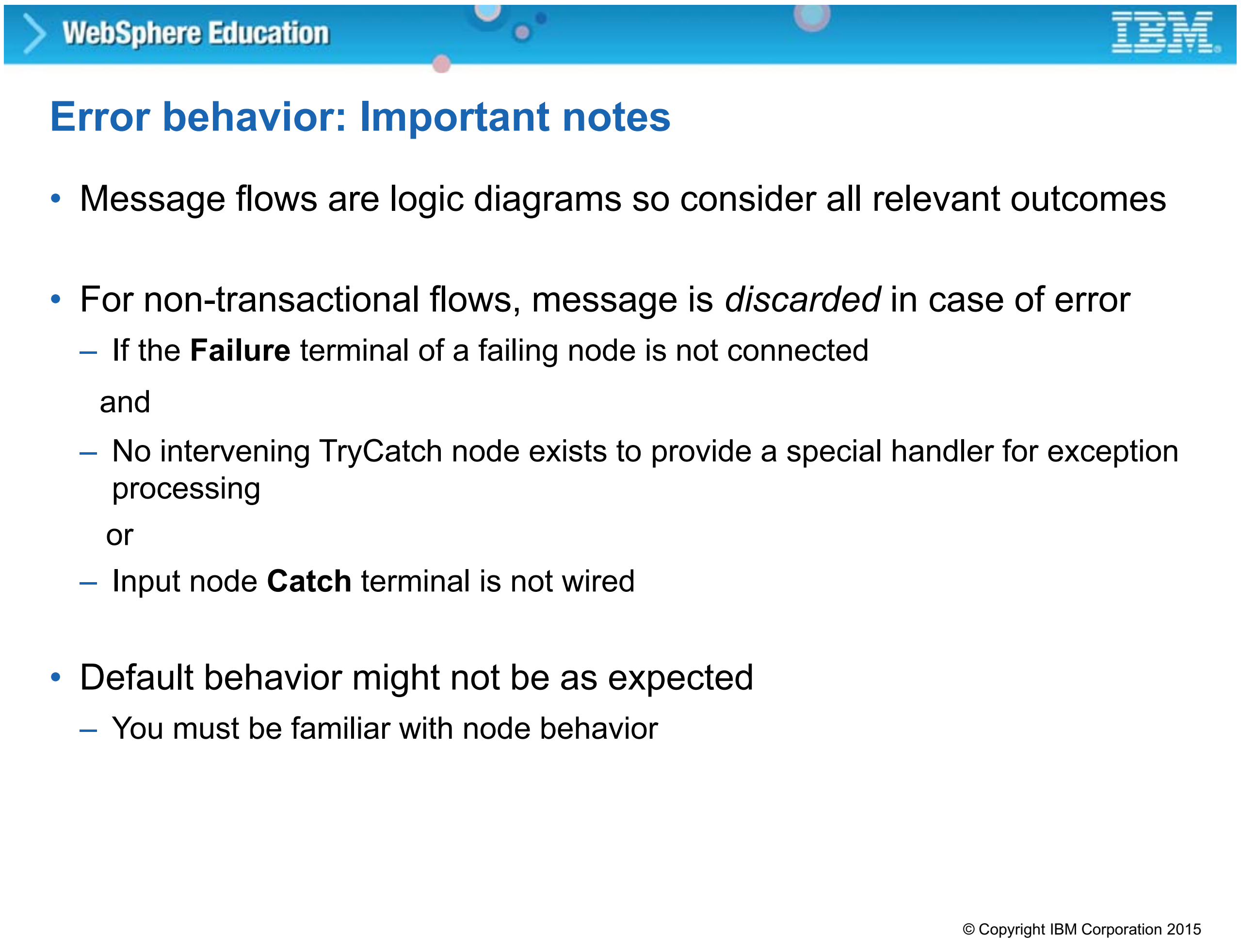
  Example nodes: Database, Compute, Filter, MQOutput, and Mapping

Figure 9-6. Transaction support WM6461.0

Notes:

The default integration node behavior is to coordinate transactions. It explicitly and separately commits (or backs out) database resources at *flow completion*, and does the same for IBM MQ resources. The default integration node behavior might work in most cases, but there is the chance of a problem if the first commit (database) succeeds, but the second (IBM MQ) does not.

Using a two-phase commit with a single transaction coordinator prevents this problem. IBM MQ can act as the transaction coordinator for its own queues and for database resources, provided the queue manager is suitably configured. The Integration Bus uses this IBM MQ facility when supervising a globally coordinated transaction (XA-coordination with a two-phase commit).



WebSphere Education IBM

Error behavior: Important notes

- Message flows are logic diagrams so consider all relevant outcomes
- For non-transactional flows, message is *discarded* in case of error
  - If the **Failure** terminal of a failing node is not connected

    and
  - No intervening TryCatch node exists to provide a special handler for exception processing

    or
  - Input node **Catch** terminal is not wired
- Default behavior might not be as expected
  - You must be familiar with node behavior

© Copyright IBM Corporation 2015

Figure 9-7. Error behavior: Important notes WM6461.0

Notes:

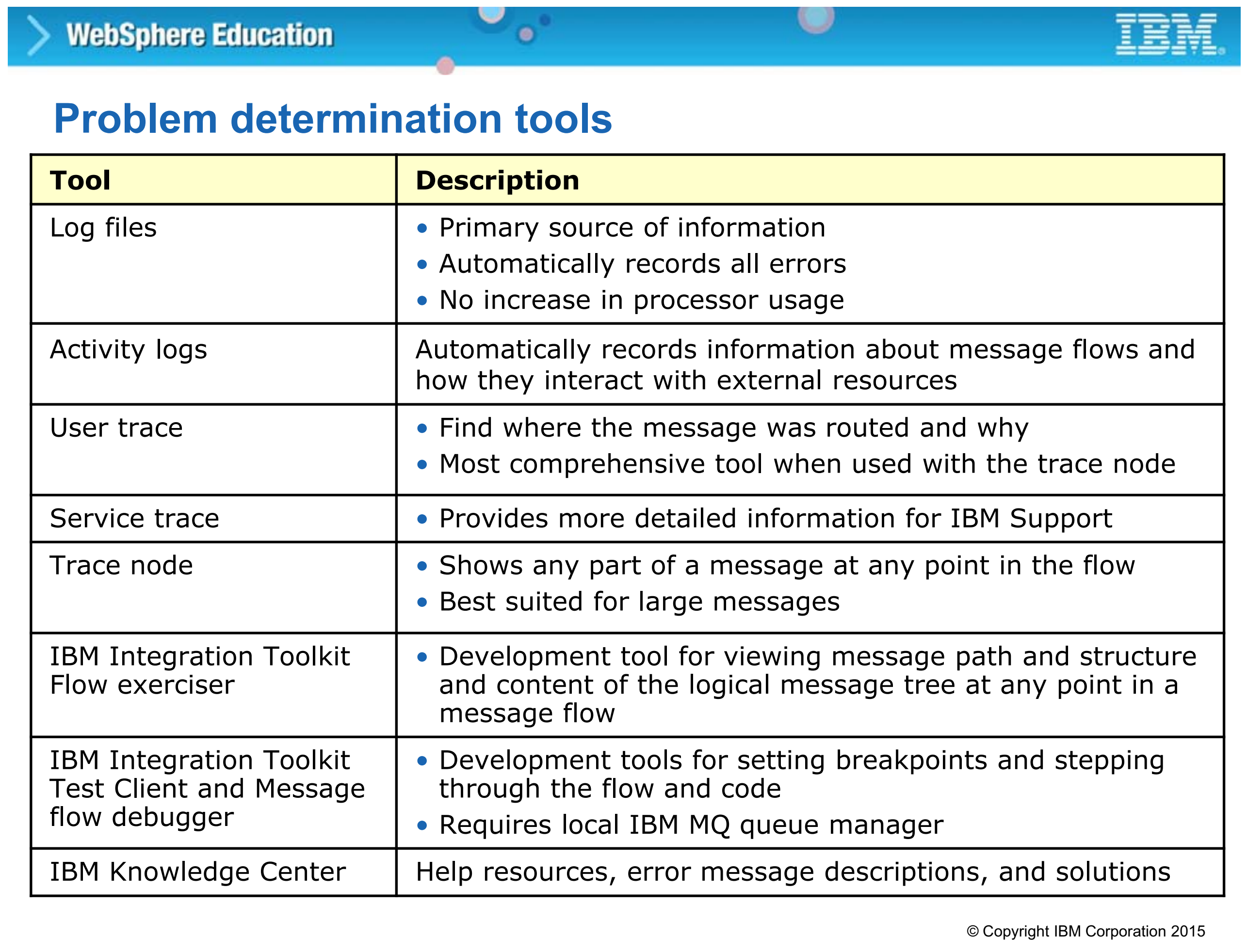
Message flows that rely on default behavior, that is, with node output terminals unwired, put the administrator in an awkward position. It is the administrator who must be aware of possible default behaviors. The default behavior includes a message flow loop as a final action. A symptom of such a loop is a message count (current depth) on an input queue not decrementing (or incrementing), implying a backlog of unprocessed messages.

Developers must anticipate and handle all potential errors within the flow. Otherwise, messages can be delivered to unexpected queues, or discarded; or looping can occur.

More complex message flows, such as those that use Filter and Compute nodes, can introduce even more possible destinations, depending on how a particular message flow is wired to handle or ignore errors.

The developer can add a TryCatch node to provide a special handler for exception processing. If a downstream node generates an exception, the TryCatch node catches it and routes the original message to its **Catch** terminal.

9.2. Testing and problem determination tools



WebSphere Education

Problem determination tools

Tool	Description
Log files	• Primary source of information • Automatically records all errors • No increase in processor usage
Activity logs	Automatically records information about message flows and how they interact with external resources
User trace	• Find where the message was routed and why • Most comprehensive tool when used with the trace node
Service trace	• Provides more detailed information for IBM Support
Trace node	• Shows any part of a message at any point in the flow • Best suited for large messages
IBM Integration Toolkit Flow exerciser	• Development tool for viewing message path and structure and content of the logical message tree at any point in a message flow
IBM Integration Toolkit Test Client and Message flow debugger	• Development tools for setting breakpoints and stepping through the flow and code • Requires local IBM MQ queue manager
IBM Knowledge Center	Help resources, error message descriptions, and solutions

Figure 9-8. Problem determination tools WM6461.0

Notes:

The table summarizes the testing and problem determination tools available in Integration Bus.

In many cases, a good practice is to use a combination of tools and methods to determine the reason for an integration node, integration server, or message flow failure.

Each of these tools is described in detail in this unit.

IBM MQ SupportPacs and IBM DeveloperWorks have more test utilities. For more information, see the following websites:

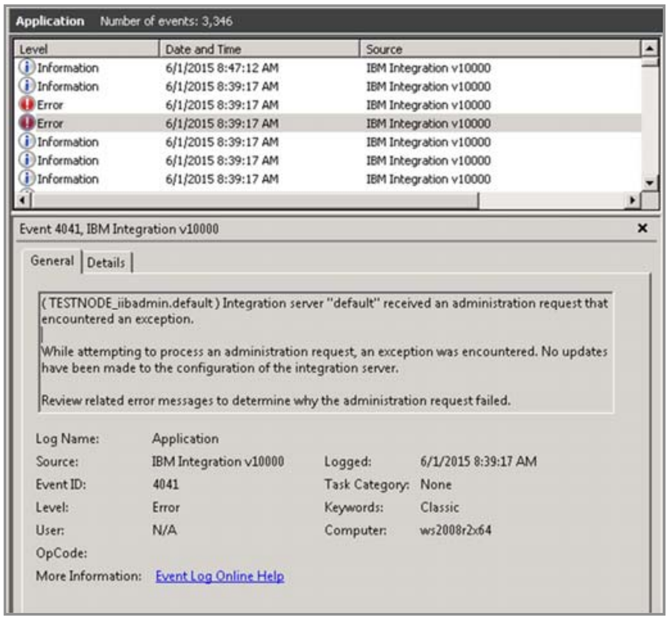
`http://www.ibm.com/software/integration/support/supportpacs`

`http://www.ibm.com/developerworks/`

WebSphere Education

Local error log (syslog)

- Multiple messages per runtime error (review them all)
  - Status of integration node
  - Run time errors
  - Some database errors
- Always check after a system start, after deployment, and during testing of message flows
- Location differs based on the operating system:
  - On Windows, the local error log is the Windows Event log (**Application** view)
  - On Linux and UNIX, syslog location is based on entry in `/etc/syslog.conf`
  - On z/OS, the local error log is the operator console



Application Number of events: 3,346

Level	Date and Time	Source
Information	6/1/2015 8:47:12 AM	IBM Integration v10000
Information	6/1/2015 8:39:17 AM	IBM Integration v10000
Error	6/1/2015 8:39:17 AM	IBM Integration v10000
Error	6/1/2015 8:39:17 AM	IBM Integration v10000
Information	6/1/2015 8:39:17 AM	IBM Integration v10000
Information	6/1/2015 8:39:17 AM	IBM Integration v10000
Information	6/1/2015 8:39:17 AM	IBM Integration v10000

Event 4041, IBM Integration v10000

General | Details

```
( TESTNODE_iibadmin.default ) Integration server "default" received an administration request that
encountered an exception.

While attempting to process an administration request, an exception was encountered. No updates
have been made to the configuration of the integration server.

Review related error messages to determine why the administration request failed.
```

Log Name:	Application		
Source:	IBM Integration v10000	Logged:	6/1/2015 8:39:17 AM
Event ID:	4041	Task Category:	None
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	ws2008r2x64
OpCode:			
More Information:	Event Log Online Help		

© Copyright IBM Corporation 2015

Figure 9-9. Local error log (syslog) WM6461.0

Notes:

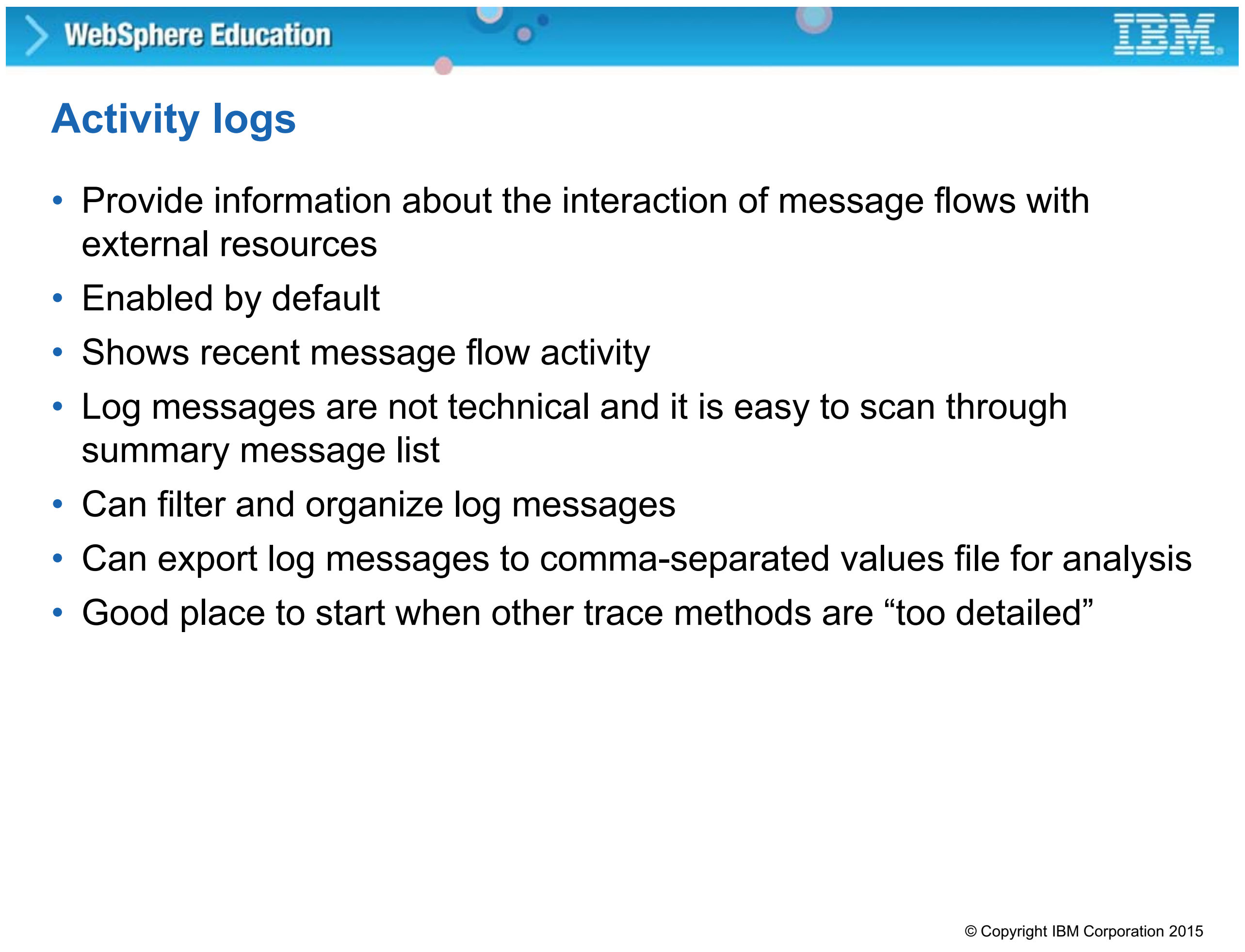
The local error log is the primary source of runtime problem determination because all errors and warnings are recorded there. It is possible that errors beyond the integration node can occur; for instance, with a database that is referenced in a message flow.

It is important to check the local error log regularly. Errors can accumulate in the local error log and fill it if it is not acted upon.

In Windows, Integration Bus automatically writes to the Event Viewer **Application** log.

On operating systems other than Windows, the system log (syslog) contains only the first line of the message, and the user is expected to look up the remainder of the message.

In UNIX and Linux, you must explicitly configure the location of the syslog.



WebSphere Education

Activity logs

- Provide information about the interaction of message flows with external resources
- Enabled by default
- Shows recent message flow activity
- Log messages are not technical and it is easy to scan through summary message list
- Can filter and organize log messages
- Can export log messages to comma-separated values file for analysis
- Good place to start when other trace methods are “too detailed”

© Copyright IBM Corporation 2015

Figure 9-10. Activity logs WM6461.0

Notes:

The Integration Bus *activity log* reports interactions between message flows and external resources, such as queues, databases, files, and JMS.

The activity log is enabled by default. It reports on recent message flow activity, so if you discover a problem with a message flow, you do not need to enable any debugging or tracing facilities; you need only to review the activity log. The summary log messages are not detailed, so you can quickly spot a problem in the log. When you select the log message, you see the complete details.

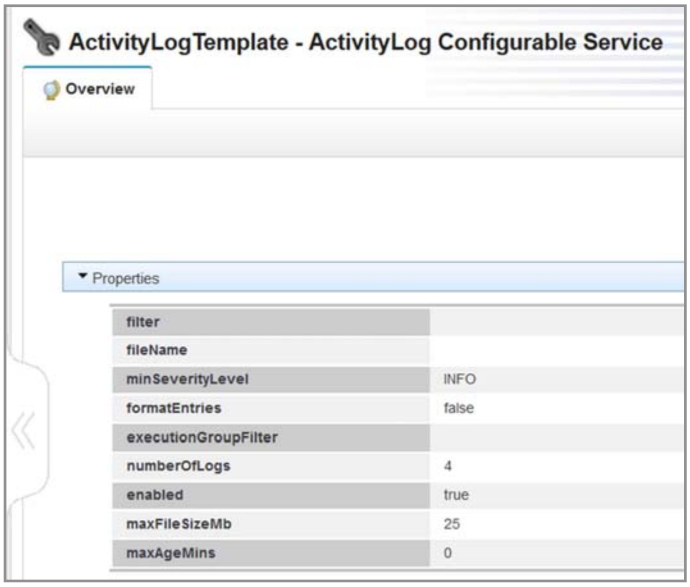
When you discover a message flow problem, you might save time and effort by checking the activity log before enabling other tracing and debugging tools.

You can export the log information to a comma-separated values or a flat file.

WebSphere Education IBM

Writing Activity logs to files

- Define an ActivityLog configurable service to write Activity logs to a file
- Option1: Use the IBM Integration web user interface
  1. Right-click **Operational Policy > Configurable Services** and then click **Create**
  2. Enter a configurable service name and then select **Activity Log Template**
  3. Populate template and then click **Save**
- Option 2: Use the **`mqsicreateconfigurableservice`** command


ActivityLogTemplate - ActivityLog Configurable Service

Overview

Properties

filter	
fileName	
minSeverityLevel	INFO
formatEntries	false
executionGroupFilter	
numberOfLogs	4
enabled	true
maxFileSizeMb	25
maxAgeMins	0

© Copyright IBM Corporation 2015

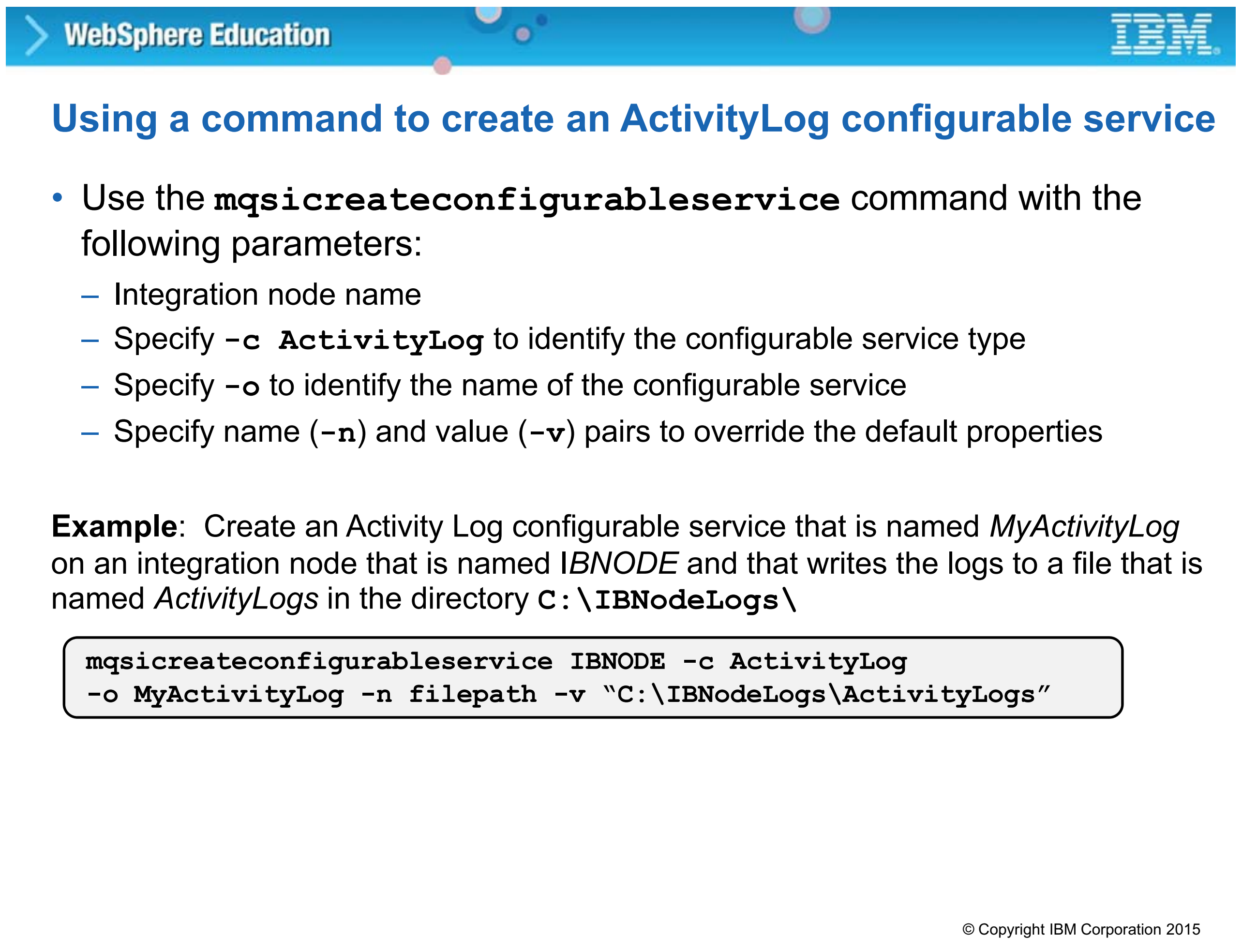
Figure 9-11. Writing Activity logs to files WM6461.0

Notes:

You can use the ActivityLog configurable service to configure the following Activity log file characteristics:

- Whether file logging is enabled
- The maximum number of files that are used for each log
- The maximum size of the log files (before the log rotates to the next file)
- The maximum age of the log files (before the log rotates to the next file)
- Filters that determine the content of the logs
- The severity levels of messages logged
- The log path and file name
- Whether messages are formatted with inserts included in the message text
- The scope of the configurable service (which integration server it applies to)

The configurable service name must not duplicate the name of an existing configurable service or IBM predefined configurable service template.



WebSphere Education IBM

Using a command to create an ActivityLog configurable service

- Use the **`mqsicreateconfigurableservice`** command with the following parameters:
  - Integration node name
  - Specify **`-c ActivityLog`** to identify the configurable service type
  - Specify **`-o`** to identify the name of the configurable service
  - Specify name (**`-n`**) and value (**`-v`**) pairs to override the default properties

Example: Create an Activity Log configurable service that is named *MyActivityLog* on an integration node that is named I*BNODE* and that writes the logs to a file that is named *ActivityLogs* in the directory **`C:\IBNodeLogs\`**

```
mqsicreateconfigurableservice IBNODE -c ActivityLog
-o MyActivityLog -n filepath -v “C:\IBNodeLogs\ActivityLogs”
```

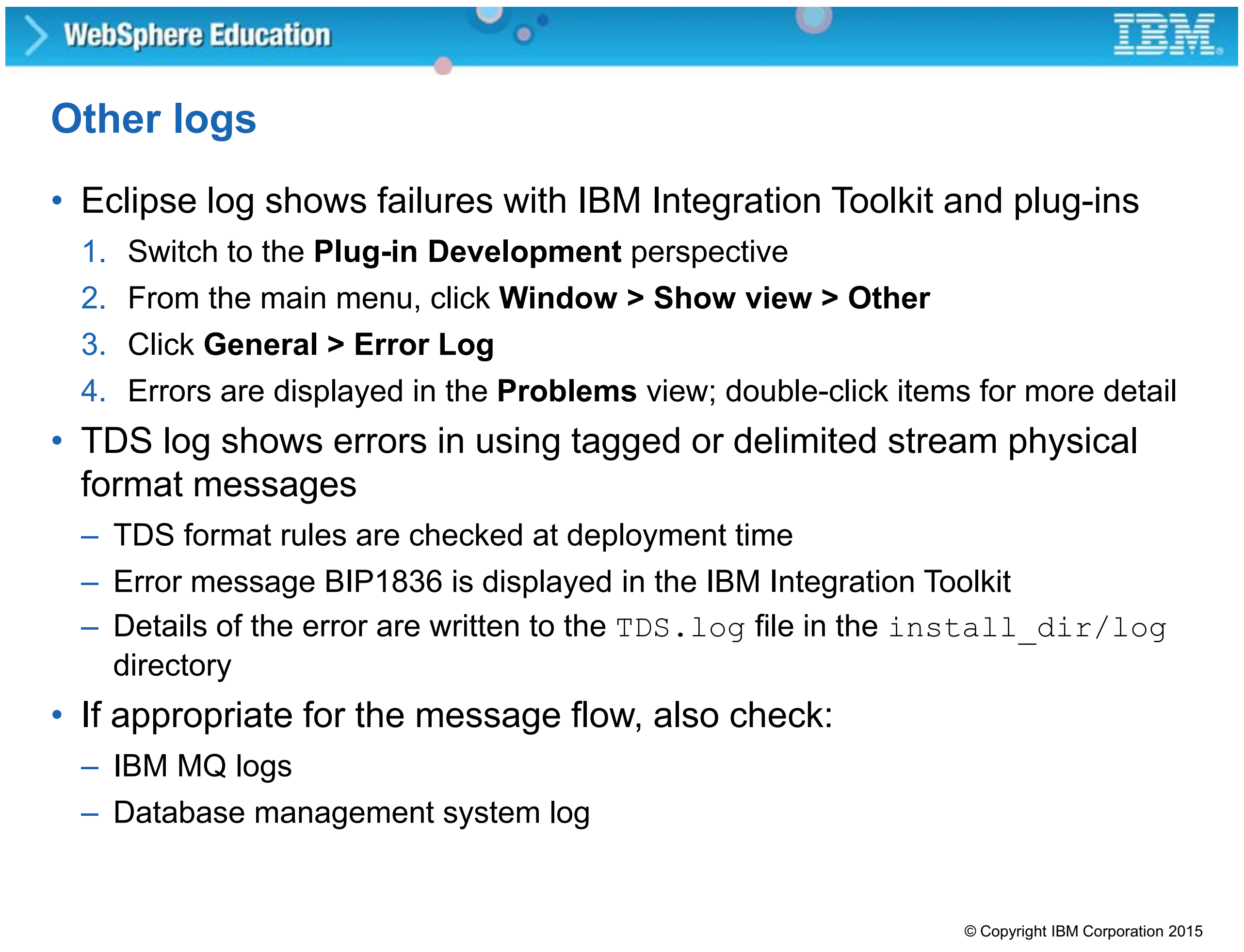
© Copyright IBM Corporation 2015

Figure 9-12. Using a command to create an ActivityLog configurable service WM6461.0

Notes:

A string value to specify the fully qualified path and the file name for Activity log files. This value is not set by default. If it is not set, file logging is not enabled. To enable file logging, set the value, which must be distinct for each Activity Log configurable service instance.

The most recent activities are logged to a file named *fileName*. The names of the other files in the circular log are constructed from *fileName* and standard suffixes. For example, a value of `filePath\myLog` results in log files named `myLog` (the current log file), `myLog.1`, `myLog.2`, and so on, where files with higher prefixes contain older log entries. These files are written to the directory specified by the **filePath** property.



WebSphere Education IBM

Other logs

- Eclipse log shows failures with IBM Integration Toolkit and plug-ins
  1. Switch to the **Plug-in Development** perspective
  2. From the main menu, click **Window > Show view > Other**
  3. Click **General > Error Log**
  4. Errors are displayed in the **Problems** view; double-click items for more detail
- TDS log shows errors in using tagged or delimited stream physical format messages
  - TDS format rules are checked at deployment time
  - Error message BIP1836 is displayed in the IBM Integration Toolkit
  - Details of the error are written to the `TDS.log` file in the `install_dir/log` directory
- If appropriate for the message flow, also check:
  - IBM MQ logs
  - Database management system log

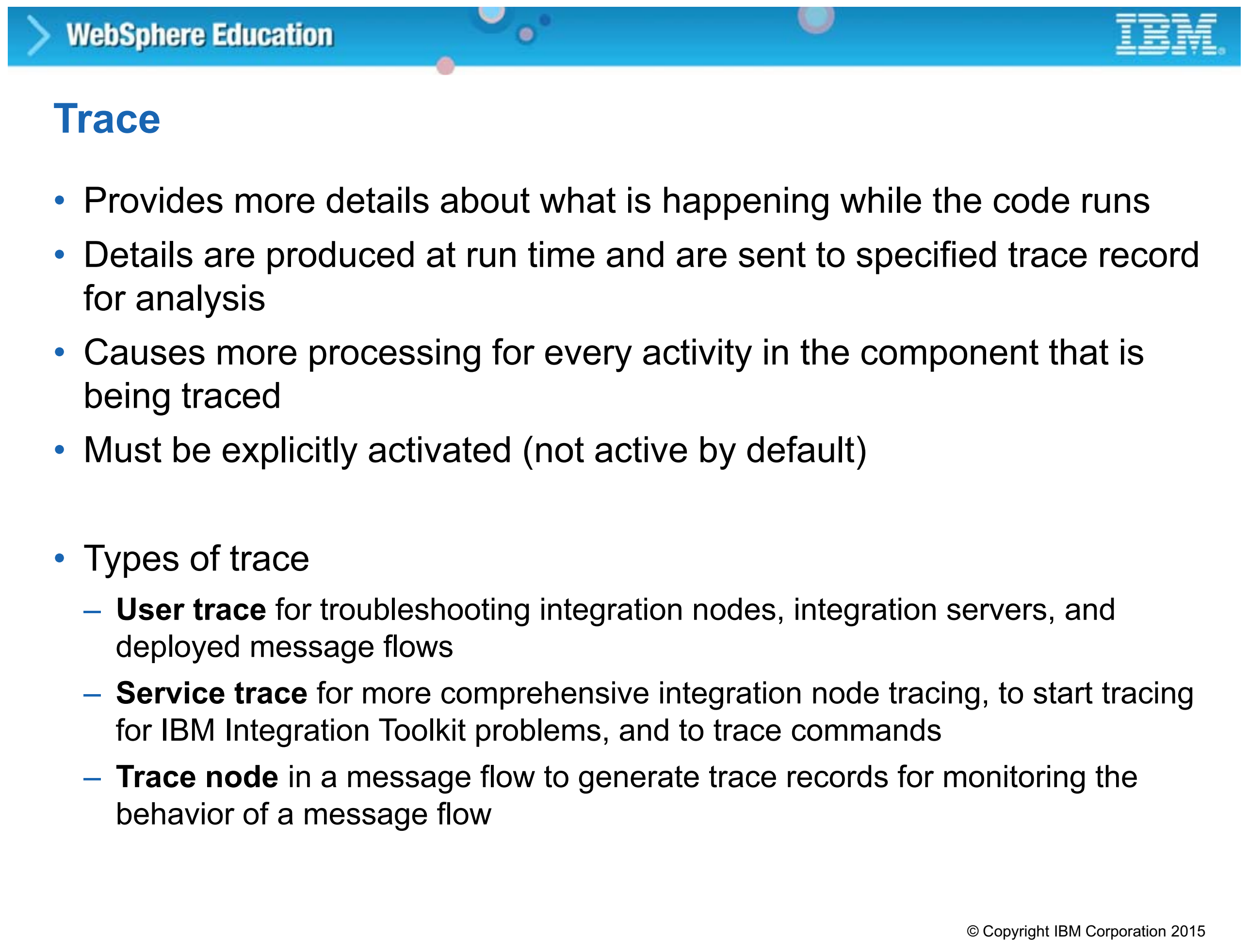
© Copyright IBM Corporation 2015

Figure 9-13. Other logs WM6461.0

Notes:

In addition to the system log, activity log, and the other message repositories that you already are familiar, other Integration Bus logs exist. While these logs might not be the first places that you check when errors occur, they can be useful in particular instances. These logs include the Eclipse log, TDS log, and external application logs.

If you engage IBM support, you might be asked to review these logs, or provide them to the support team.



WebSphere Education

Trace

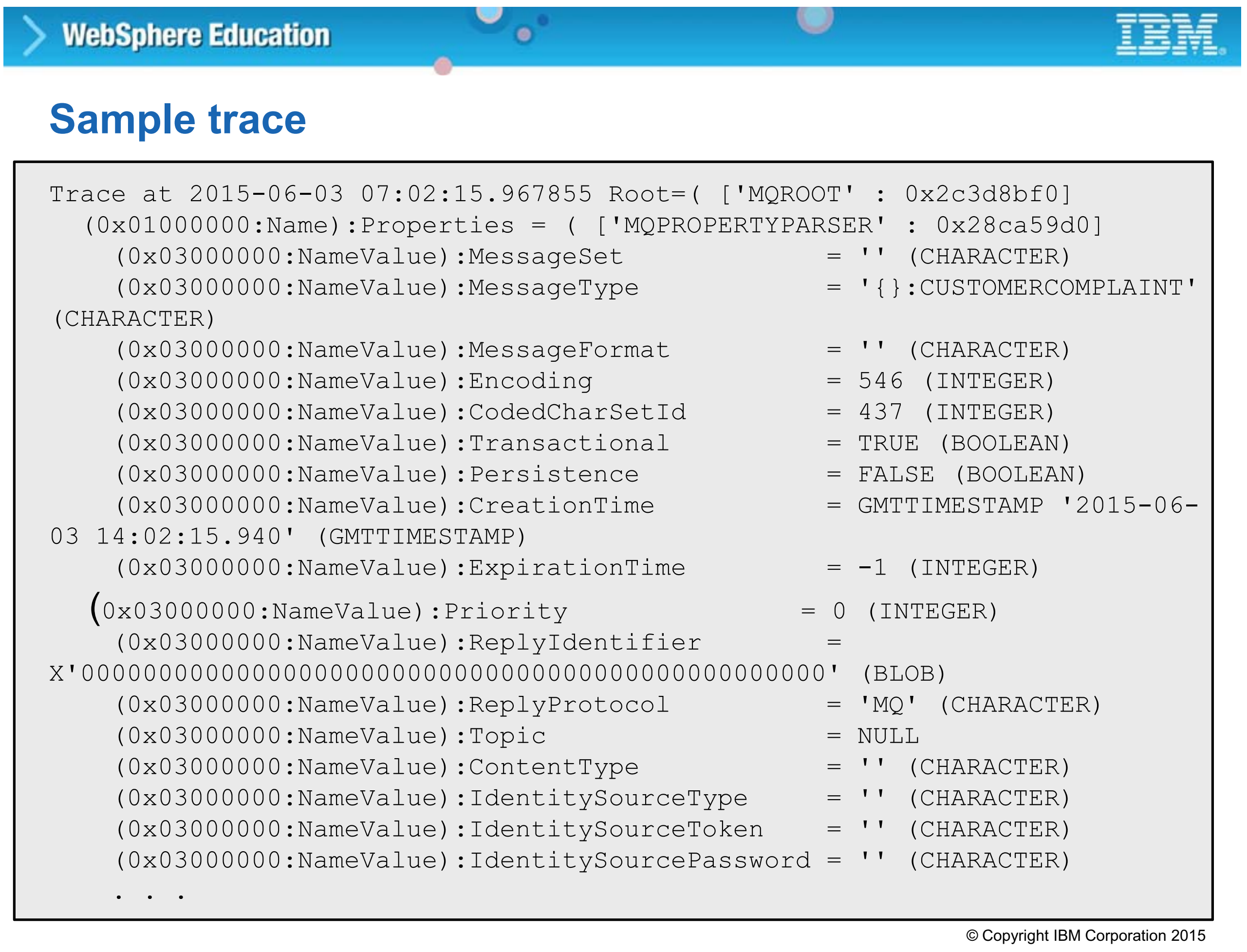
- Provides more details about what is happening while the code runs
- Details are produced at run time and are sent to specified trace record for analysis
- Causes more processing for every activity in the component that is being traced
- Must be explicitly activated (not active by default)

- Types of trace
  - **User trace** for troubleshooting integration nodes, integration servers, and deployed message flows
  - **Service trace** for more comprehensive integration node tracing, to start tracing for IBM Integration Toolkit problems, and to trace commands
  - **Trace node** in a message flow to generate trace records for monitoring the behavior of a message flow

Figure 9-14. Trace WM6461.0

Notes:

If you cannot get enough information about a particular problem from the entries that are available in the various logs, the next troubleshooting method to consider is using trace. Trace provides more details about what is happening while code runs. The information that is produced from trace is sent to a specified trace record so that you or IBM support personnel can analyze it to discover the cause of your problem.



WebSphere Education IBM

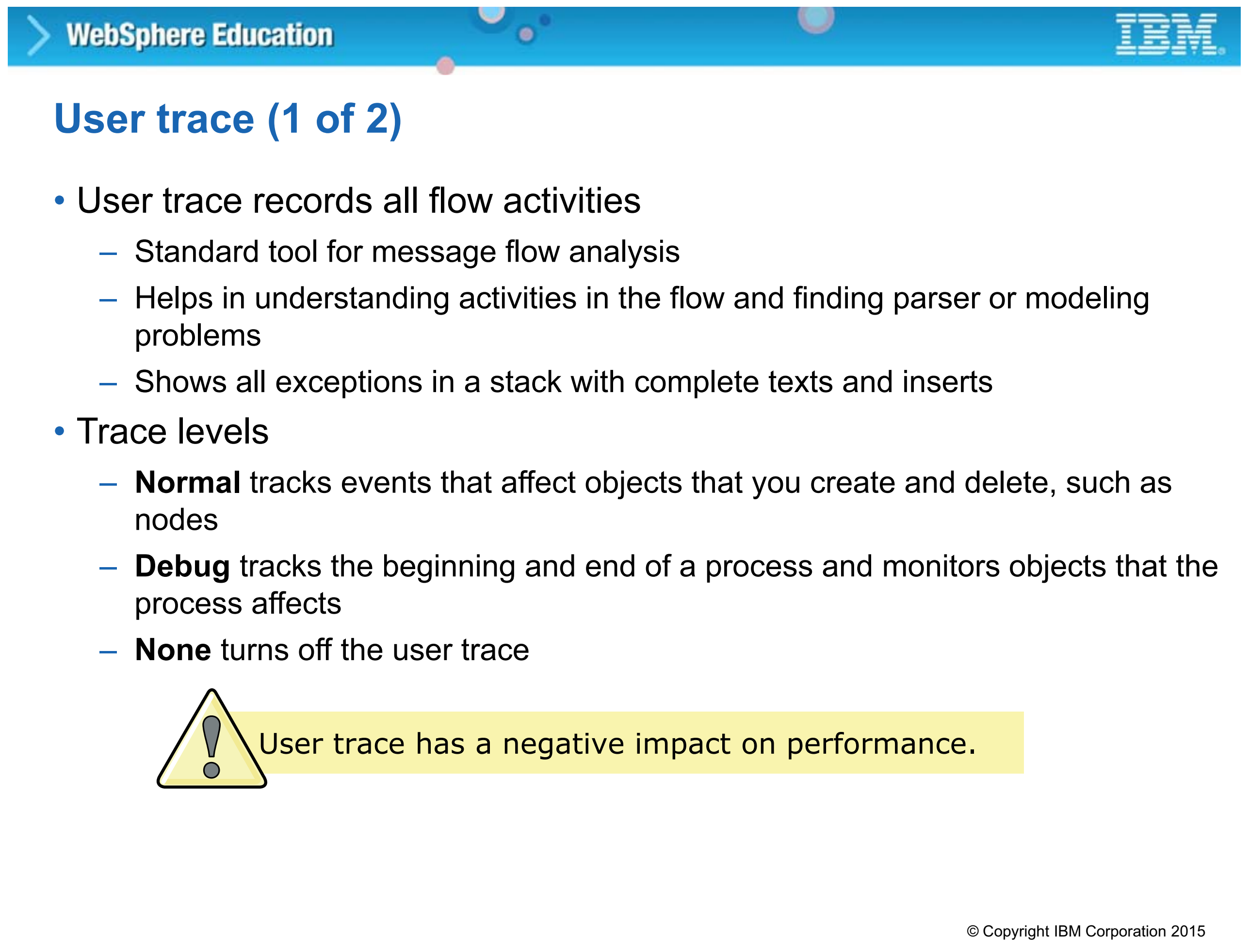
Sample trace

```
Trace at 2015-06-03 07:02:15.967855 Root=( ['MQROOT' : 0x2c3d8bf0]
  (0x01000000:Name):Properties = ( ['MQPROPERTYPARSER' : 0x28ca59d0]
    (0x03000000:NameValue):MessageSet             = '' (CHARACTER)
    (0x03000000:NameValue):MessageType            = '{}:CUSTOMERCOMPLAINT'
(CHARACTER)
    (0x03000000:NameValue):MessageFormat          = '' (CHARACTER)
    (0x03000000:NameValue):Encoding               = 546 (INTEGER)
    (0x03000000:NameValue):CodedCharSetId         = 437 (INTEGER)
    (0x03000000:NameValue):Transactional          = TRUE (BOOLEAN)
    (0x03000000:NameValue):Persistence            = FALSE (BOOLEAN)
    (0x03000000:NameValue):CreationTime           = GMTTIMESTAMP '2015-06-
03 14:02:15.940' (GMTTIMESTAMP)
    (0x03000000:NameValue):ExpirationTime         = -1 (INTEGER)
    (0x03000000:NameValue):Priority               = 0 (INTEGER)
    (0x03000000:NameValue):ReplyIdentifier        =
X'000000000000000000000000000000000000000000000000' (BLOB)
    (0x03000000:NameValue):ReplyProtocol          = 'MQ' (CHARACTER)
    (0x03000000:NameValue):Topic                  = NULL
    (0x03000000:NameValue):ContentType            = '' (CHARACTER)
    (0x03000000:NameValue):IdentitySourceType     = '' (CHARACTER)
    (0x03000000:NameValue):IdentitySourceToken    = '' (CHARACTER)
    (0x03000000:NameValue):IdentitySourcePassword = '' (CHARACTER)
    . . .
```

Figure 9-15. Sample trace WM6461.0

Notes:

This figure shows an example of a trace.



WebSphere Education

User trace (1 of 2)

- User trace records all flow activities
  - Standard tool for message flow analysis
  - Helps in understanding activities in the flow and finding parser or modeling problems
  - Shows all exceptions in a stack with complete texts and inserts
- Trace levels
  - **Normal** tracks events that affect objects that you create and delete, such as nodes
  - **Debug** tracks the beginning and end of a process and monitors objects that the process affects
  - **None** turns off the user trace

User trace has a negative impact on performance.

Figure 9-16. User trace (1 of 2) WM6461.0

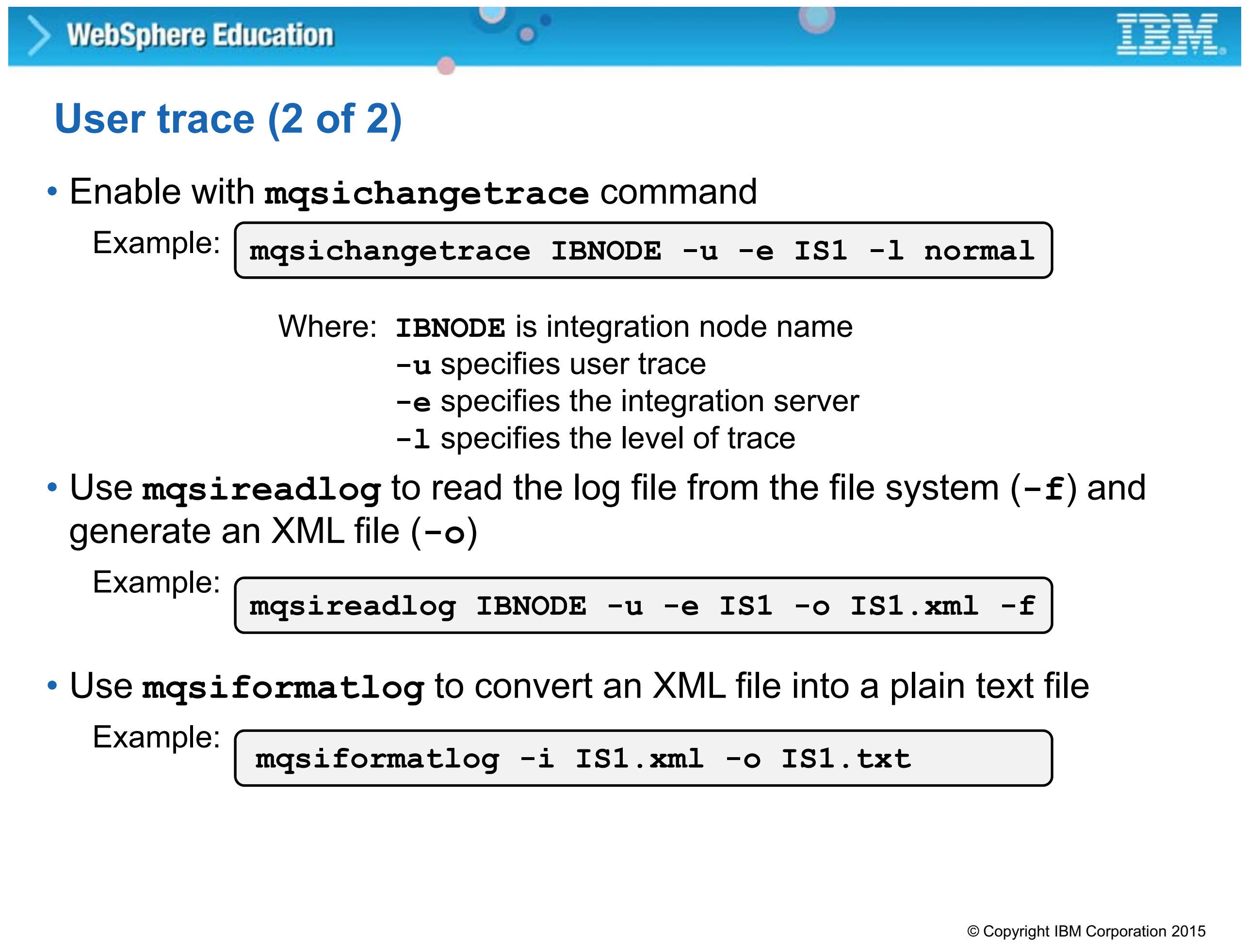
Notes:

Exceptions are often longer than a single line, and they normally come in groups of two or three. User trace shows the full text, with inserts, of all the exceptions in the stack. Some parsers emit user trace messages while parsing. Some parsing or modeling problems are hard to diagnose without this information.

You can usually use user trace for debugging your applications. You can trace integration nodes, integration servers, and deployed message flows.

Warning

When you turn on user trace, you are causing extra processing for every activity in the component you are tracing. The components generate large quantities of data. Therefore, you must expect to see some impact in performance while the trace is active. However, you can limit this additional processing by being selective about what you trace, and by restricting the time during which trace is active.


WebSphere Education IBM

User trace (2 of 2)

- Enable with **`mqsichangetrace`** command

  Example:
  ```
  mqsichangetrace IBNODE -u -e IS1 -l normal
  ```
  Where: **`IBNODE`** is integration node name
  `-u` specifies user trace
  `-e` specifies the integration server
  `-l` specifies the level of trace

- Use **`mqsireadlog`** to read the log file from the file system (**`-f`**) and generate an XML file (**`-o`**)

  Example:
  ```
  mqsireadlog IBNODE -u -e IS1 -o IS1.xml -f
  ```

- Use **`mqsiformatlog`** to convert an XML file into a plain text file

  Example:
  ```
  mqsiformatlog -i IS1.xml -o IS1.txt
  ```

© Copyright IBM Corporation 2015

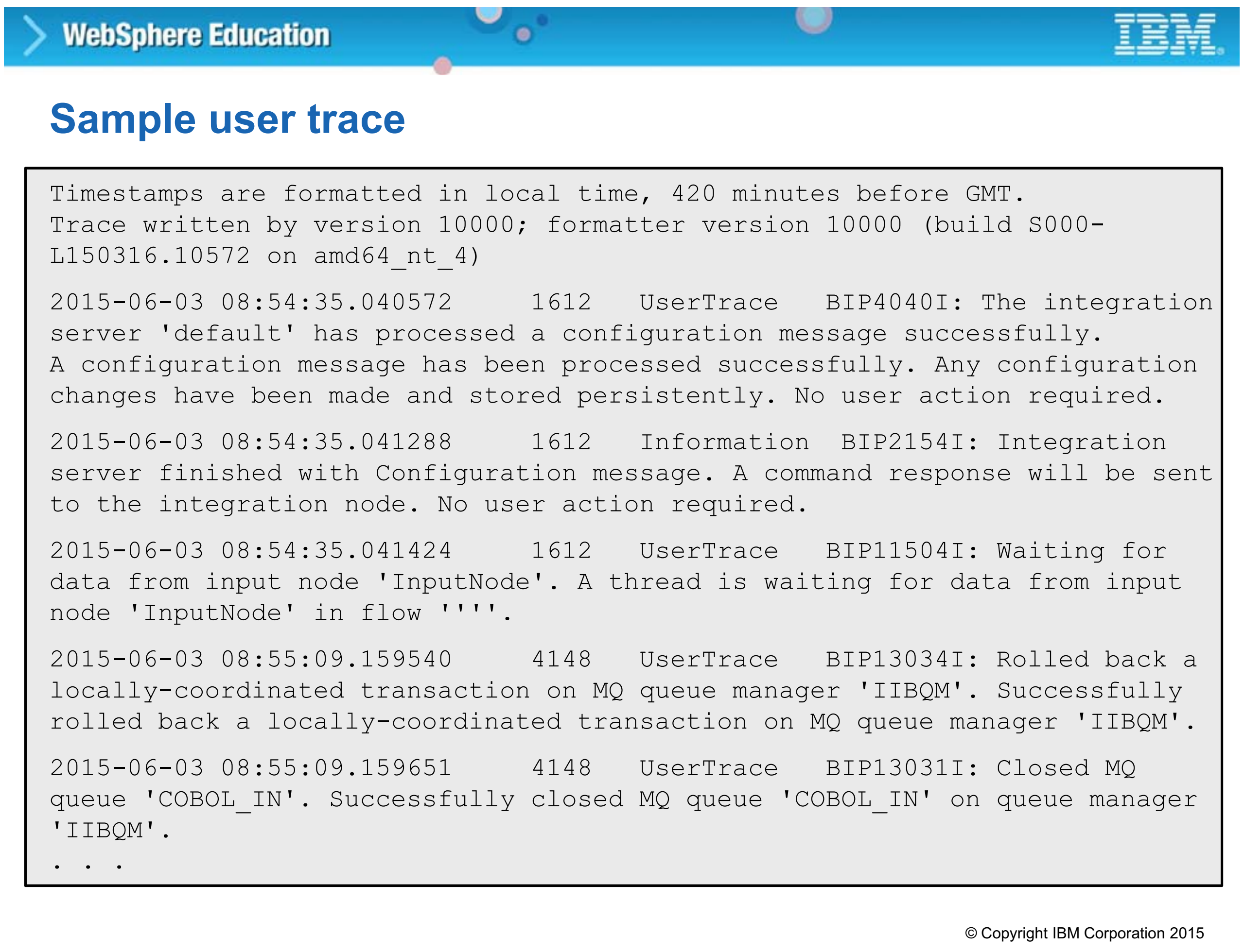
Figure 9-17. User trace (2 of 2) WM6461.0

Notes:

You can activate user trace by using the `mqsichangetrace` command.

Trace data is collected in binary form. After the trace data is collected, you must use the `mqsireadlog` command to process the data and generates an XML file. To get the XML data into plain text, you must run another command, `mqsiformatlog`.

To determine the current trace status, use the `mqsireporttrace` command or check the status in the IBM Integration web user interface.



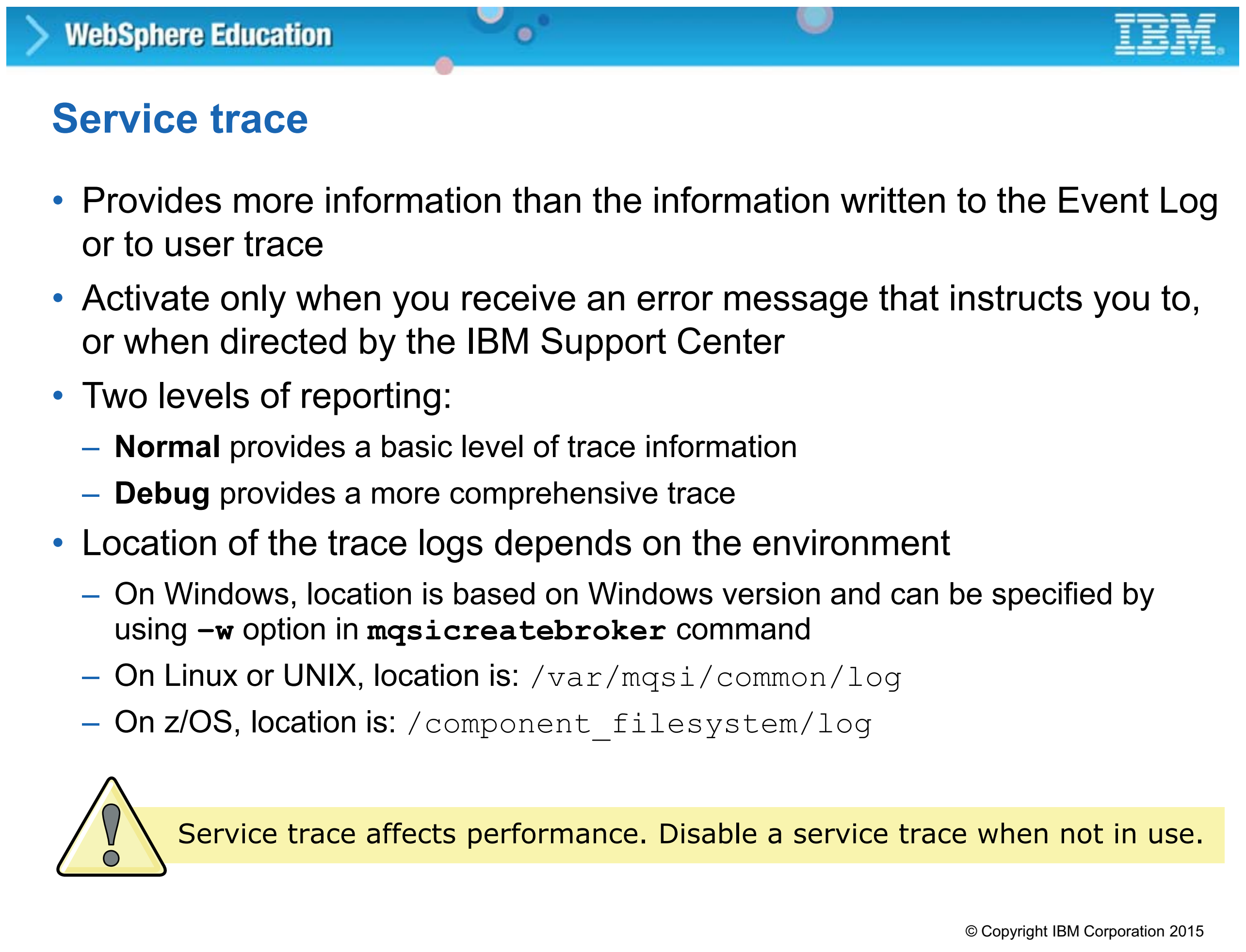
Sample user trace

```
Timestamps are formatted in local time, 420 minutes before GMT.
Trace written by version 10000; formatter version 10000 (build S000-
L150316.10572 on amd64_nt_4)

2015-06-03 08:54:35.040572     1612   UserTrace   BIP4040I: The integration
server 'default' has processed a configuration message successfully.
A configuration message has been processed successfully. Any configuration
changes have been made and stored persistently. No user action required.

2015-06-03 08:54:35.041288     1612   Information  BIP2154I: Integration
server finished with Configuration message. A command response will be sent
to the integration node. No user action required.

2015-06-03 08:54:35.041424     1612   UserTrace   BIP11504I: Waiting for
data from input node 'InputNode'. A thread is waiting for data from input
node 'InputNode' in flow ''''.

2015-06-03 08:55:09.159540     4148   UserTrace   BIP13034I: Rolled back a
locally-coordinated transaction on MQ queue manager 'IIBQM'. Successfully
rolled back a locally-coordinated transaction on MQ queue manager 'IIBQM'.

2015-06-03 08:55:09.159651     4148   UserTrace   BIP13031I: Closed MQ
queue 'COBOL_IN'. Successfully closed MQ queue 'COBOL_IN' on queue manager
'IIBQM'.
. . .
```

Figure 9-18. Sample user trace WM6461.0

Notes:

This figure shows an example of user trace.



Service trace

- Provides more information than the information written to the Event Log or to user trace
- Activate only when you receive an error message that instructs you to, or when directed by the IBM Support Center
- Two levels of reporting:
  - **Normal** provides a basic level of trace information
  - **Debug** provides a more comprehensive trace
- Location of the trace logs depends on the environment
  - On Windows, location is based on Windows version and can be specified by using **`-w`** option in **`mqsicreatebroker`** command
  - On Linux or UNIX, location is: `/var/mqsi/common/log`
  - On z/OS, location is: `/component_filesystem/log`

Service trace affects performance. Disable a service trace when not in use.

Figure 9-19. Service trace WM6461.0

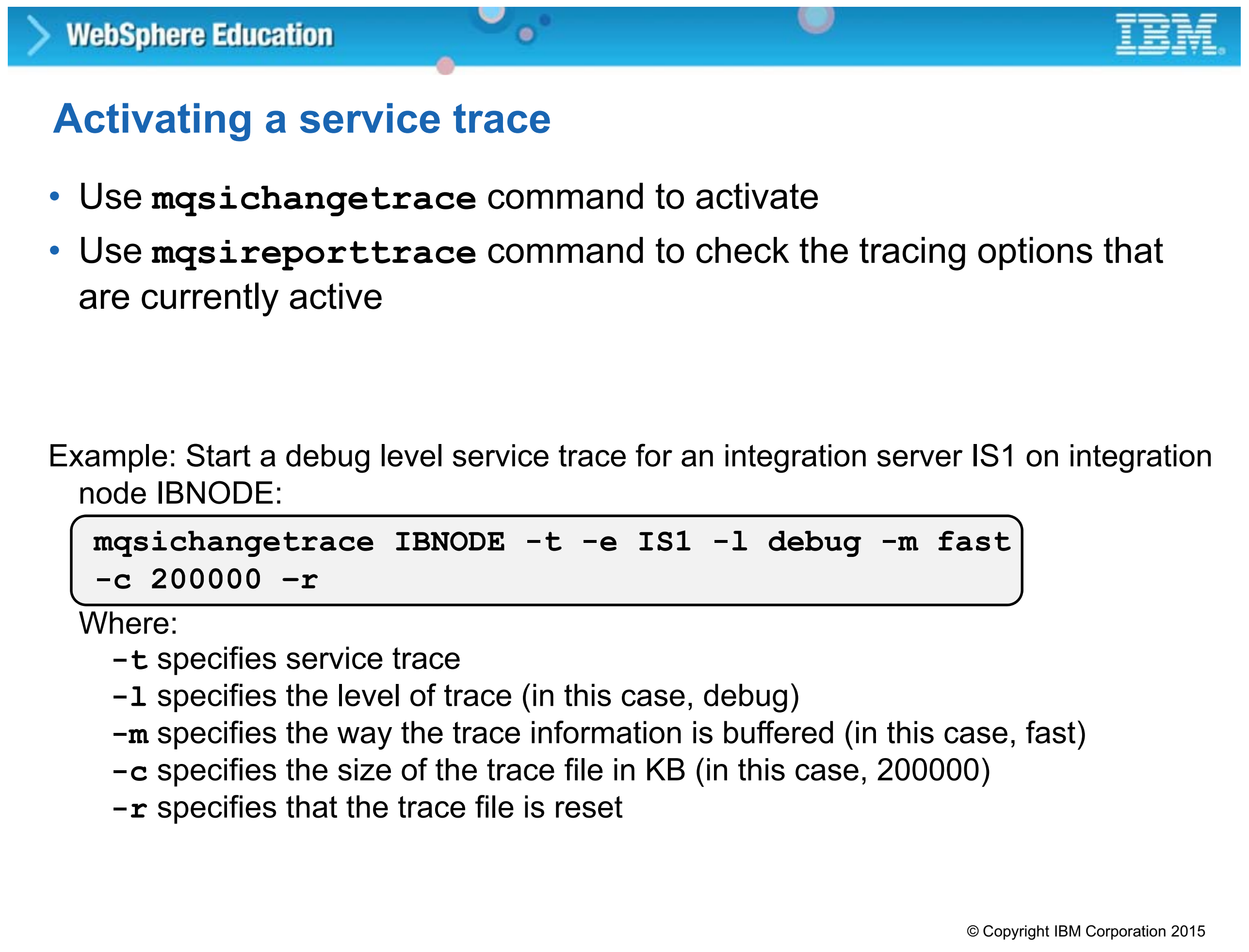
Notes:

With service trace, you can activate more comprehensive integration node tracing, and start tracing for the IBM Integration Toolkit. You can also trace the execution of all the commands, including the trace commands themselves.

When you activate service trace, you cause more processing for every activity in the component that you are tracing. The components generate large quantities of data. Performance can be affected while service trace is active. You can limit this additional processing by being selective about what you trace, and by restricting the time during which trace is active.

The location of the trace log depends on the environment. If you set the work path by using the `-w` parameter of the `mqsicreatebroker` command, the location is `workpath\log`.

If you did not specify the integration node work path, the default location is `%ALLUSERSPROFILE%\Application Data\IBM\MQSI\common\log` where `%ALLUSERSPROFILE%` is the environment variable that defines the system work directory.



WebSphere Education

Activating a service trace

- Use **`mqsichangetrace`** command to activate
- Use **`mqsireporttrace`** command to check the tracing options that are currently active

Example: Start a debug level service trace for an integration server IS1 on integration node IBNODE:

```
mqsichangetrace IBNODE -t -e IS1 -l debug -m fast
-c 200000 -r
```

Where:

`-t` specifies service trace
`-l` specifies the level of trace (in this case, debug)
`-m` specifies the way the trace information is buffered (in this case, fast)
`-c` specifies the size of the trace file in KB (in this case, 200000)
`-r` specifies that the trace file is reset

Figure 9-20. Activating a service trace WM6461.0

Notes:

The `mqsichangetrace` command can be used to activate a service trace.

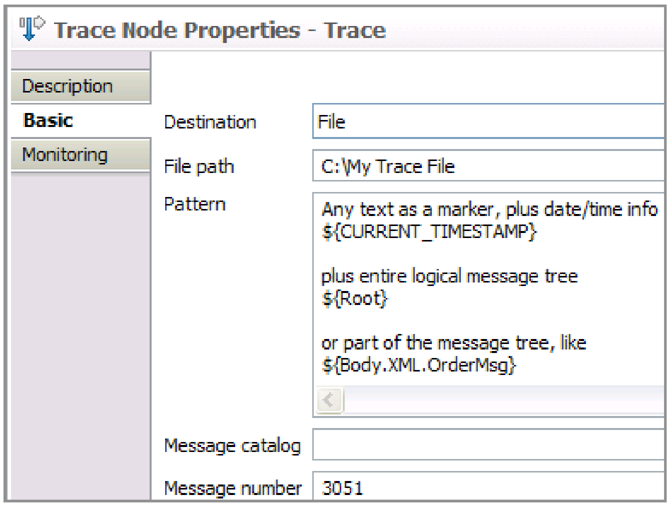
The `mqsireporttrace` command can be used to check the currently active tracing options.

The figure shows an example of the command to activate a service trace and a summary of the command options.

WebSphere Education IBM

Trace node

- Provides snapshot of (part of) a logical message
- Destination options:
  - File
  - User Trace log
  - Local system error log
- Content:
  - User-defined text
  - Message content
  - Date/time information
- Can be switched "on" or "off"



Trace Node Properties - Trace

Destination	File
File path	C:\My Trace File
Pattern	Any text as a marker, plus date/time info ${CURRENT_TIMESTAMP} plus entire logical message tree ${Root} or part of the message tree, like ${Body.XML.OrderMsg}
Message catalog	
Message number	3051

You can significantly improve the performance of a flow that includes Trace nodes by switching Trace nodes "off"

© Copyright IBM Corporation 2015

Figure 9-21. Trace node WM6461.0

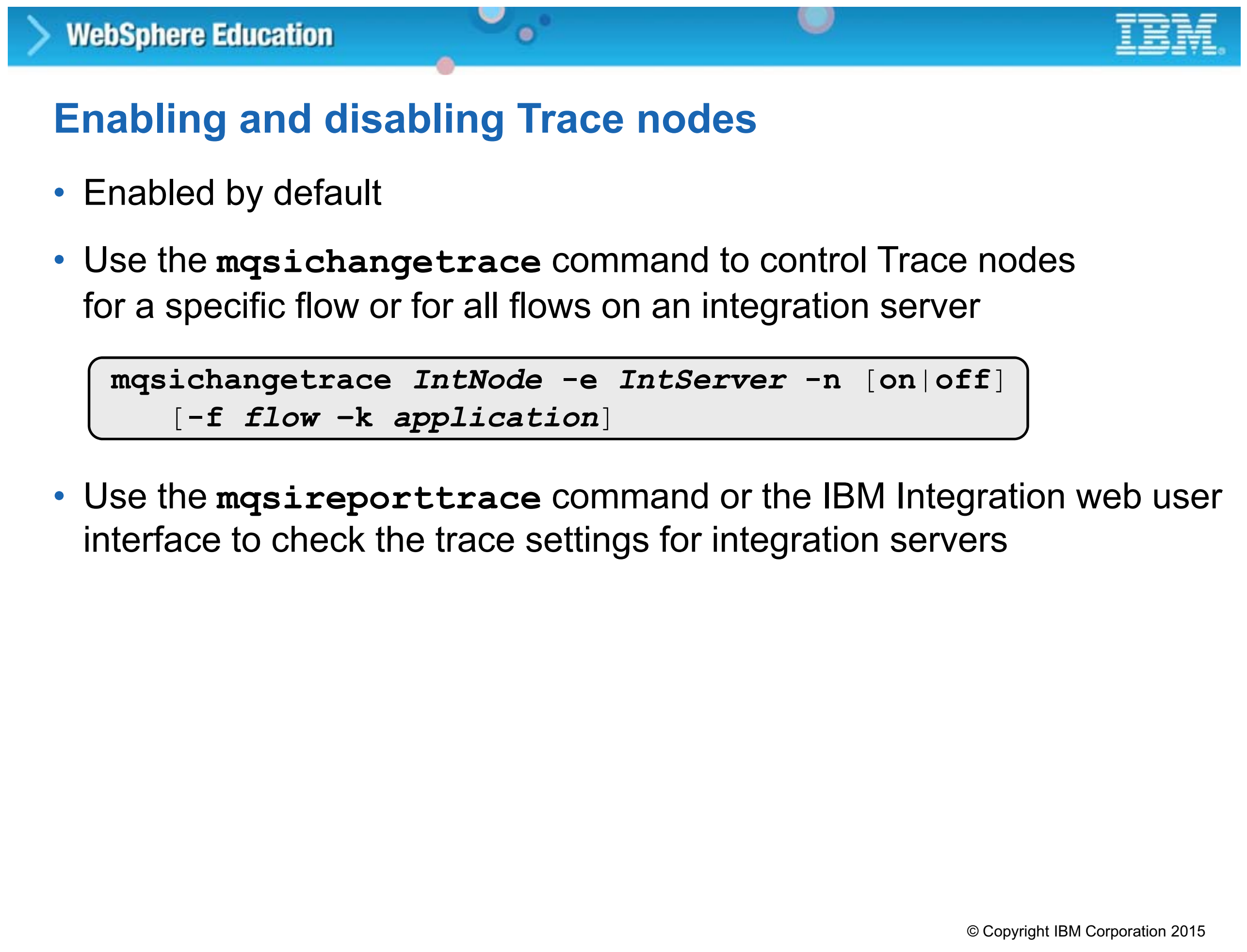
Notes:

If the flow is processing large messages, it is better to inspect only small parts of the message that is running through the flow by using a Trace node.

The Trace node can be inserted at any point within a message flow to capture data. The information can be sent to the user trace, the local error log, or to a file on the integration node system. Select the option to write the trace results to a file in readable text. This format makes it easier and faster to see the isolated trace results.

The **Pattern** property on the Trace node is an ESQL pattern that specifies the information that is written to the location that is specified in the **Destination** and **File Path** properties. You can write plain text, which is copied into the trace record exactly as you entered it. You can identify parts of the message that the trace writes, by specifying the full field identifiers, which are enclosed within the characters `${` and `}`. If you want to record the entire message, specify `${Root}`. You can also use the ESQL functions to provide more information. For example, you can use the ESQL function CURRENT_DATE to record the date, time, or date and time at which the trace record was written.

Trace nodes are enabled and disabled by using the `mqsichangetrace` command.



Enabling and disabling Trace nodes

- Enabled by default
- Use the **`mqsichangetrace`** command to control Trace nodes for a specific flow or for all flows on an integration server

```
mqsichangetrace IntNode -e IntServer -n [on|off]
    [-f flow -k application]
```

- Use the **`mqsireporttrace`** command or the IBM Integration web user interface to check the trace settings for integration servers

© Copyright IBM Corporation 2015

Figure 9-22. Enabling and disabling Trace nodes WM6461.0

Notes:

Trace nodes are enabled and disabled by using the `mqsichangetrace` command.

If the Trace node setting for an integration server is “off”, all Trace nodes in its flows are disabled.

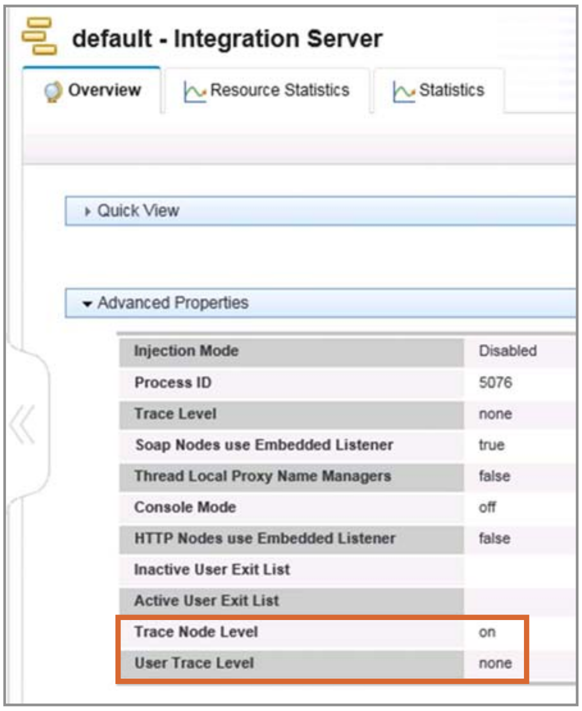
WebSphere Education IBM

Reporting current trace options for an integration node

- IBM Integration web user interface integration server status shows **Trace Node Level** and **User Trace Level** status
- `mqsireporttrace` reports the current trace current options on an active integration node
  - By default, lists all **active** user and service trace settings for the integration node
  - For status of User Trace, specify `-u`
  - For status of Trace node, specify `-n`

Examples:

```
mqsireporttrace IBNODE
mqsireporttrace IBNODE -u
mqsireporttrace IBNODE -n -e default
```



Injection Mode	Disabled
Process ID	5076
Trace Level	none
Soap Nodes use Embedded Listener	true
Thread Local Proxy Name Managers	false
Console Mode	off
HTTP Nodes use Embedded Listener	false
Inactive User Exit List	
Active User Exit List	
Trace Node Level	on
User Trace Level	none

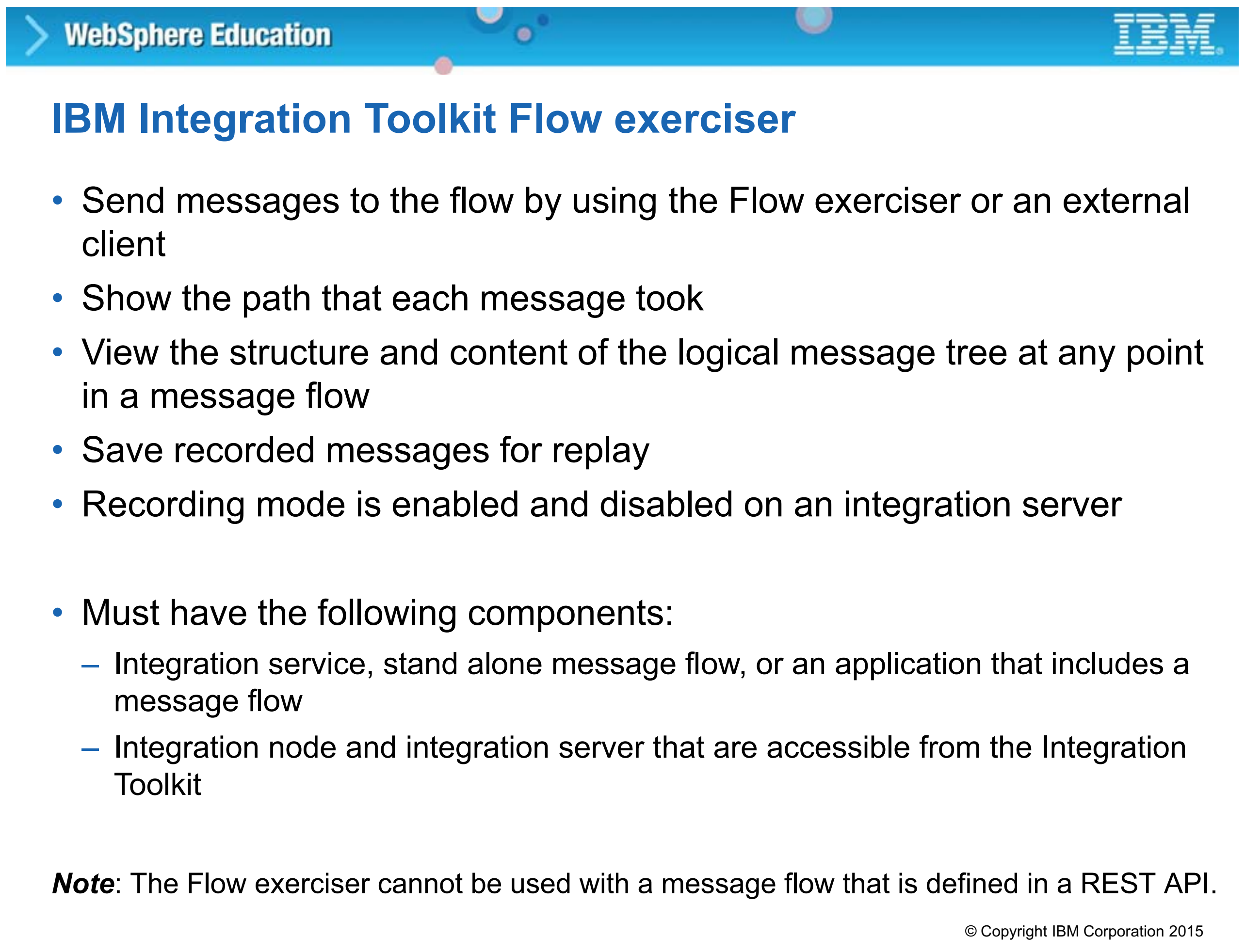
Figure 9-23. Reporting current trace options for an integration node WM6461.0

Notes:

You can use the IBM Integration web user interface or the `mqsireporttrace` command to view the current trace settings for an integration node or integration server.

The integration node name is a mandatory parameter on the `mqsireporttrace` command. If no other parameters are specified, the `mqsireporttrace` lists all **active** user and service trace settings for that integration node. Alternatively, you can use the following options for more specific reports on the current trace settings.

- Specify the `-u` parameter to view the status of a user trace.
- Specify the `-n` parameter to view the status of Trace nodes.
- Specify the `-t` parameter to view the status of a service trace.



WebSphere Education IBM

IBM Integration Toolkit Flow exerciser

- Send messages to the flow by using the Flow exerciser or an external client
- Show the path that each message took
- View the structure and content of the logical message tree at any point in a message flow
- Save recorded messages for replay
- Recording mode is enabled and disabled on an integration server

- Must have the following components:
  - Integration service, stand alone message flow, or an application that includes a message flow
  - Integration node and integration server that are accessible from the Integration Toolkit

Note: The Flow exerciser cannot be used with a message flow that is defined in a REST API.

© Copyright IBM Corporation 2015

Figure 9-24. IBM Integration Toolkit Flow exerciser WM6461.0

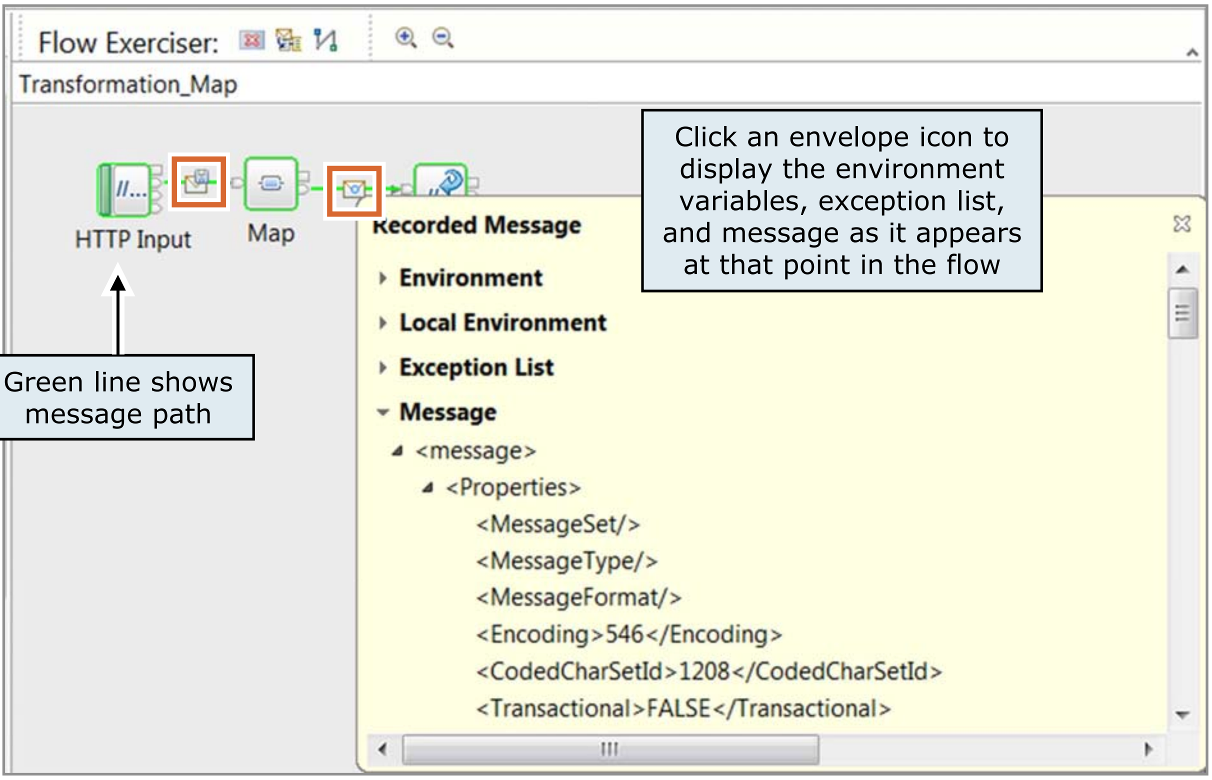
Notes:

The Flow exerciser is the integrated test client in the Integration Toolkit. It allows the developer to verify the flow of the message and examine the data contents before and after each node.

Before using the Flow exerciser, you must have the following components:

- A resource such as an integration service, a stand-alone message flow, or an application that includes a message flow.
- In integration node and associated integration server that are accessible from the IBM Integration Toolkit.

Flow exerciser example

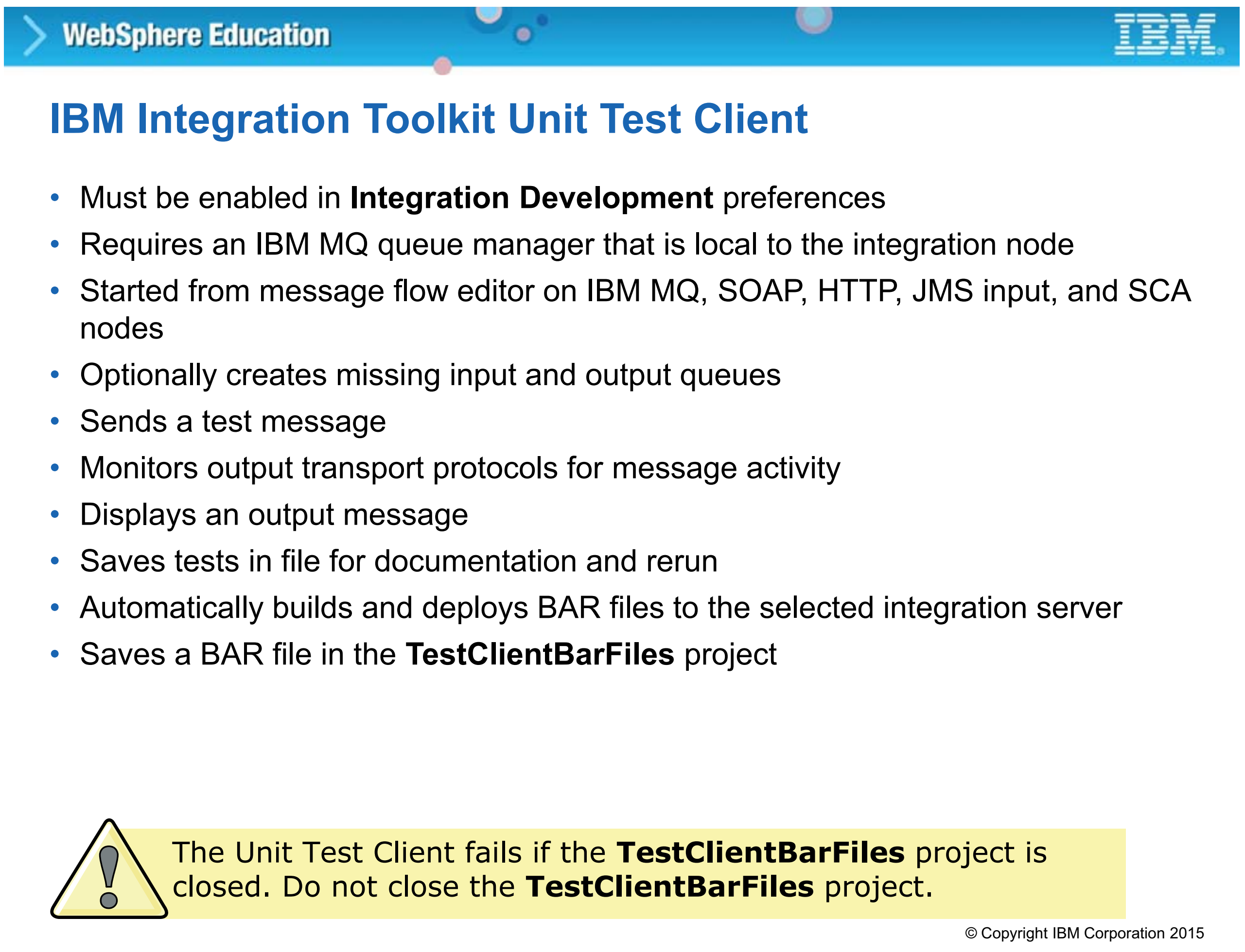

Flow Exerciser:
Transformation_Map
HTTP Input
Map
Click an envelope icon to display the environment variables, exception list, and message as it appears at that point in the flow
Green line shows message path
Recorded Message
Environment
Local Environment
Exception List
Message

```
<message>
  <Properties>
    <MessageSet/>
    <MessageType/>
    <MessageFormat/>
    <Encoding>546</Encoding>
    <CodedCharSetId>1208</CodedCharSetId>
    <Transactional>FALSE</Transactional>
```

Figure 9-25. Flow exerciser example WM6461.0

Notes:

The figure shows an example of the Flow exerciser. The green line shows the message path. You can click an envelope icon to display the message as it appears at that point in the message flow.



WebSphere Education IBM

IBM Integration Toolkit Unit Test Client

- Must be enabled in **Integration Development** preferences
- Requires an IBM MQ queue manager that is local to the integration node
- Started from message flow editor on IBM MQ, SOAP, HTTP, JMS input, and SCA nodes
- Optionally creates missing input and output queues
- Sends a test message
- Monitors output transport protocols for message activity
- Displays an output message
- Saves tests in file for documentation and rerun
- Automatically builds and deploys BAR files to the selected integration server
- Saves a BAR file in the **TestClientBarFiles** project

The Unit Test Client fails if the **TestClientBarFiles** project is closed. Do not close the **TestClientBarFiles** project.

© Copyright IBM Corporation 2015

Figure 9-26. IBM Integration Toolkit Unit Test Client WM6461.0

Notes:

The Integration Toolkit Unit Test Client can be used to test message flows that start with IBM MQ, SOAP, HTTP, JMS Input, and SCA Input nodes.

The Unit Test Client must be enabled in **Test Client** preferences.

1. Click **Window > Preference**
2. Expand the item for **Integration Development** on the left and click **Unit Test Client**

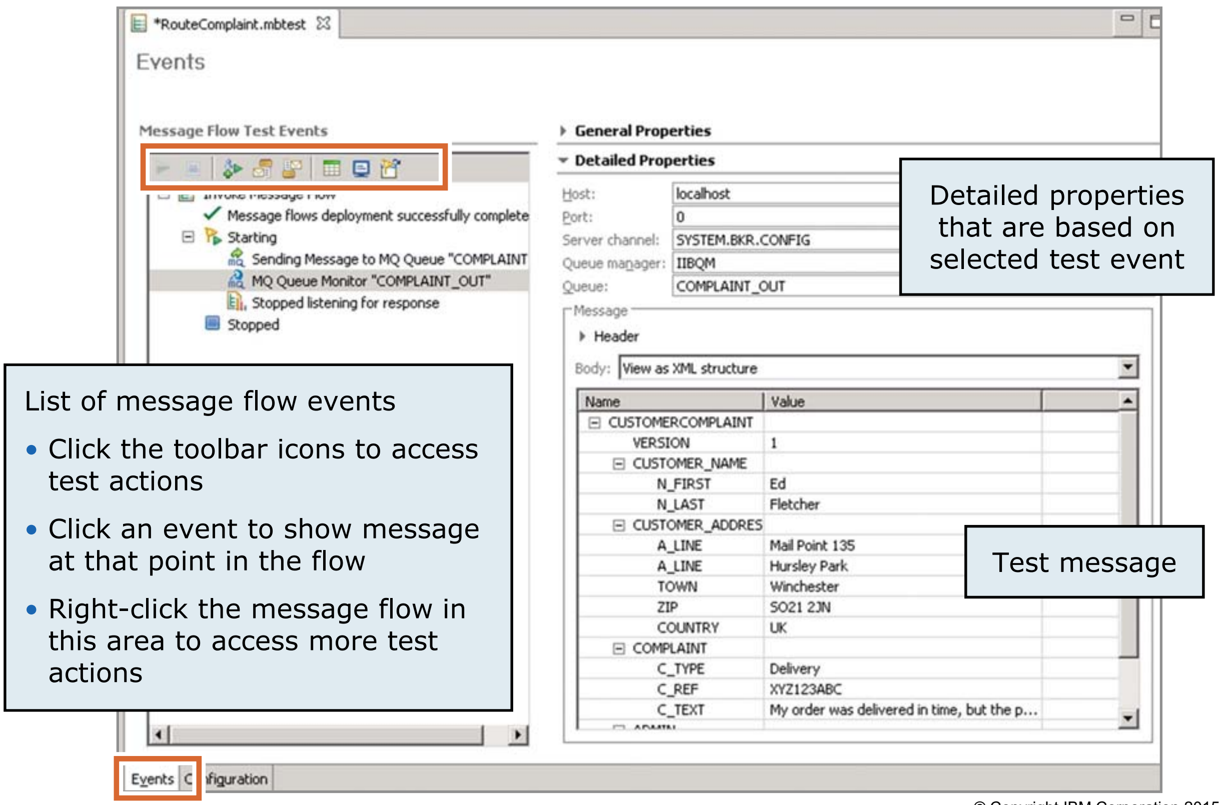
The Test Client does all the steps necessary to test a flow. You can test message flows in the context of an application or library, or you can test flows in isolation.

You can select the structure of an XML test message from the contents of message models and sets in your workspace and enter values for the elements. These values can be stored and reused in later test runs.

The Unit Test Client monitors output nodes in the message flow so that you can see which nodes output messages are received on. When an error message is produced as the message passes along the flow, or when a message is received on an output node, a test event is recorded in the Test Client.

WebSphere Education

Unit Test Client events


*RouteComplaint.mbtest

Events

Message Flow Test Events

- Invoke Message Flow
  - Message flows deployment successfully complete
  - Starting
    - Sending Message to MQ Queue "COMPLAINT
    - MQ Queue Monitor "COMPLAINT_OUT"
    - Stopped listening for response
  - Stopped

General Properties

Detailed Properties

Host:	localhost
Port:	0
Server channel:	SYSTEM.BKR.CONFIG
Queue manager:	IIBQM
Queue:	COMPLAINT_OUT

Message

Header

Body: View as XML structure

Name	Value
CUSTOMERCOMPLAINT	
VERSION	1
CUSTOMER_NAME	
N_FIRST	Ed
N_LAST	Fletcher
CUSTOMER_ADDRES	
A_LINE	Mail Point 135
A_LINE	Hursley Park
TOWN	Winchester
ZIP	SO21 2JN
COUNTRY	UK
COMPLAINT	
C_TYPE	Delivery
C_REF	XYZ123ABC
C_TEXT	My order was delivered in time, but the p...

Events | Configuration

Detailed properties that are based on selected test event

List of message flow events

- Click the toolbar icons to access test actions
- Click an event to show message at that point in the flow
- Right-click the message flow in this area to access more test actions

Test message

© Copyright IBM Corporation 2015

Figure 9-27. Unit Test Client events WM6461.0

Notes:

To start the Unit Test Client, right-click the input node of the flow you are testing and click **Test** from the menu. Clicking **Test** opens the Unit Test Client on the **Events** tab, which is shown in the figure.

While running the Unit Test Client, the list of message flow events is shown on the left side of the view. The right side of the view contains the event properties and information about the test message. You can also edit the properties and content of your test messages.

Several icons are provided on the upper right of the **Events** tab:

- **Invoke** starts a new “Invoke Message Flow” event where you can enter a request message and start the test.
- **Enqueue** puts the message to the specified queue manager, queue, port, and host name as defined in the **Detailed Properties** section on the right of the page.
- **Dequeue** gets the message from the specified queue manager, queue, port, and host name as defined in the **Detailed Properties** section on the right of the page.
- **Saved Message** displays the Data Pool editor in which you can select values that you used in a previous test session.

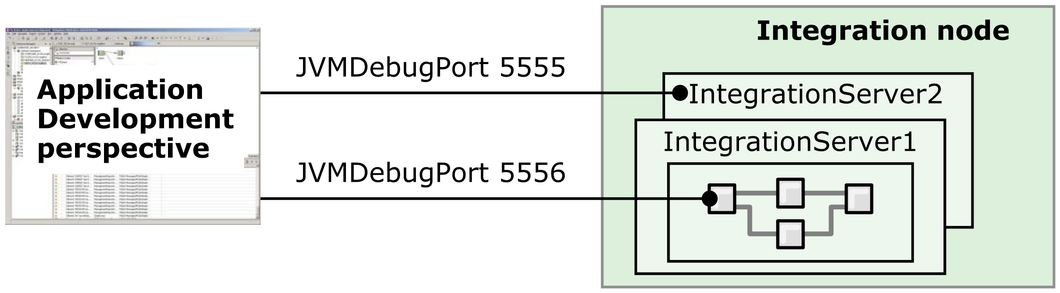
- **Stop** stops the current test.
- **Show Event Viewer** displays the Event Viewer if the operating system is Windows.
- **Show Console** opens the Integration Bus runtime console view. This view shows more details of the test run.

More actions can be initiated on the **Events** tab, by right-clicking on the message flow in **Message Flow Test events** area:

- **Re-run** clones and reruns a previously started test message.
- **Duplicate** duplicates the current message. To duplicate a previously started test message, right-click the message flow and click **Duplicate**.
- **Invoke** restarts the current message.

WebSphere Education

Message flow debugger



- Eclipse-based integrated debugging tool attaches to integration servers
  - Can attach to multiple integration servers at the same time, allowing for debugging of multiple flows in multiple integration servers from the same Integration Toolkit
  - Only one Integration Toolkit can debug one integration server at a time
- Debugs flows, subflows, ESQL, Java code, and maps
- A message flow must:
  - Be deployed and running
  - Be open in the Message Flow editor
  - Have at least one breakpoint that is defined in the flow
- Developer must use the Unit Test Client or a third-party tool to send a message to the flow

Figure 9-28. Message flow debugger WM6461.0

Notes:

The message flow debugger provides a convenient, easy-to-use graphical interface for flow testing. You can debug a wide variety of error conditions in flows, including:

- Nodes that are wired incorrectly (for example, outputs that are connected to the wrong inputs)
- Incorrect conditional branching in transition conditions
- Unintended loops in flow

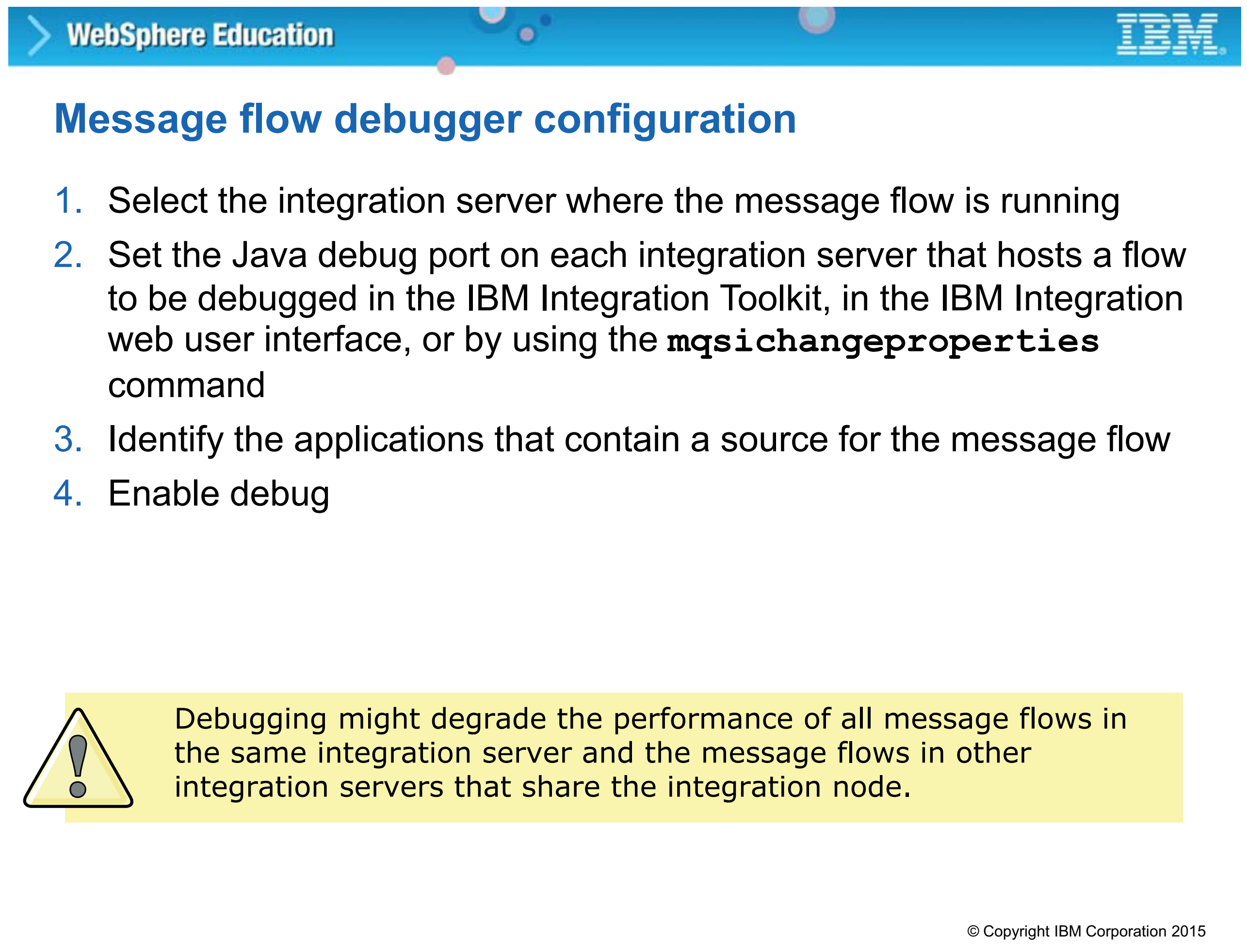
The debugger is an excellent tool for normal debugging scenarios. Its main advantage is that special skills or authorizations are not needed on the operating system of integration nodes; however, it cannot do everything.

The debugger does not always give an accurate representation of the message tree; a Trace node is always accurate, and it can optionally write its output to the user trace. Make user trace the standard tool for diagnosis.

Warning

Do not use the debugger on messages greater than 1 MB, as you might experience performance problems in the workspace because of excessive memory requirements.

When you debug message flows, use an integration node that is not being used in a production environment. Debugging might degrade the performance of all message flows in the same integration server and those flows in other integration servers that use the same integration node because of potential resource contention.



WebSphere Education IBM

Message flow debugger configuration

1. Select the integration server where the message flow is running
2. Set the Java debug port on each integration server that hosts a flow to be debugged in the IBM Integration Toolkit, in the IBM Integration web user interface, or by using the **`mqsichangeproperties`** command
3. Identify the applications that contain a source for the message flow
4. Enable debug

Debugging might degrade the performance of all message flows in the same integration server and the message flows in other integration servers that share the integration node.

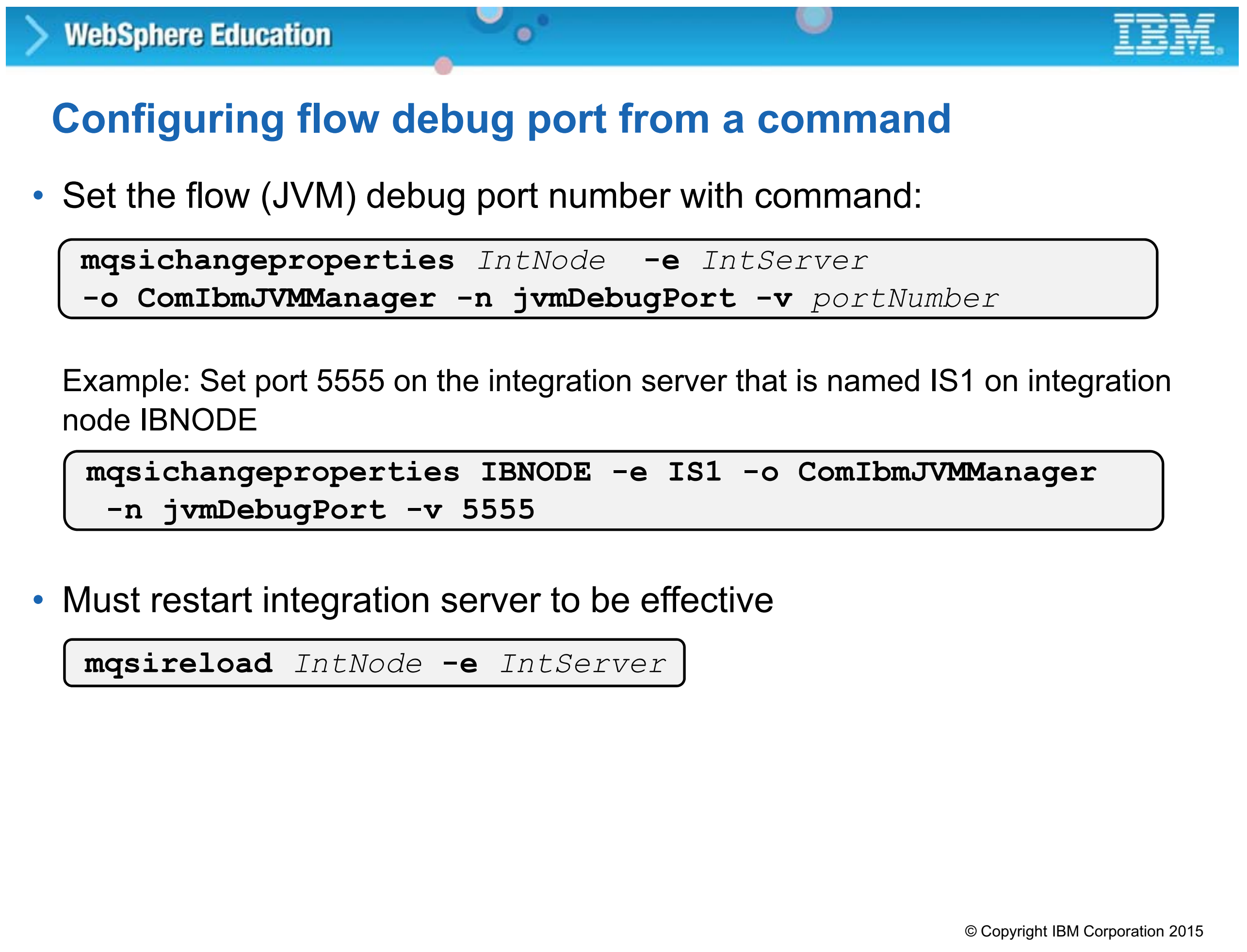
Figure 9-29. Message flow debugger configuration WM6461.0

Notes:

The message flow debugger attaches to the integration server that contains the message flow that you want to debug. Some one-time configuration is required to use the debugger:

- The flow must be deployed to an integration server.
- The Java debug port must be set on each integration server that hosts a flow to be debugged. For the debug port to be active, the integration server (or the integration node) must be restarted. If you set the debugger port in the Integration Toolkit **Integrations Node** view, the integration server is restarted automatically.
- Debugger configuration must select the integration server where the flow is running.
- Debugger configuration must point at the projects that contain source for the message flow.

The debug port can be set by using the Integration Toolkit or by using the `mqsichangeproperties` command.



WebSphere Education

Configuring flow debug port from a command

- Set the flow (JVM) debug port number with command:

```
mqsichangeproperties IntNode  -e IntServer
-o ComIbmJVMManager -n jvmDebugPort -v portNumber
```

Example: Set port 5555 on the integration server that is named IS1 on integration node IBNODE

```
mqsichangeproperties IBNODE -e IS1 -o ComIbmJVMManager
 -n jvmDebugPort -v 5555
```

- Must restart integration server to be effective

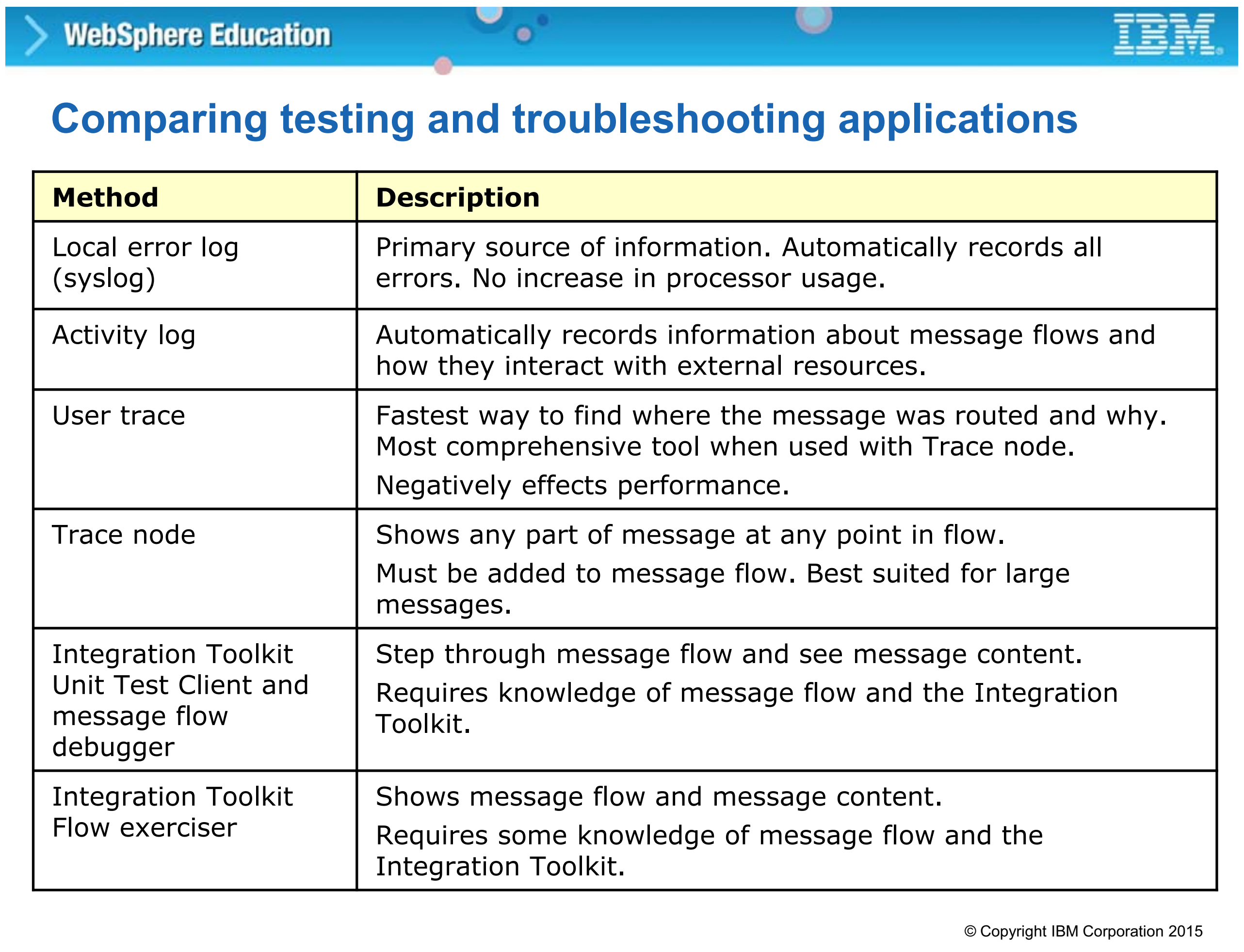
```
mqsireload IntNode -e IntServer
```

Figure 9-30. Configuring flow debug port from a command WM6461.0

Notes:

You can use the `mqsichangeproperties` command to configure the Java debug port. If the integration node run time is separated from the Integration Toolkit by a firewall, then configure the firewall to allow this port.

The property change becomes active after a restart of the integration server, or the entire integration node. You can use the `mqsireload` command, as shown in the figure, or the `mqsistop` and `mqsistart` commands.



WebSphere Education IBM

Comparing testing and troubleshooting applications

Method	Description
Local error log (syslog)	Primary source of information. Automatically records all errors. No increase in processor usage.
Activity log	Automatically records information about message flows and how they interact with external resources.
User trace	Fastest way to find where the message was routed and why. Most comprehensive tool when used with Trace node. Negatively effects performance.
Trace node	Shows any part of message at any point in flow. Must be added to message flow. Best suited for large messages.
Integration Toolkit Unit Test Client and message flow debugger	Step through message flow and see message content. Requires knowledge of message flow and the Integration Toolkit.
Integration Toolkit Flow exerciser	Shows message flow and message content. Requires some knowledge of message flow and the Integration Toolkit.

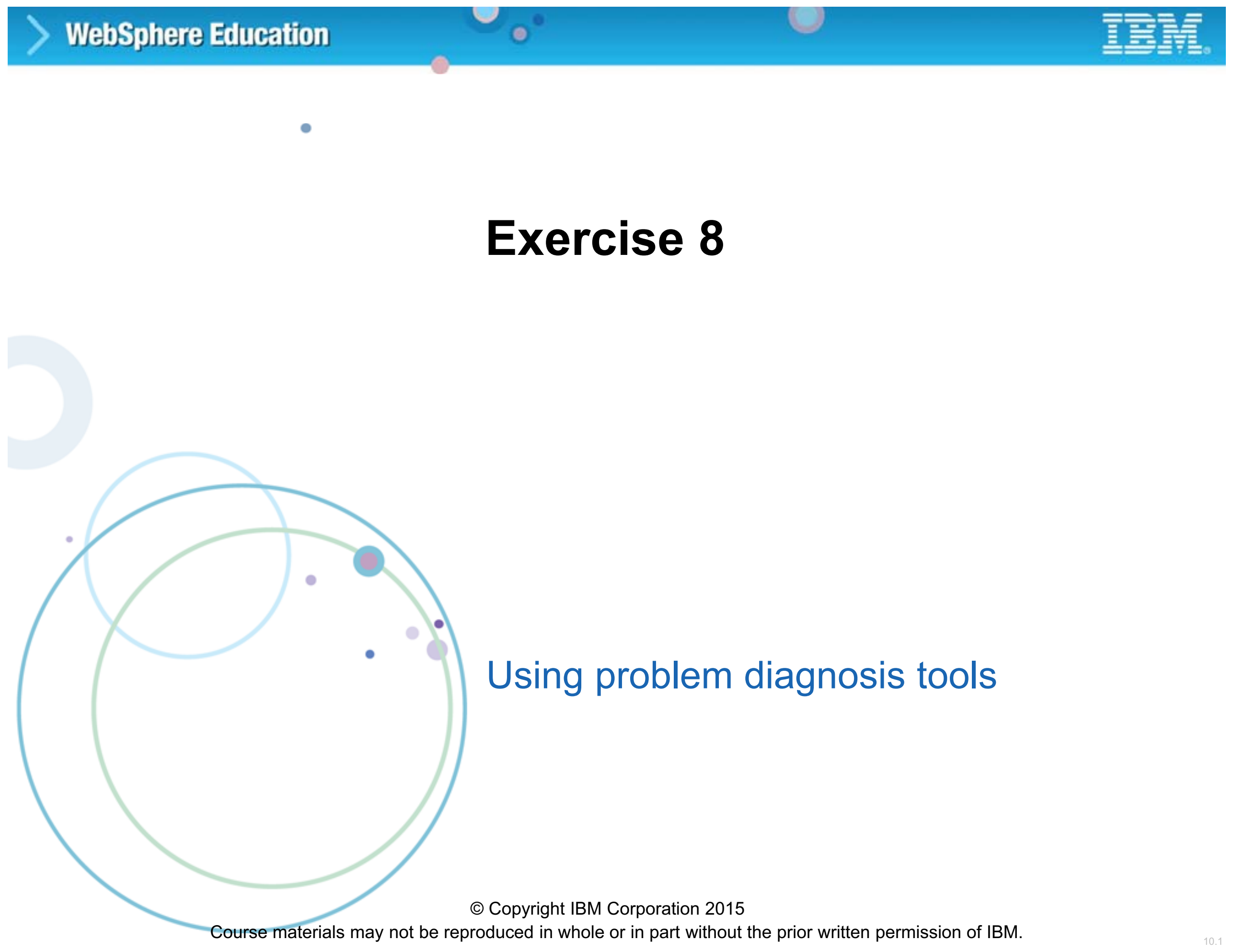
Figure 9-31. Comparing testing and troubleshooting applications WM6461.0

Notes:

The table summarizes testing and debugging applications. In many cases, one troubleshooting application is not sufficient to diagnose a problem; it might be necessary to use a combination of troubleshooting applications.

Exceptions are often longer than a single line, and they usually come in groups of two or three. User trace shows the full text, with inserts, of all the exceptions in the stack. Some parsing or modeling problems are hard to diagnose without this information.

The message flow debugger is an excellent tool for identifying problems in message flows. However, the Message flow debugger does assume an understanding of the message flow design and requires the Integration Toolkit. It is not the best choice for troubleshooting in a production environment.



WebSphere Education

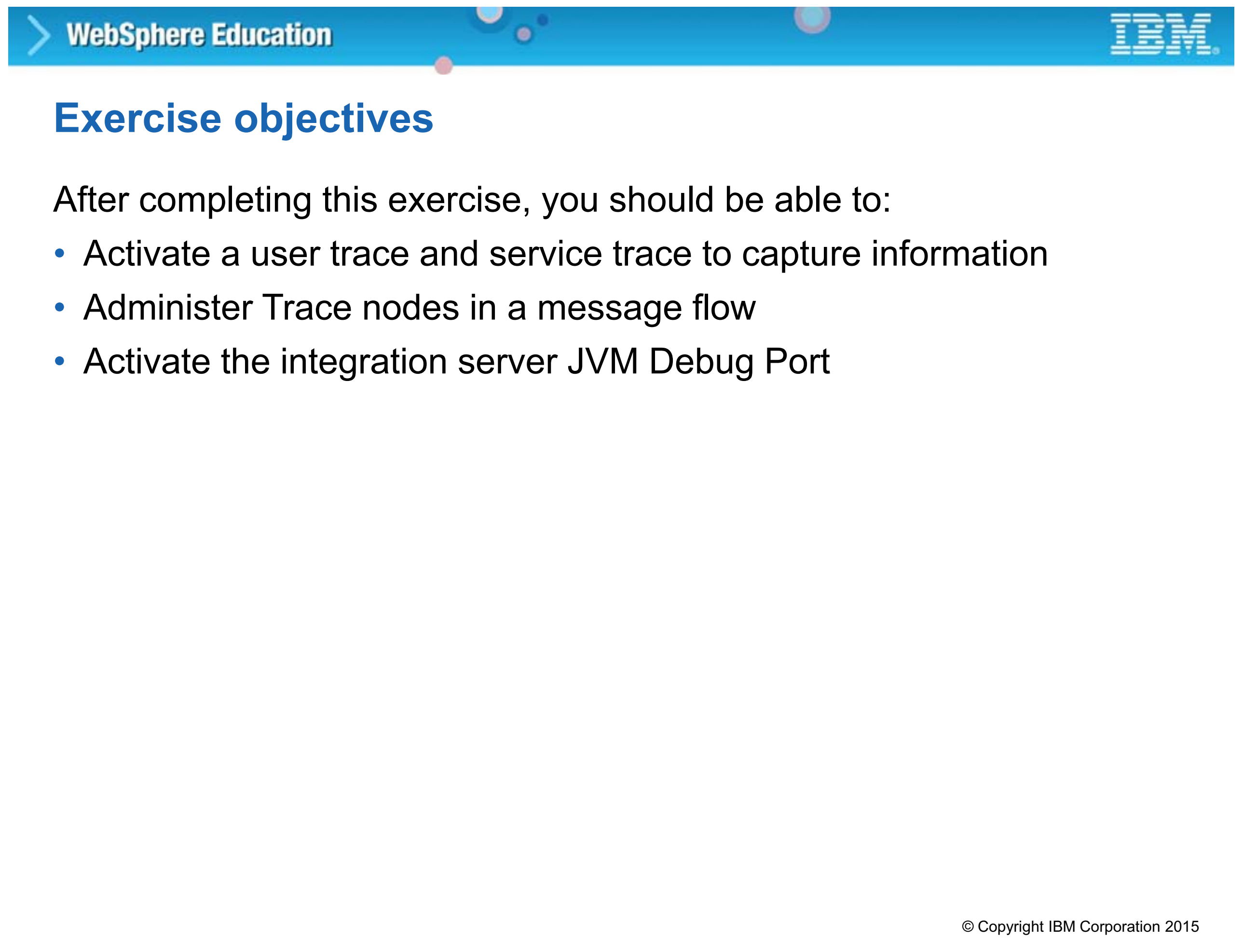
Exercise 8

Using problem diagnosis tools

Figure 9-32. Exercise 8 WM6461.0

Notes:

In this exercise, you use the Integration Bus problem diagnosis tools to identify problems in message flow applications. You enable and disable trace nodes, and use a user trace and service trace to capture runtime information. You also enable a debug port on an integration server.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Activate a user trace and service trace to capture information
- Administer Trace nodes in a message flow
- Activate the integration server JVM Debug Port

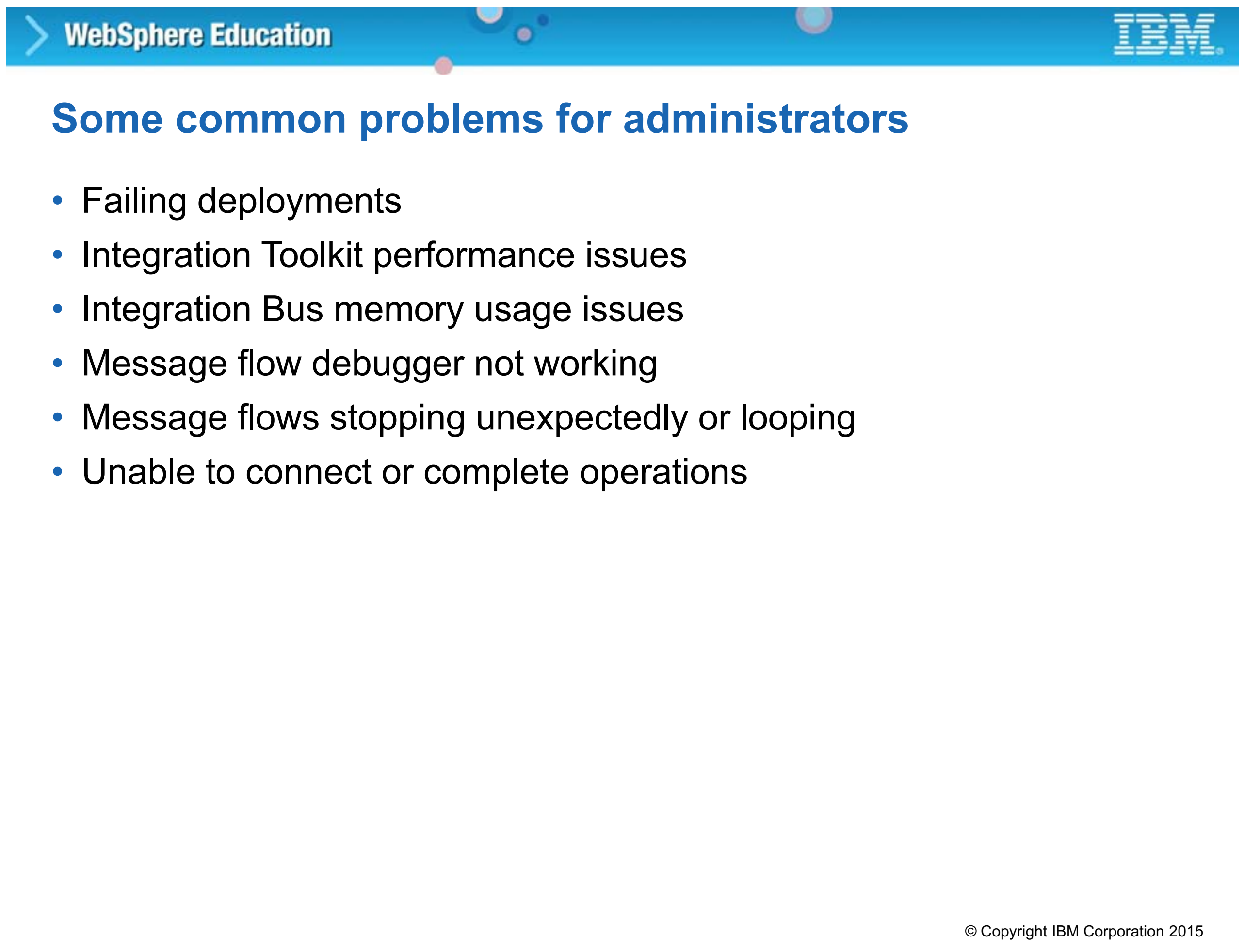
© Copyright IBM Corporation 2015

Figure 9-33. Exercise objectives WM6461.0

Notes:

See the *Student Exercise Guide* for detailed instructions.

9.3. Diagnosing common problems



WebSphere Education

Some common problems for administrators

- Failing deployments
- Integration Toolkit performance issues
- Integration Bus memory usage issues
- Message flow debugger not working
- Message flows stopping unexpectedly or looping
- Unable to connect or complete operations

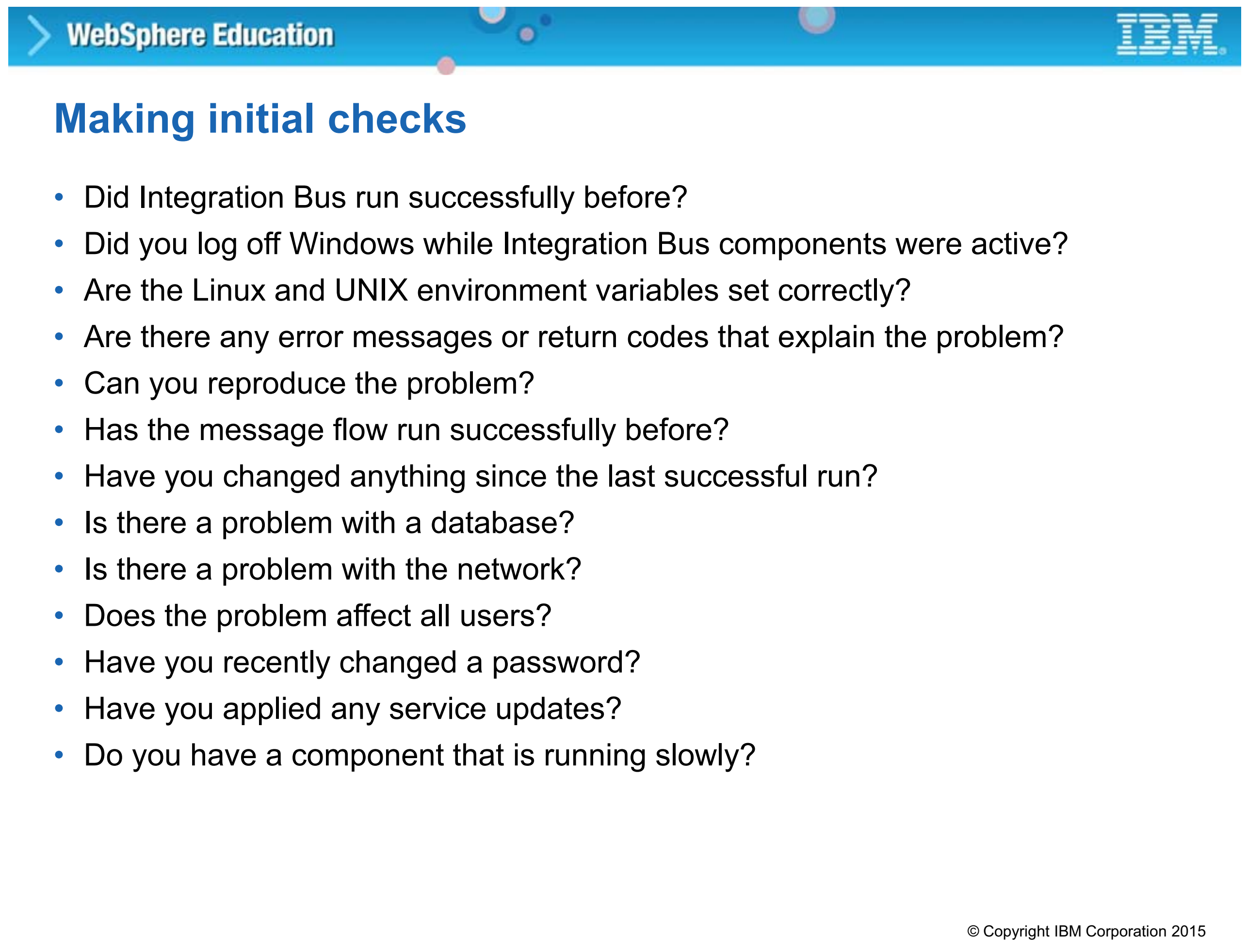
© Copyright IBM Corporation 2015

Figure 9-34. Some common problems for administrators WM6461.0

Notes:

The figure lists some common problems that an Integration Bus administrator might encounter.

One of the best resources for identifying common problems is the IBM Support Center. The information that is in this topic was excerpted from information that the Support Center generated to aid in identifying some of the most common problems and deciding how to approach resolving them.



WebSphere Education

IBM

Making initial checks

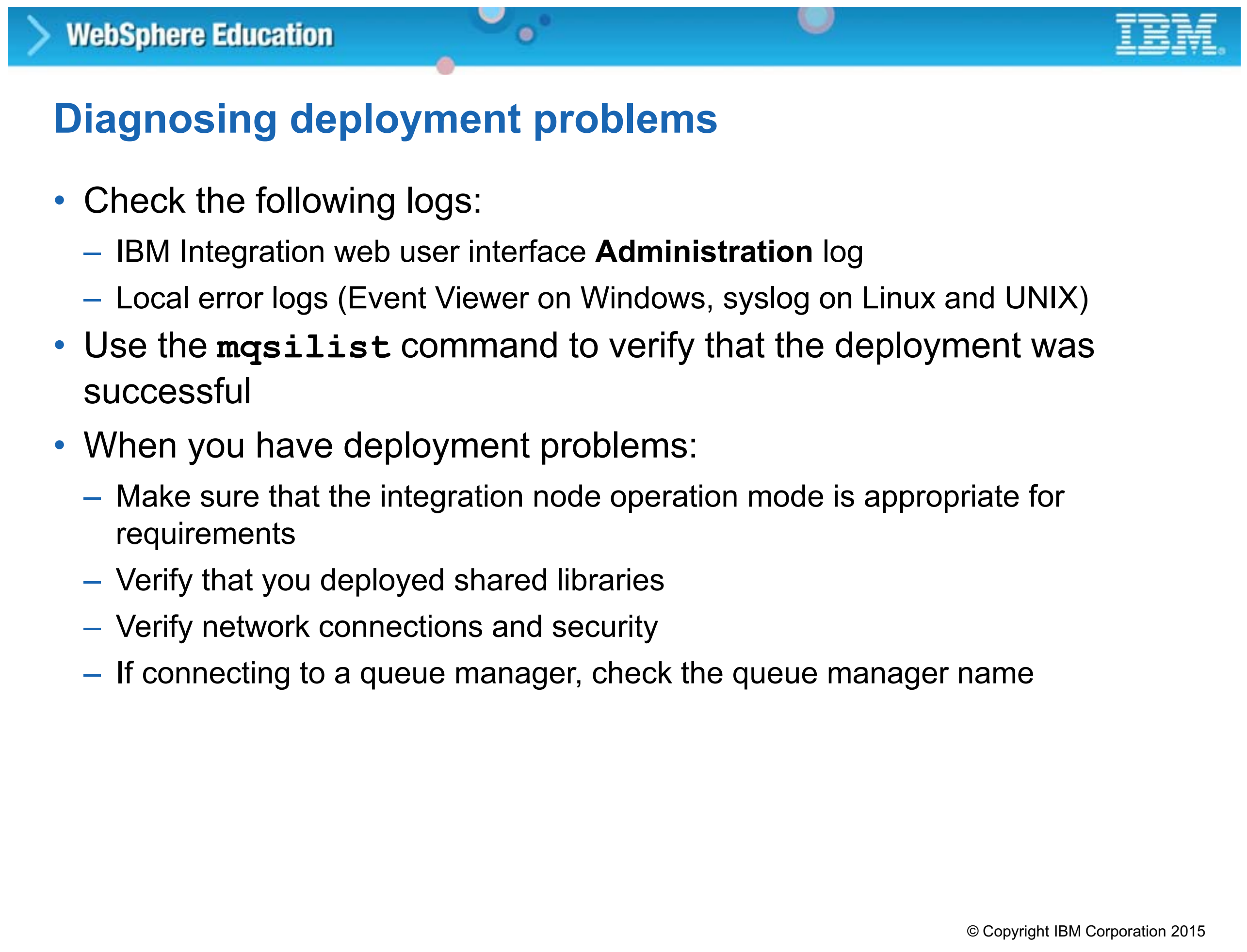
- Did Integration Bus run successfully before?
- Did you log off Windows while Integration Bus components were active?
- Are the Linux and UNIX environment variables set correctly?
- Are there any error messages or return codes that explain the problem?
- Can you reproduce the problem?
- Has the message flow run successfully before?
- Have you changed anything since the last successful run?
- Is there a problem with a database?
- Is there a problem with the network?
- Does the problem affect all users?
- Have you recently changed a password?
- Have you applied any service updates?
- Do you have a component that is running slowly?

Figure 9-35. Making initial checks WM6461.0

Notes:

Before you start problem determination in detail, consider whether there is an obvious cause of the problem, or an area of investigation that is likely to give useful results. This approach to diagnosis can often save much work by highlighting a simple error, or by narrowing down the range of possibilities.

The figure lists a number of questions you should ask when starting the problem determination process.



WebSphere Education

Diagnosing deployment problems

- Check the following logs:
  - IBM Integration web user interface **Administration** log
  - Local error logs (Event Viewer on Windows, syslog on Linux and UNIX)
- Use the **`mqsilist`** command to verify that the deployment was successful
- When you have deployment problems:
  - Make sure that the integration node operation mode is appropriate for requirements
  - Verify that you deployed shared libraries
  - Verify network connections and security
  - If connecting to a queue manager, check the queue manager name

© Copyright IBM Corporation 2015

Figure 9-36. Diagnosing deployment problems WM6461.0

Notes:

Some possible deployment errors are:

- No response
- Negative response
- Error on deployment

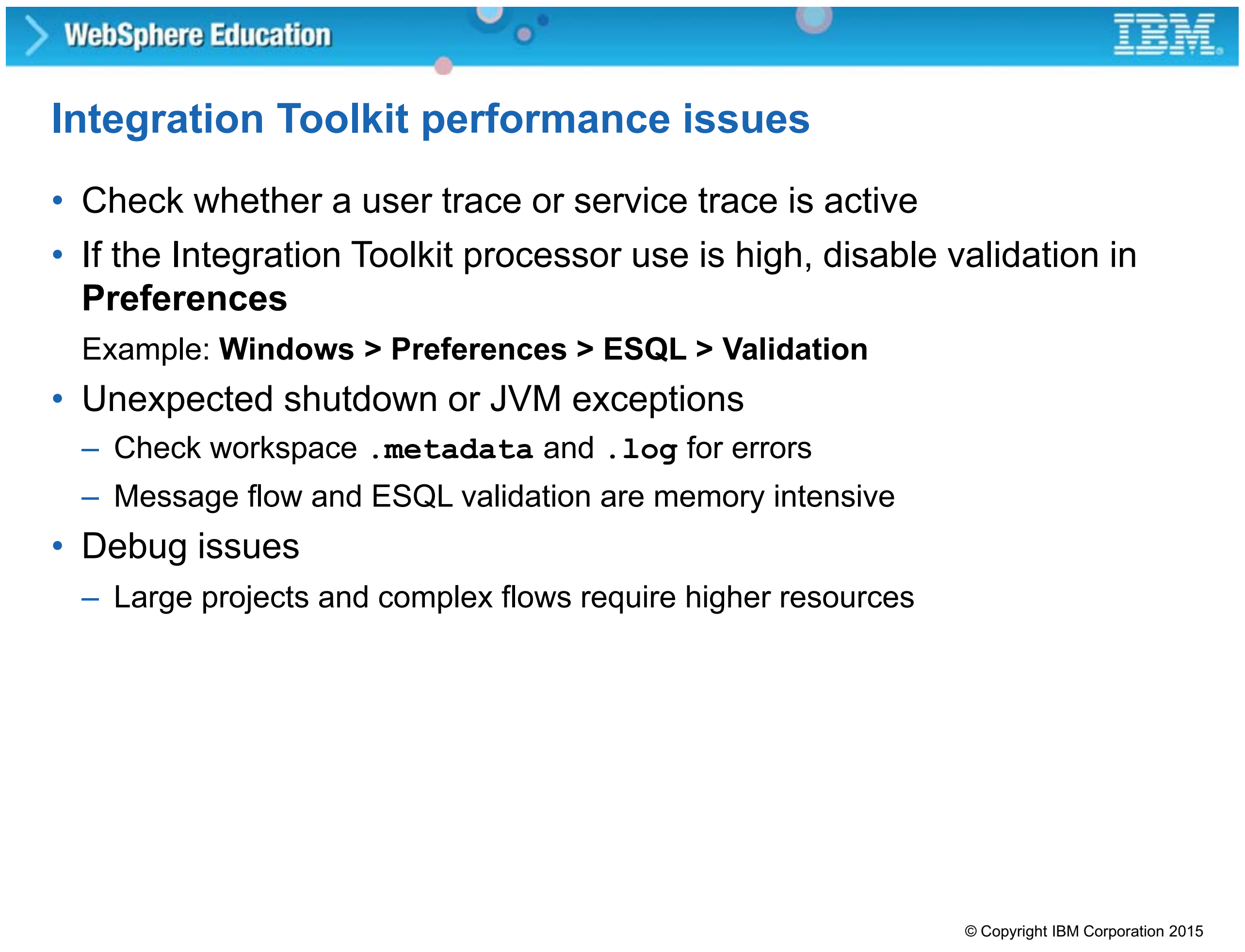
Sometimes it seems as if a deployment does not start. It is not a good idea to resubmit or try the deployment again because it can cause a series of deployments to stack up. If a deployment is not responding, check the local error logs and the **Administration** log.

Negative responses can be received if the user does not have the correct permissions. Check that the integration node service user ID and the user ID are authorized on the integration node system. It is also possible that the configuration timeout and **Configuration Delay Timeout** values are not adequate. These values can be set with the `mqsichangebroker` command.

You can view the progress of a deployment by setting the MQSI_SHOW_DEPLOY environment variable. If a deployment fails, check for operating system errors in the appropriate logs. This example shows how to set the environment variable on a UNIX system:

```
export MQSI_SHOW_DEPLOY=1
```

The integration node logs BIP8099I messages to the Windows Event Viewer or the system log at every logical step of the deployment operation. The integration node must be restarted to recognize the environment variable.



WebSphere Education

Integration Toolkit performance issues

- Check whether a user trace or service trace is active
- If the Integration Toolkit processor use is high, disable validation in **Preferences**

  Example: **Windows > Preferences > ESQL > Validation**
- Unexpected shutdown or JVM exceptions
  - Check workspace **`.metadata`** and **`.log`** for errors
  - Message flow and ESQL validation are memory intensive
- Debug issues
  - Large projects and complex flows require higher resources

Figure 9-37. Integration Toolkit performance issues WM6461.0

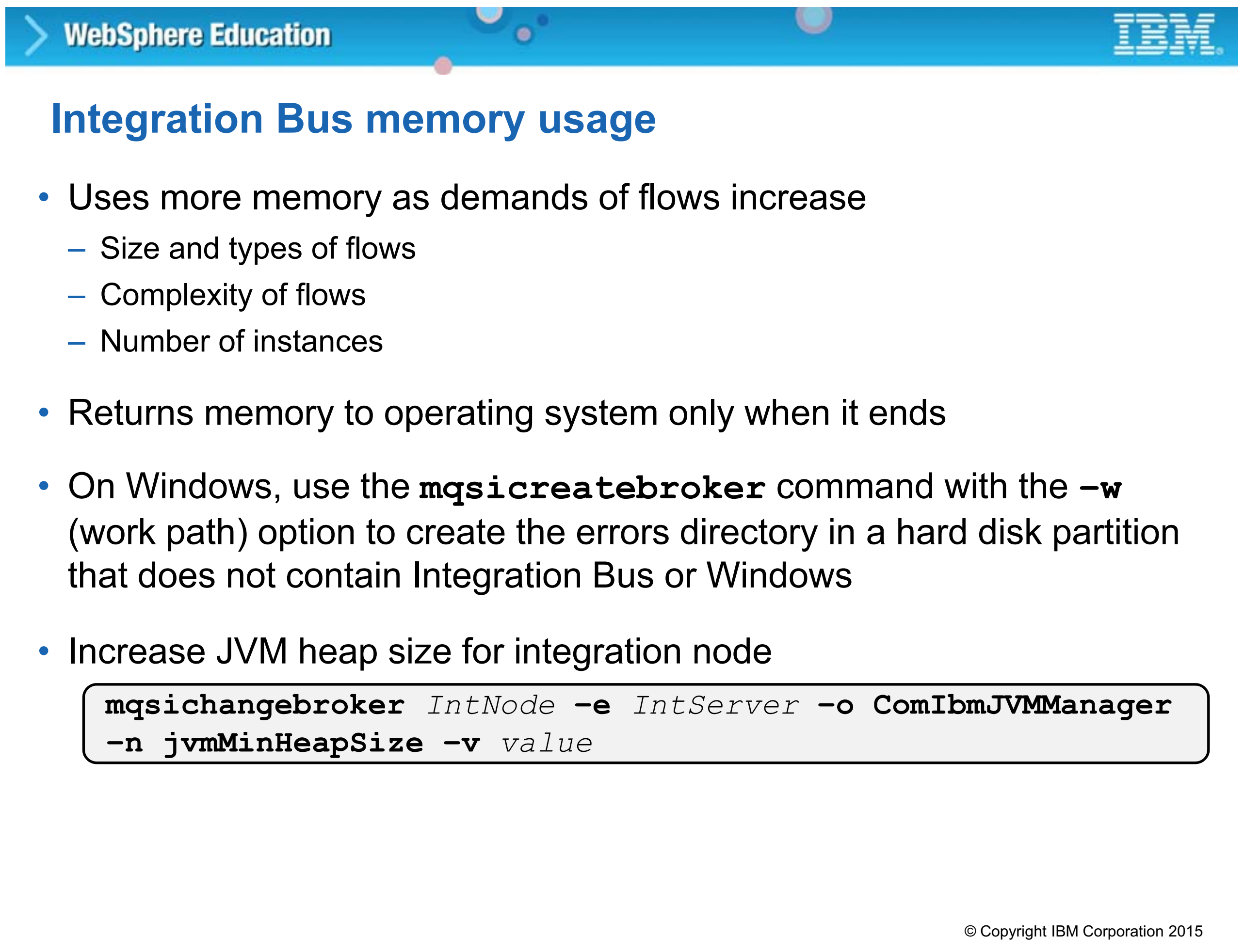
Notes:

If the Integration Toolkit seems to be slow when saving a project or when creating a BAR file, make sure that validation is turned off. Validation of ESQL can be memory-intensive.

Large projects, debugging complex flows, and testing large messages in the Integration Toolkit are resource-intensive; you might need to increase the size of the heap size that the Integration Toolkit uses. If the Integration Toolkit shuts down, it is possible it is related to a JVM exception; be sure to check the messages in the local error log.

The Integration Toolkit (`mb.exe`) creates the JVM with a default maximum heap size. The JVM default maximum heap size is half the real memory on the server. Increase the Integration Toolkit JVM maximum heap size by starting the Integration Toolkit from a command. For example:

`<ToolkitInstDir>\mb.exe -data <path>WorkspaceDir -vmargs –Xmx1024M` starts the toolkit with JVM storage of 1 GB.



Integration Bus memory usage

- Uses more memory as demands of flows increase
  - Size and types of flows
  - Complexity of flows
  - Number of instances
- Returns memory to operating system only when it ends
- On Windows, use the **`mqsicreatebroker`** command with the **`-w`** (work path) option to create the errors directory in a hard disk partition that does not contain Integration Bus or Windows
- Increase JVM heap size for integration node

```
mqsichangebroker IntNode -e IntServer -o ComIbmJVMManager
-n jvmMinHeapSize -v value
```

Figure 9-38. Integration Bus memory usage WM6461.0

Notes:

Each integration server starts one JVM and then uses it for all Java code that runs in all flows on that JVM. Each time a new message flow is added to an integration server, memory consumption is increased in that integration server. These integration servers have a finite limit on the number of threads each supports (255). As those threads are created, more memory is required.

If a particular flow is processing-intensive and must process larger volumes of messages, it might be better to deploy the flow to multiple integration servers, rather than increasing the number of instances within an integration server.

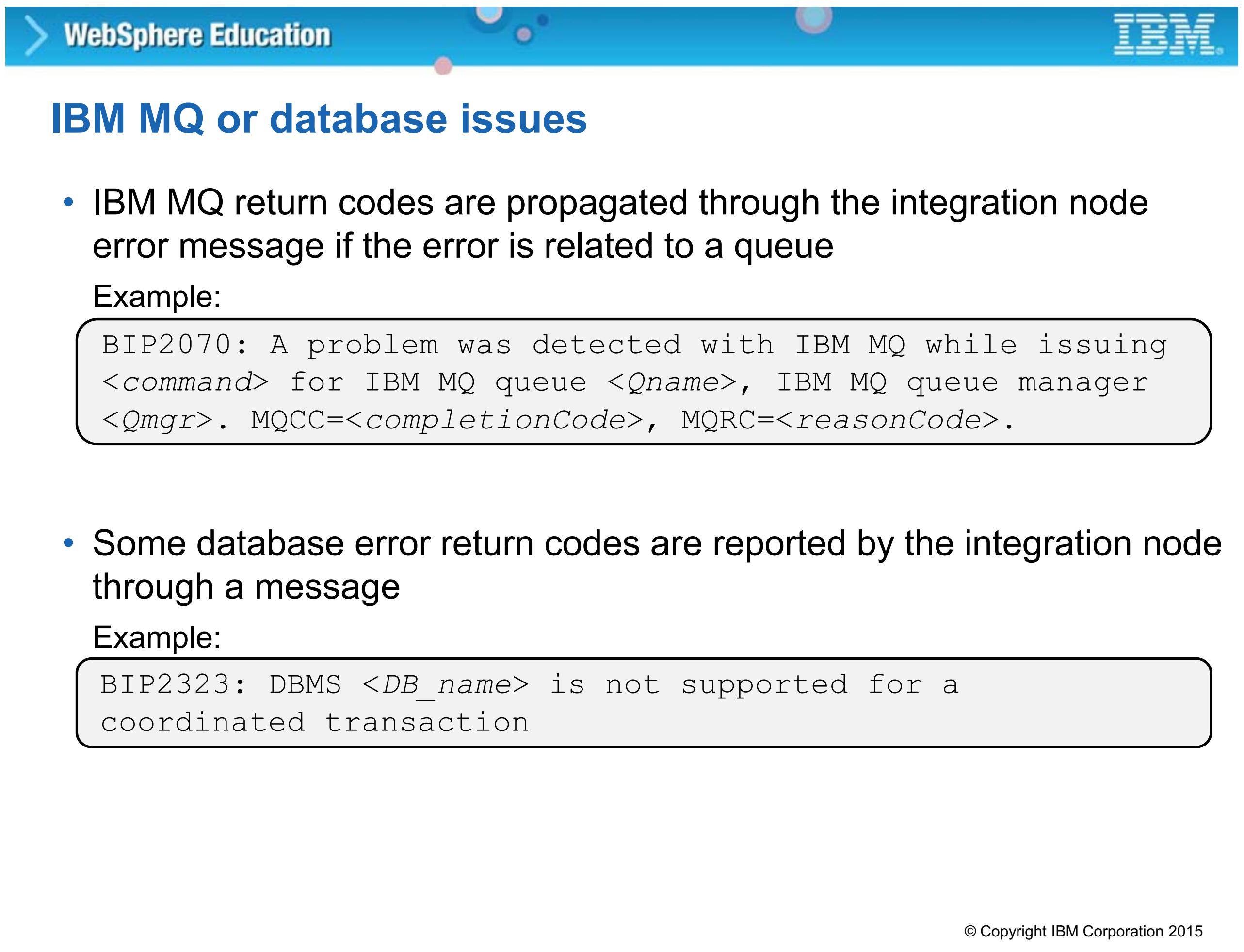
- A message flow sends a “free” on all its storage after completing message processing, but the “free” does not release memory to the operating system.
- Memory is returned to the operating system only upon termination of the application.
- The operating system does not retrieve storage from a process until termination; the memory is released on a process restart.

The integration server can be restarted with a command:

```
mqsireload <iNode> -e <iServer>
```

Here are some things to remember:

- The integration server requires increasing memory with an increased number of deployed messages. The memory requirements vary with the number of flows that are deployed, the types of messages, and message sizes processed. Memory increases when processing large messages.
- Message flows run as threads in a process and request memory for message processing.
- The integration server is a multithreaded process where many threads can request storage at the same time, causing memory requests to the operating system.



WebSphere Education

IBM MQ or database issues

- IBM MQ return codes are propagated through the integration node error message if the error is related to a queue

  Example:

```
BIP2070: A problem was detected with IBM MQ while issuing
<command> for IBM MQ queue <Qname>, IBM MQ queue manager
<Qmgr>. MQCC=<completionCode>, MQRC=<reasonCode>.
```

- Some database error return codes are reported by the integration node through a message

  Example:

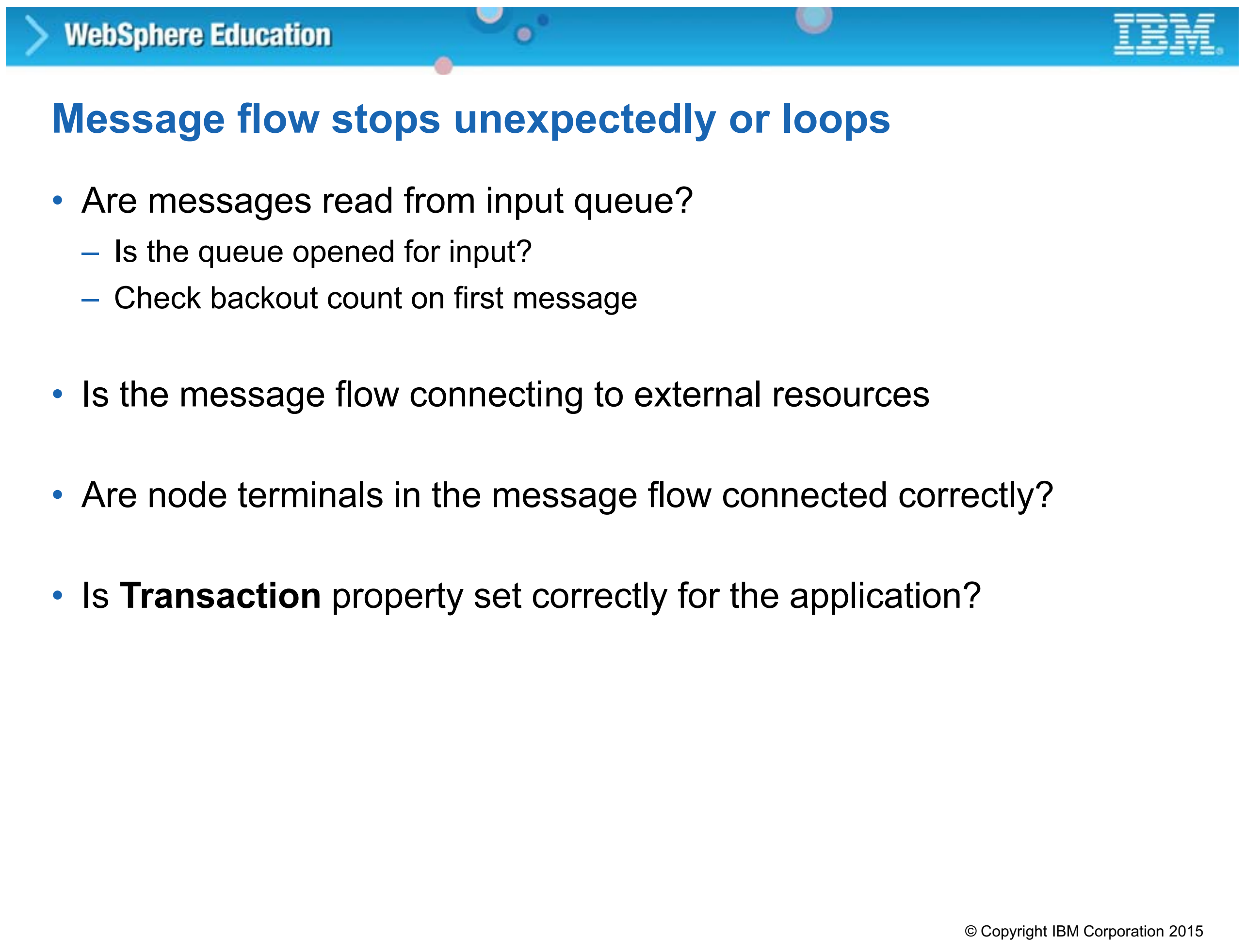
```
BIP2323: DBMS <DB_name> is not supported for a
coordinated transaction
```

© Copyright IBM Corporation 2015

Figure 9-39. IBM MQ or database issues WM6461.0

Notes:

Several IBM MQ errors can be reported when working with an integration node. One to be aware of is MQRC_OBJECT_IN_USE (MQRC 2042). This error can occur when altering a queue used in an MQInput node after stopping a message flow. Although the flow stops, the queue is not closed. It might be necessary to remove the message flow from the integration server so it no longer is in the runtime environment, change the queue attributes, and then redeploy the message flow.



WebSphere Education IBM

Message flow stops unexpectedly or loops

- Are messages read from input queue?
  - Is the queue opened for input?
  - Check backout count on first message
- Is the message flow connecting to external resources
- Are node terminals in the message flow connected correctly?
- Is **Transaction** property set correctly for the application?

© Copyright IBM Corporation 2015

Figure 9-40. Message flow stops unexpectedly or loops WM6461.0

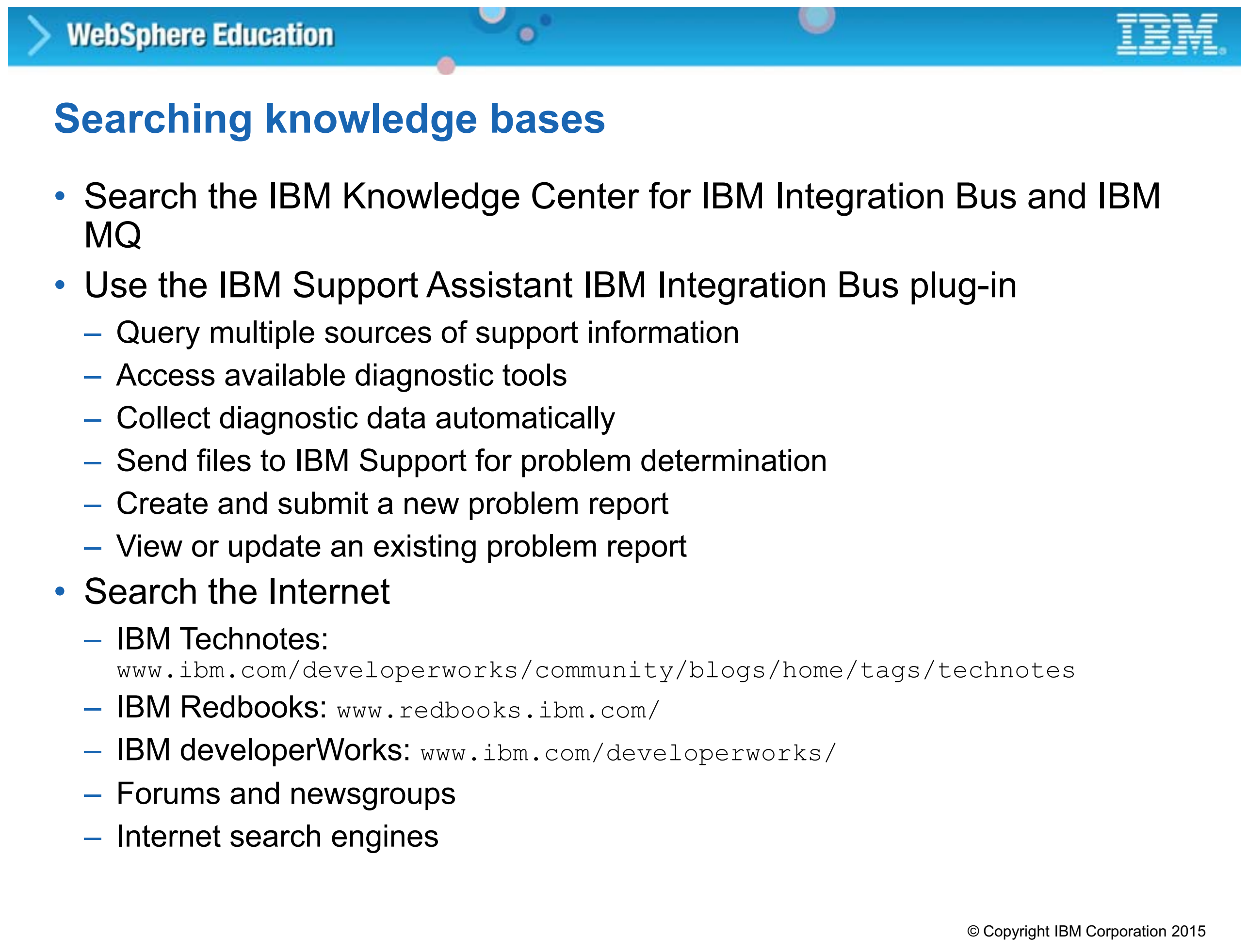
Notes:

You might encounter message flows that stop unexpectedly or appear to be stuck in a loop.

After deploying a message flow that contains an MQInput node, always check to see whether the input queue is opened for input. In some cases, the problem can be that the flow is not set up to use the specified queue. If there are messages in the input queue, look at the backout count of the first message. If it is more than 0, there is likely a problem where the message has no place to go; check the output queues to ensure that PUT calls are enabled. If messages disappear from the input queue, check the backout requeue queue or dead-letter queue if no explicit failure processing is in the flow. A flow might be in a loop, in which case you might need assistance from the developer. It can be detected by using a user trace.

Messages can also be discarded because a flow is not properly set up. If a message is transactional, it is not discarded; however, it is possible that incorrect (or missing) terminal connections can cause a flow to end at a point that was not intended.

The problem can also be that the deployment failed; so always check the results of a deployment in the IBM Integration web user interface **Admin Log**.



Searching knowledge bases

- Search the IBM Knowledge Center for IBM Integration Bus and IBM MQ
- Use the IBM Support Assistant IBM Integration Bus plug-in
  - Query multiple sources of support information
  - Access available diagnostic tools
  - Collect diagnostic data automatically
  - Send files to IBM Support for problem determination
  - Create and submit a new problem report
  - View or update an existing problem report
- Search the Internet
  - IBM Technotes: `www.ibm.com/developerworks/community/blogs/home/tags/technotes`
  - IBM Redbooks: `www.redbooks.ibm.com/`
  - IBM developerWorks: `www.ibm.com/developerworks/`
  - Forums and newsgroups
  - Internet search engines

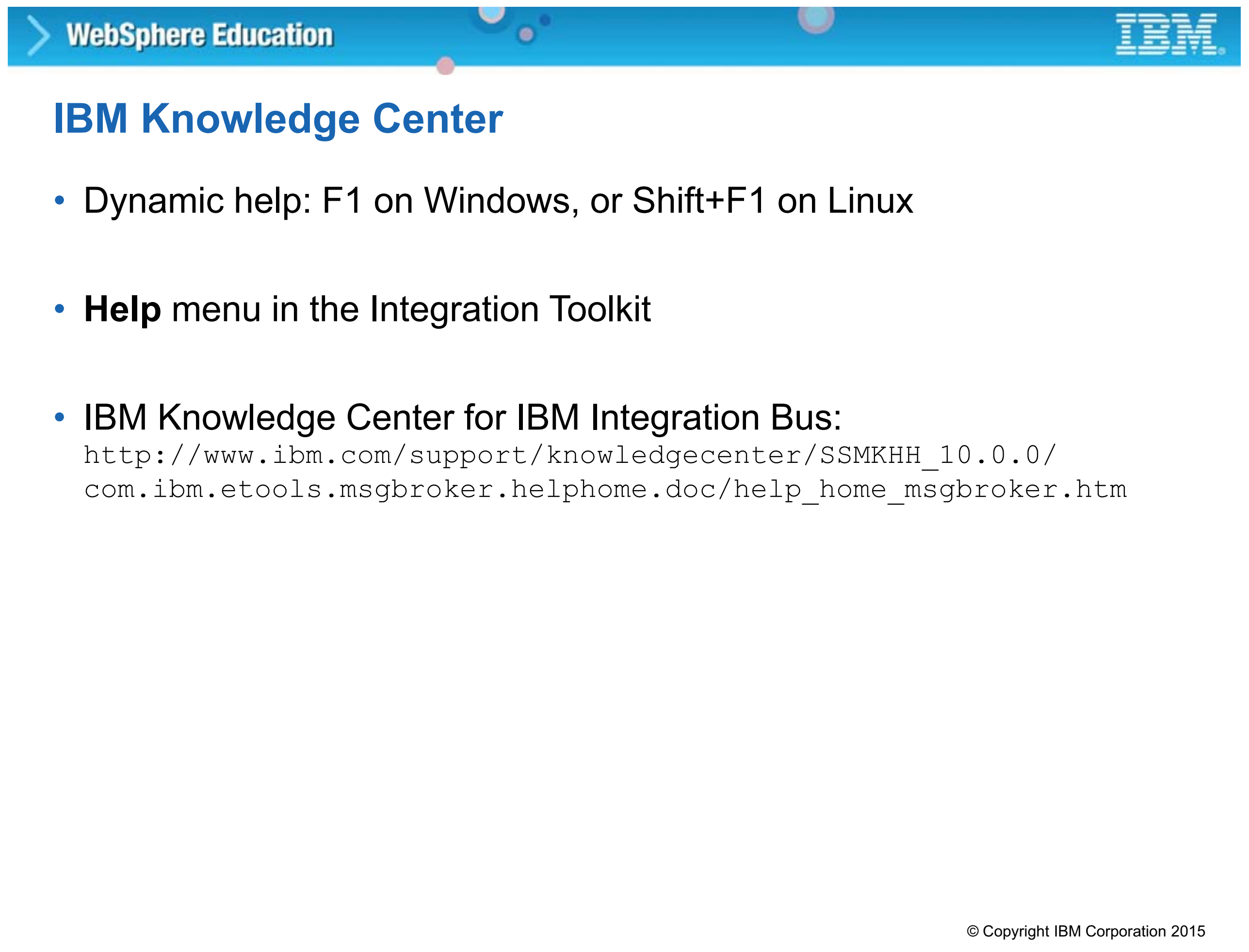
Figure 9-41. Searching knowledge bases WM6461.0

Notes:

If you have a problem with Integration Bus, you want to resolve it quickly. Begin by searching the available knowledge bases to determine whether the resolution to your problem is already documented.

IBM provides extensive documentation in the form of online IBM Knowledge Center, forums, and Redbooks.

IBM Support Assistant is available at no charge to install on your computer; you then install the relevant product add-on. IBM Support Assistant has a built-in user guide. The IBM Support Assistant download package includes a quick start installation and configuration guide. The IBM Support Assistant webpage URL is: `https://www.ibm.com/software/support/isa/`



WebSphere Education

IBM Knowledge Center

- Dynamic help: F1 on Windows, or Shift+F1 on Linux
- **Help** menu in the Integration Toolkit
- IBM Knowledge Center for IBM Integration Bus:
  `http://www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm`

© Copyright IBM Corporation 2015

Figure 9-42. IBM Knowledge Center WM6461.0

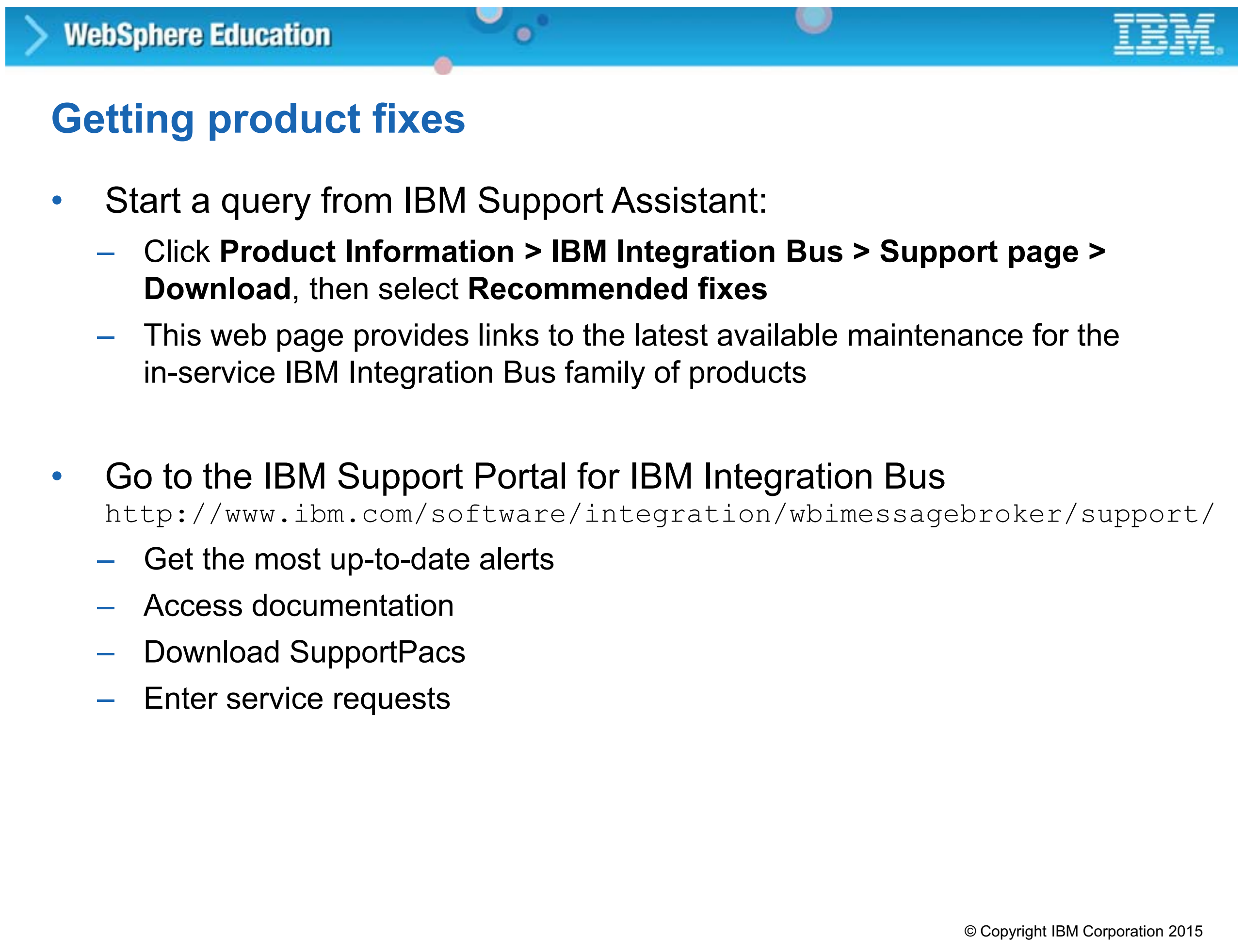
Notes:

The IBM Knowledge Center for IBM Integration Bus can be installed on your local workstation or on a local intranet.

The IBM Knowledge Center can also be viewed on the IBM website. You can use the powerful search function of the IBM Knowledge Center to query conceptual and reference information and detailed instructions for completing tasks.

You can also start the online IBM Knowledge Center for IBM Integration Bus V10 by using the following URL:

`http://www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/`
`com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm`



WebSphere Education

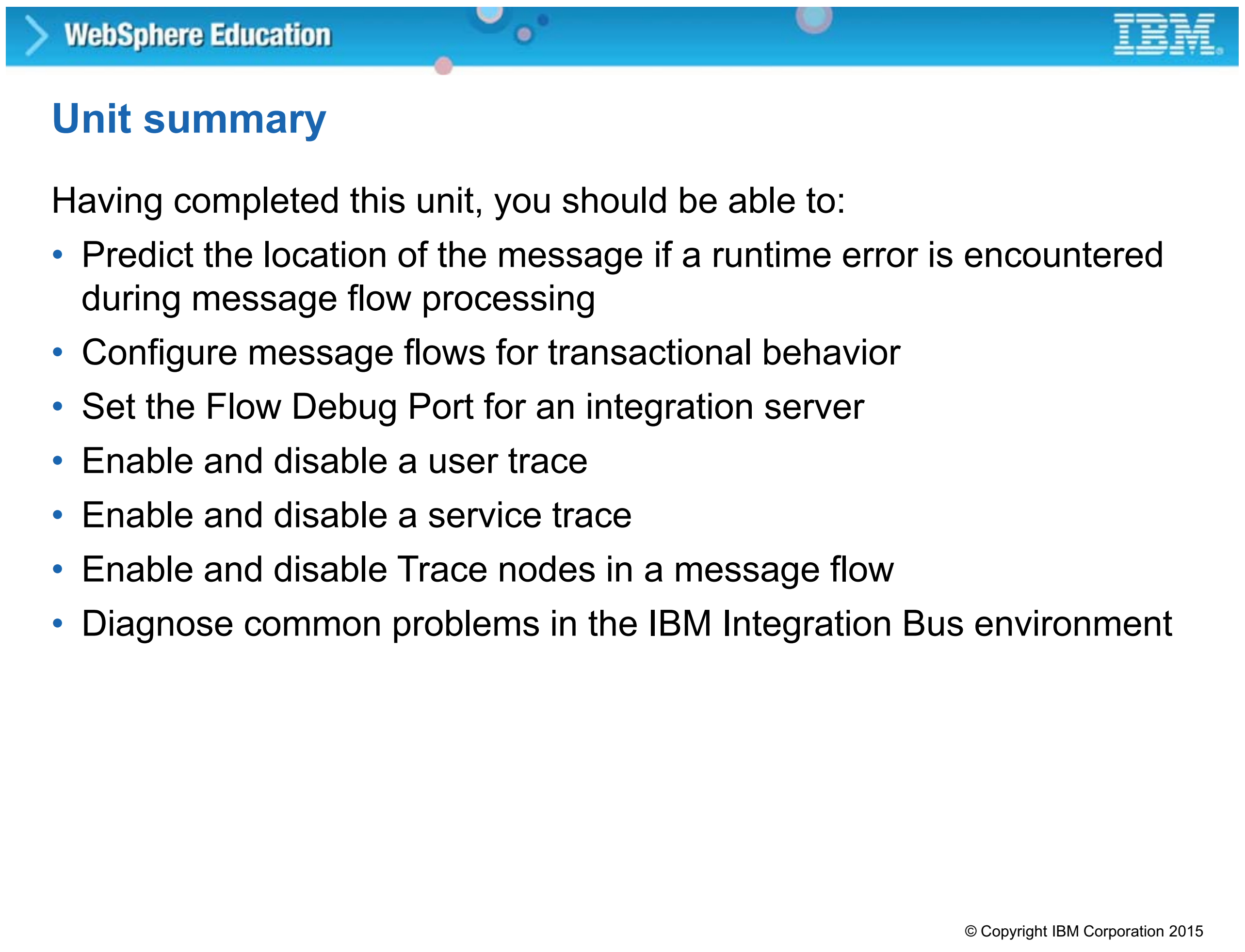
Getting product fixes

- Start a query from IBM Support Assistant:
  - Click **Product Information > IBM Integration Bus > Support page > Download**, then select **Recommended fixes**
  - This web page provides links to the latest available maintenance for the in-service IBM Integration Bus family of products
- Go to the IBM Support Portal for IBM Integration Bus
  `http://www.ibm.com/software/integration/wbimessagebroker/support/`
  - Get the most up-to-date alerts
  - Access documentation
  - Download SupportPacs
  - Enter service requests

Figure 9-43. Getting product fixes WM6461.0

Notes:

A product fix might be available to resolve your problem. You can determine what fixes are available by entering a query in the IBM Support Assistant.



WebSphere Education
IBM

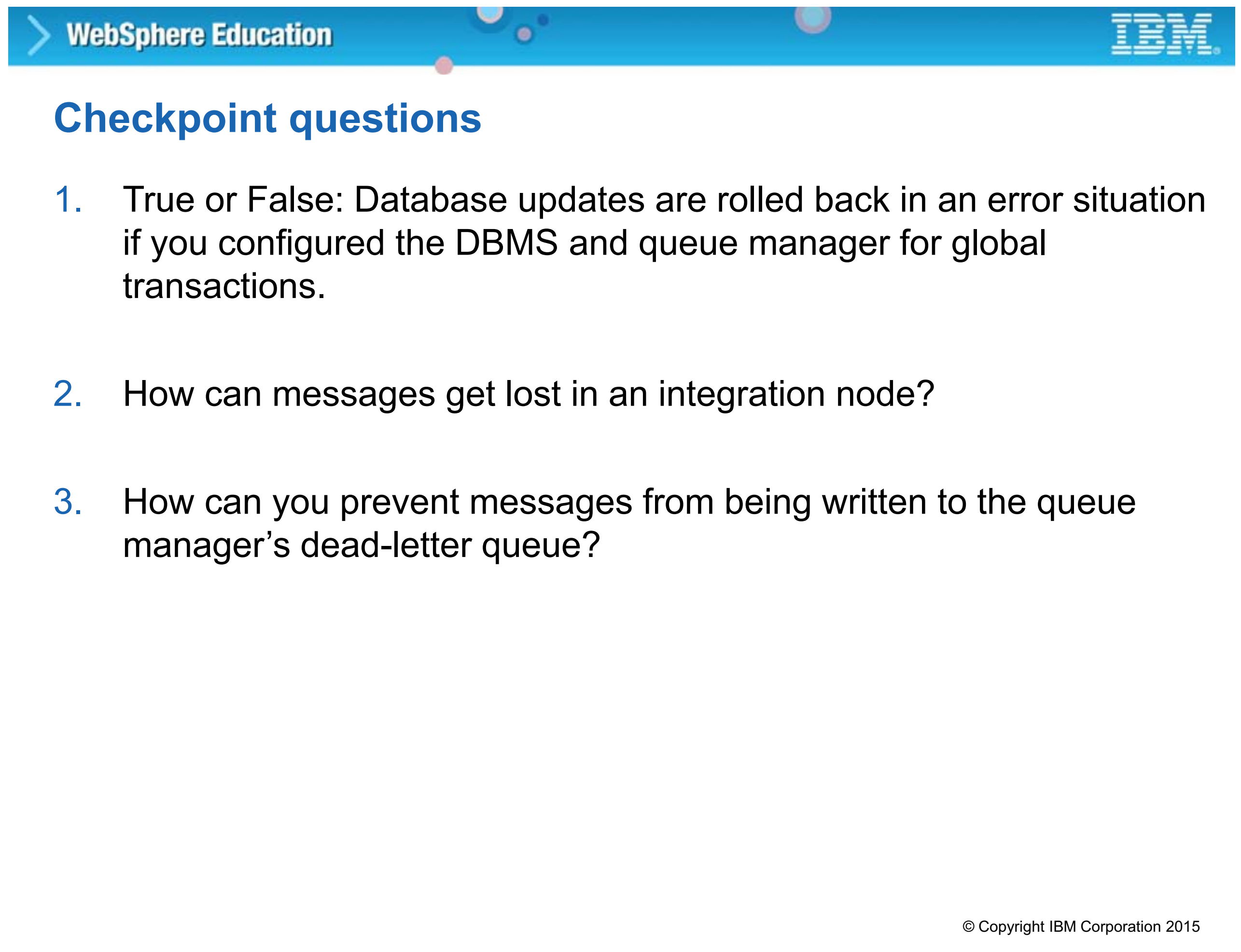
Unit summary

Having completed this unit, you should be able to:

- Predict the location of the message if a runtime error is encountered during message flow processing
- Configure message flows for transactional behavior
- Set the Flow Debug Port for an integration server
- Enable and disable a user trace
- Enable and disable a service trace
- Enable and disable Trace nodes in a message flow
- Diagnose common problems in the IBM Integration Bus environment

Figure 9-44. Unit summary WM6461.0

Notes:



WebSphere Education IBM

Checkpoint questions

1. True or False: Database updates are rolled back in an error situation if you configured the DBMS and queue manager for global transactions.

2. How can messages get lost in an integration node?

3. How can you prevent messages from being written to the queue manager's dead-letter queue?

© Copyright IBM Corporation 2015

Figure 9-45. Checkpoint questions WM6461.0

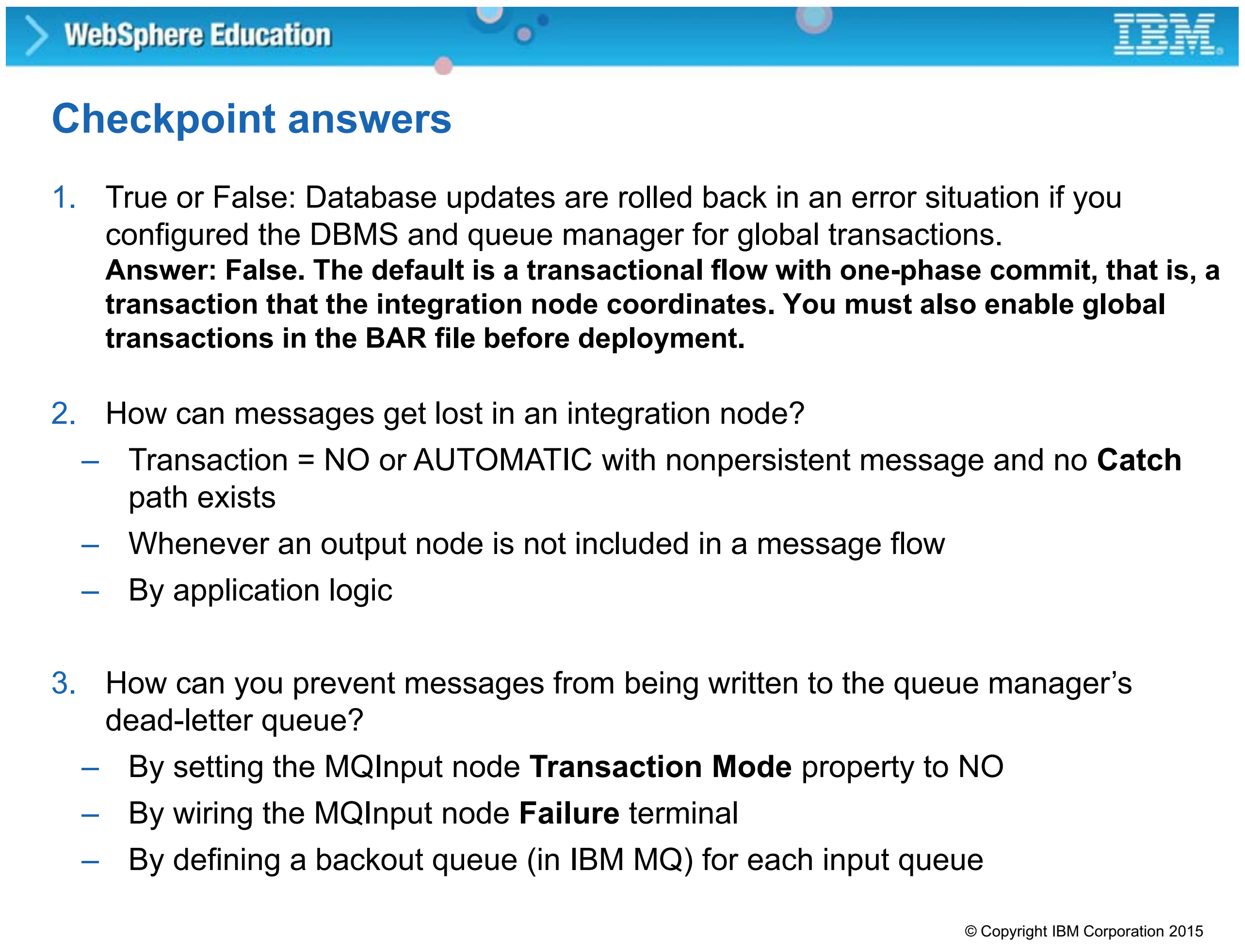
Notes:

Write your answers here:

1.

2.

3.



WebSphere Education

Checkpoint answers

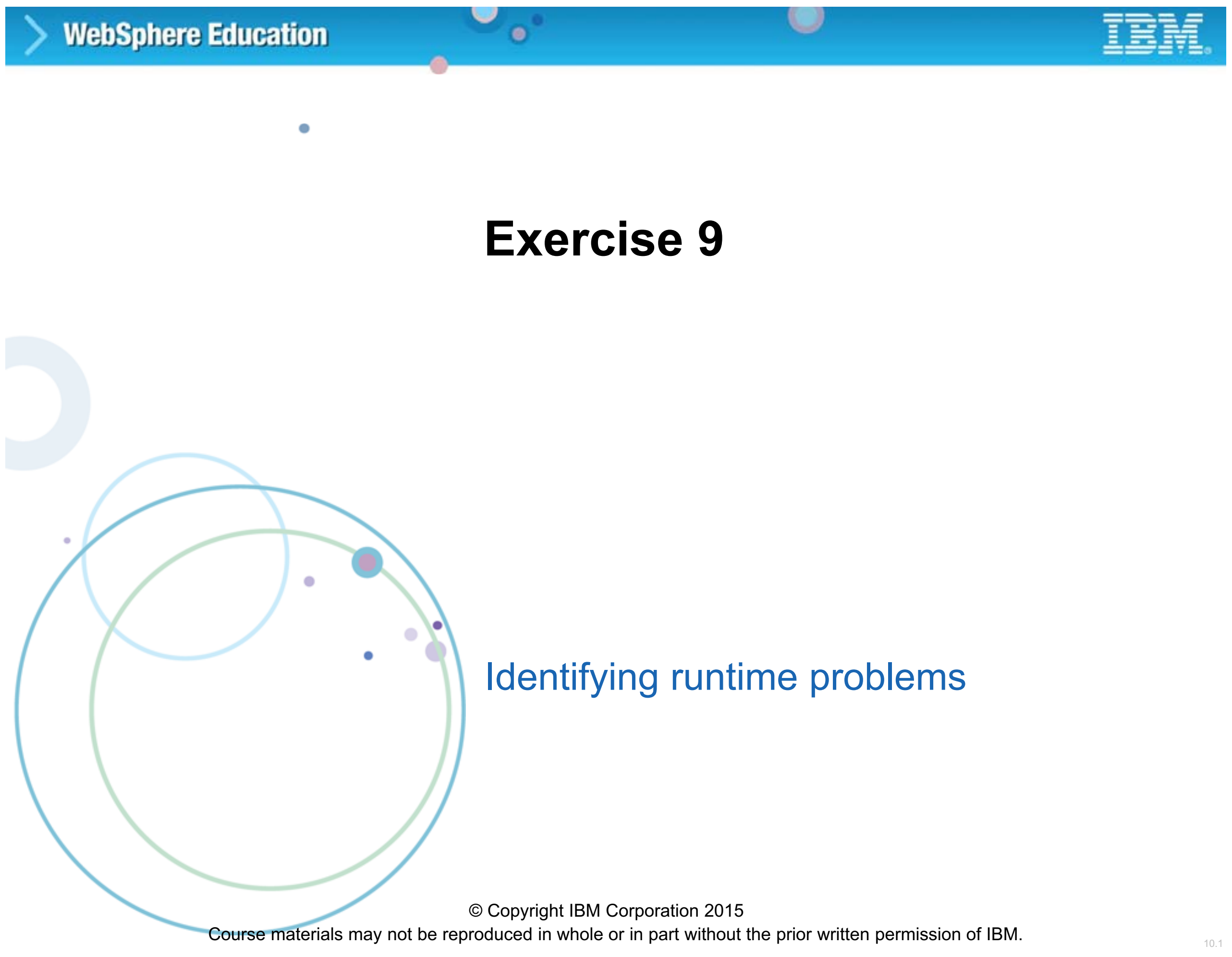
1. True or False: Database updates are rolled back in an error situation if you configured the DBMS and queue manager for global transactions.
   Answer: False. The default is a transactional flow with one-phase commit, that is, a transaction that the integration node coordinates. You must also enable global transactions in the BAR file before deployment.

2. How can messages get lost in an integration node?
   – Transaction = NO or AUTOMATIC with nonpersistent message and no **Catch** path exists
   – Whenever an output node is not included in a message flow
   – By application logic

3. How can you prevent messages from being written to the queue manager's dead-letter queue?
   – By setting the MQInput node **Transaction Mode** property to NO
   – By wiring the MQInput node **Failure** terminal
   – By defining a backout queue (in IBM MQ) for each input queue

Figure 9-46. Checkpoint answers WM6461.0

Notes:



WebSphere Education

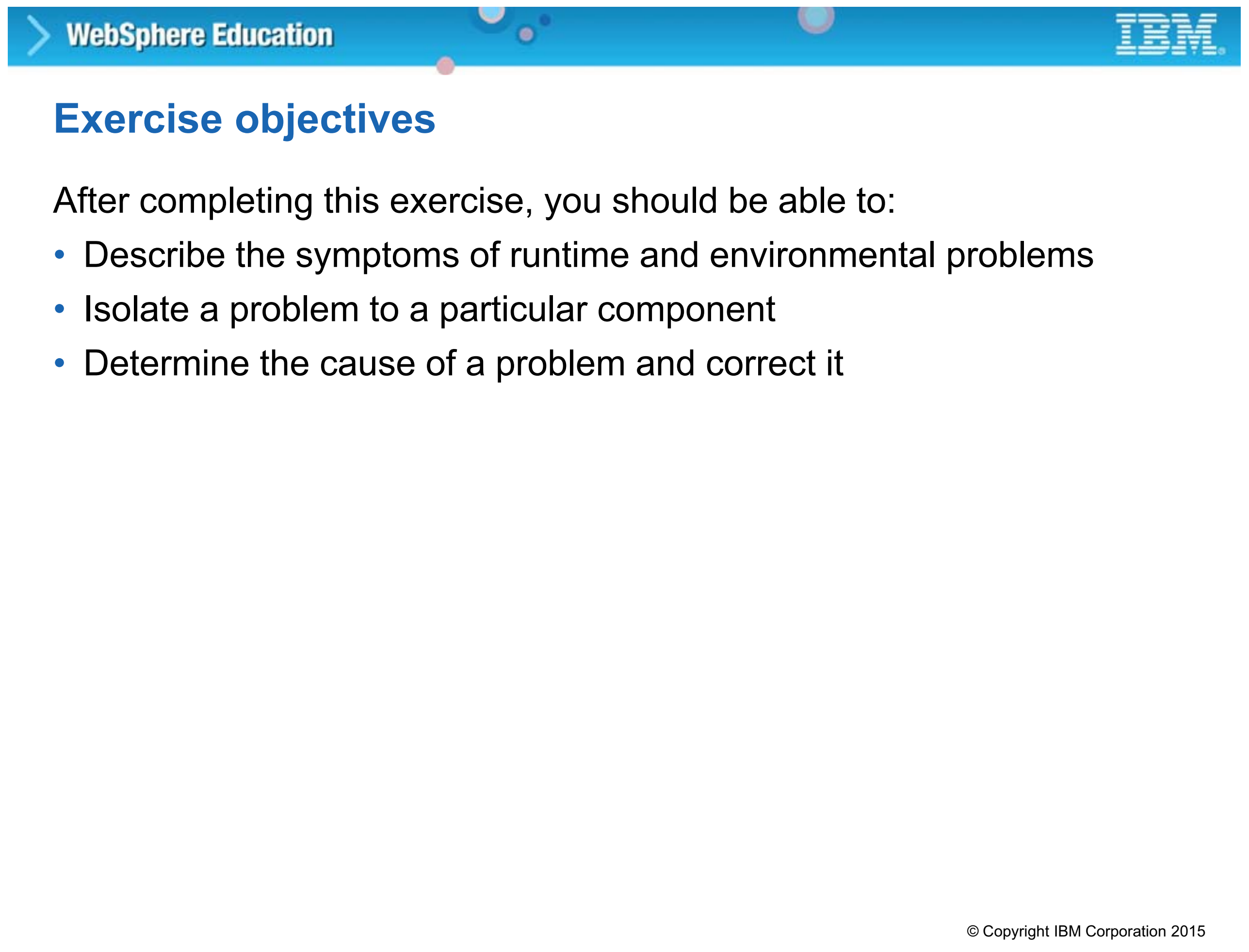
Exercise 9

Identifying runtime problems

Figure 9-47. Exercise 9 WM6461.0

Notes:

In the first part of this exercise, you analyze a series of message flows that produce runtime errors. You determine where the incoming message is found after the errors occur. In the second part of this exercise, you analyze and correct error situations that involve an integration node that connects to an IBM MQ queue manager.



Exercise objectives

After completing this exercise, you should be able to:

- Describe the symptoms of runtime and environmental problems
- Isolate a problem to a particular component
- Determine the cause of a problem and correct it

© Copyright IBM Corporation 2015

Figure 9-48. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 10. Monitoring the integration node and message flow performance

What this unit is about

In this unit, you learn how to collect performance statistics for integration node resources and message flows. You also learn the principles of tuning the integration node environment for optimal performance.

What you should be able to do

After completing this unit, you should be able to:

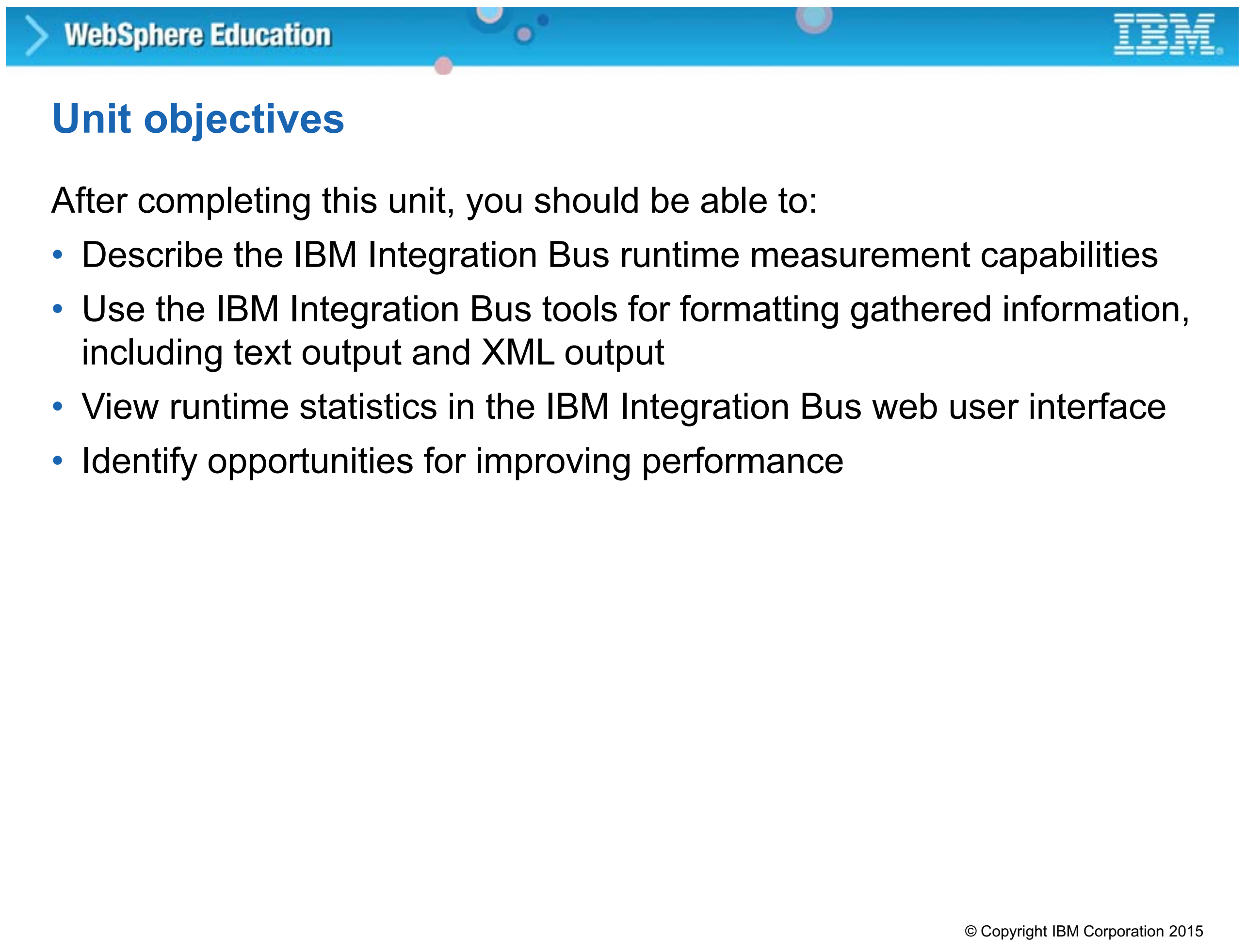
- Describe the IBM Integration Bus runtime measurement capabilities
- Use the IBM Integration Bus tools for formatting gathered information, including text output and XML output
- View runtime statistics in the IBM Integration Bus web user interface
- Identify opportunities for improving performance

How you will check your progress

- Checkpoint questions

References

IBM Knowledge Center for IBM Integration Bus V10


WebSphere Education
IBM

Unit objectives

After completing this unit, you should be able to:

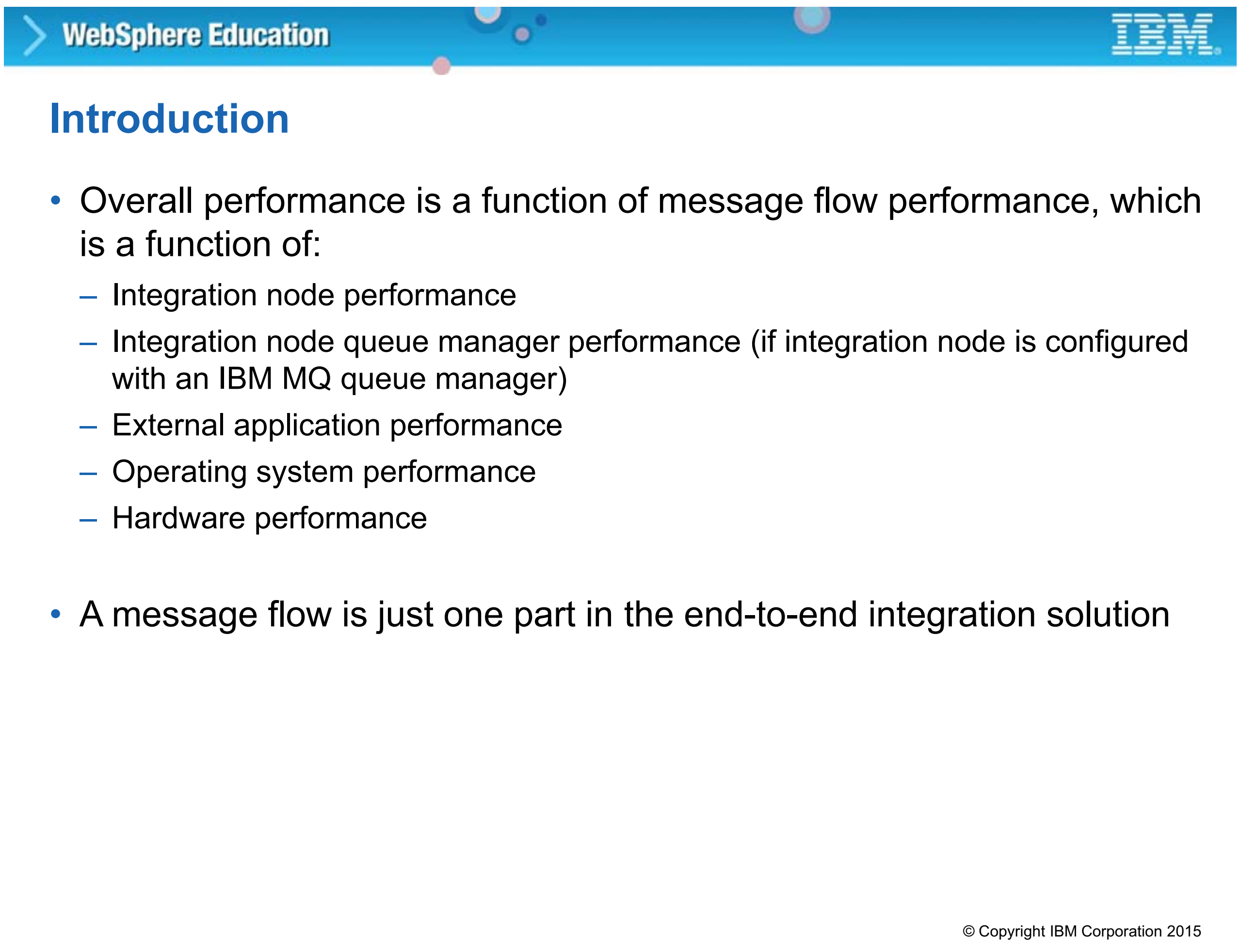
- Describe the IBM Integration Bus runtime measurement capabilities
- Use the IBM Integration Bus tools for formatting gathered information, including text output and XML output
- View runtime statistics in the IBM Integration Bus web user interface
- Identify opportunities for improving performance

© Copyright IBM Corporation 2015

Figure 10-1. Unit objectives WM6461.0

Notes:

10.1.Performance monitoring overview



WebSphere Education

Introduction

- Overall performance is a function of message flow performance, which is a function of:
  - Integration node performance
  - Integration node queue manager performance (if integration node is configured with an IBM MQ queue manager)
  - External application performance
  - Operating system performance
  - Hardware performance
- A message flow is just one part in the end-to-end integration solution

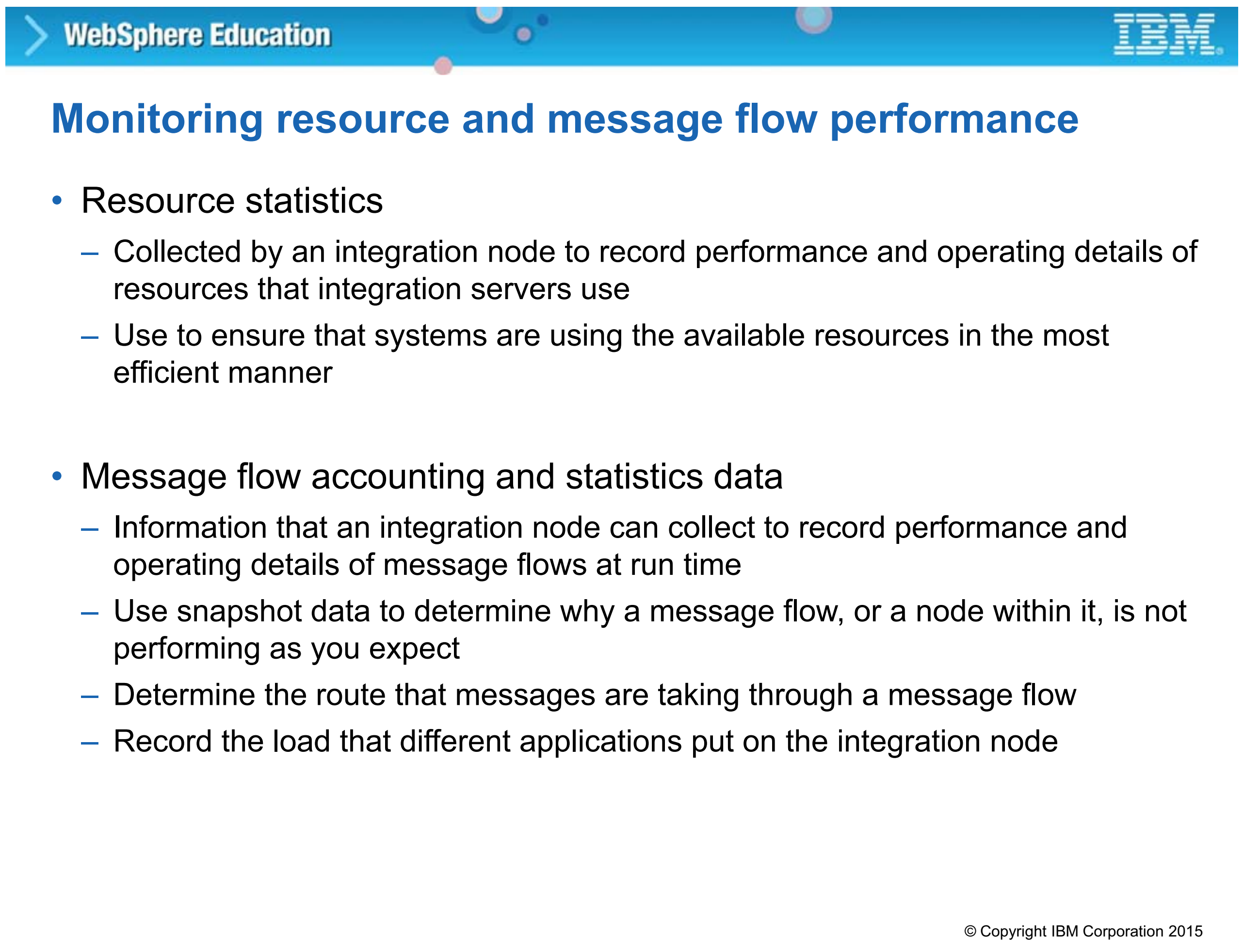
© Copyright IBM Corporation 2015

Figure 10-2. Introduction WM6461.0

Notes:

The message rate that it is possible to achieve when running any specific message flow is a function of the performance characteristics of many different but related items:

- How the message flow is written
- How the integration node is configured
- For databases, the frequency of use by message flows and the database operation (INSERT, UPDATE, SELECT, or DELTE)
- For the integration node queue manager, the efficiency of MQGET and MQPUT calls from message flows
- The selected operating system
- For hardware, the speed and number of processors, memory, number, speed, and frequency of access to disk storage



WebSphere Education

Monitoring resource and message flow performance

- Resource statistics
  - Collected by an integration node to record performance and operating details of resources that integration servers use
  - Use to ensure that systems are using the available resources in the most efficient manner
- Message flow accounting and statistics data
  - Information that an integration node can collect to record performance and operating details of message flows at run time
  - Use snapshot data to determine why a message flow, or a node within it, is not performing as you expect
  - Determine the route that messages are taking through a message flow
  - Record the load that different applications put on the integration node

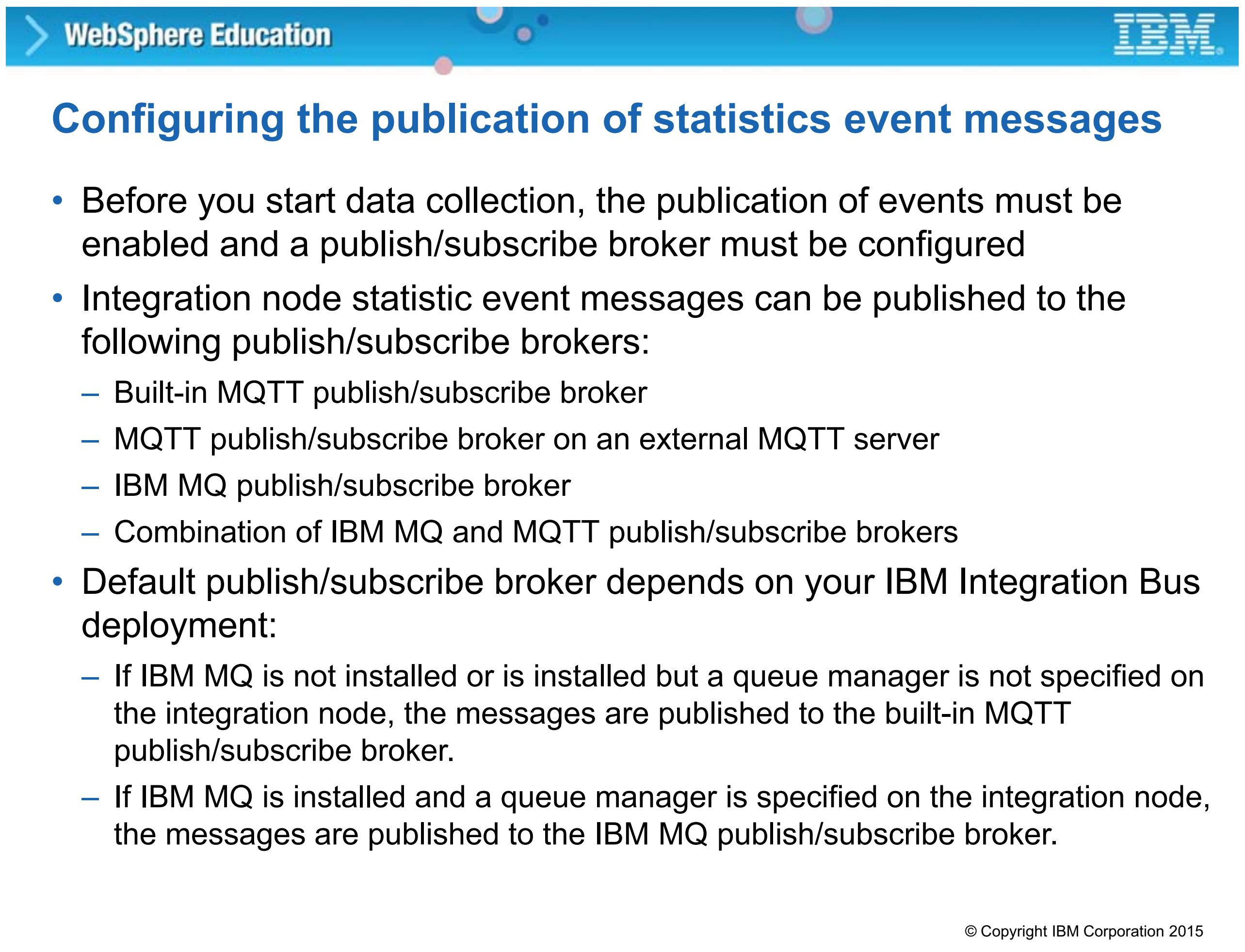
© Copyright IBM Corporation 2015

Figure 10-3. Monitoring resource and message flow performance WM6461.0

Notes:

Using the Integration Bus accounting and statistics features, administrators can dynamically measure various runtime attributes of integration node resources. For example, the total number of messages that are processed within a time interval, itemizing the associated processor cost, average message size, and even the total number of bytes processed can be accumulated.

The Integration Bus accounting and statistics features can also be used to understand how messages are being processed. For example, they can demonstrate that a particular error path is being taken with greater than expected frequency. This finding might lead to system or message flow modifications to make the message processing more effective.



WebSphere Education

Configuring the publication of statistics event messages

- Before you start data collection, the publication of events must be enabled and a publish/subscribe broker must be configured
- Integration node statistic event messages can be published to the following publish/subscribe brokers:
  - Built-in MQTT publish/subscribe broker
  - MQTT publish/subscribe broker on an external MQTT server
  - IBM MQ publish/subscribe broker
  - Combination of IBM MQ and MQTT publish/subscribe brokers
- Default publish/subscribe broker depends on your IBM Integration Bus deployment:
  - If IBM MQ is not installed or is installed but a queue manager is not specified on the integration node, the messages are published to the built-in MQTT publish/subscribe broker.
  - If IBM MQ is installed and a queue manager is specified on the integration node, the messages are published to the IBM MQ publish/subscribe broker.

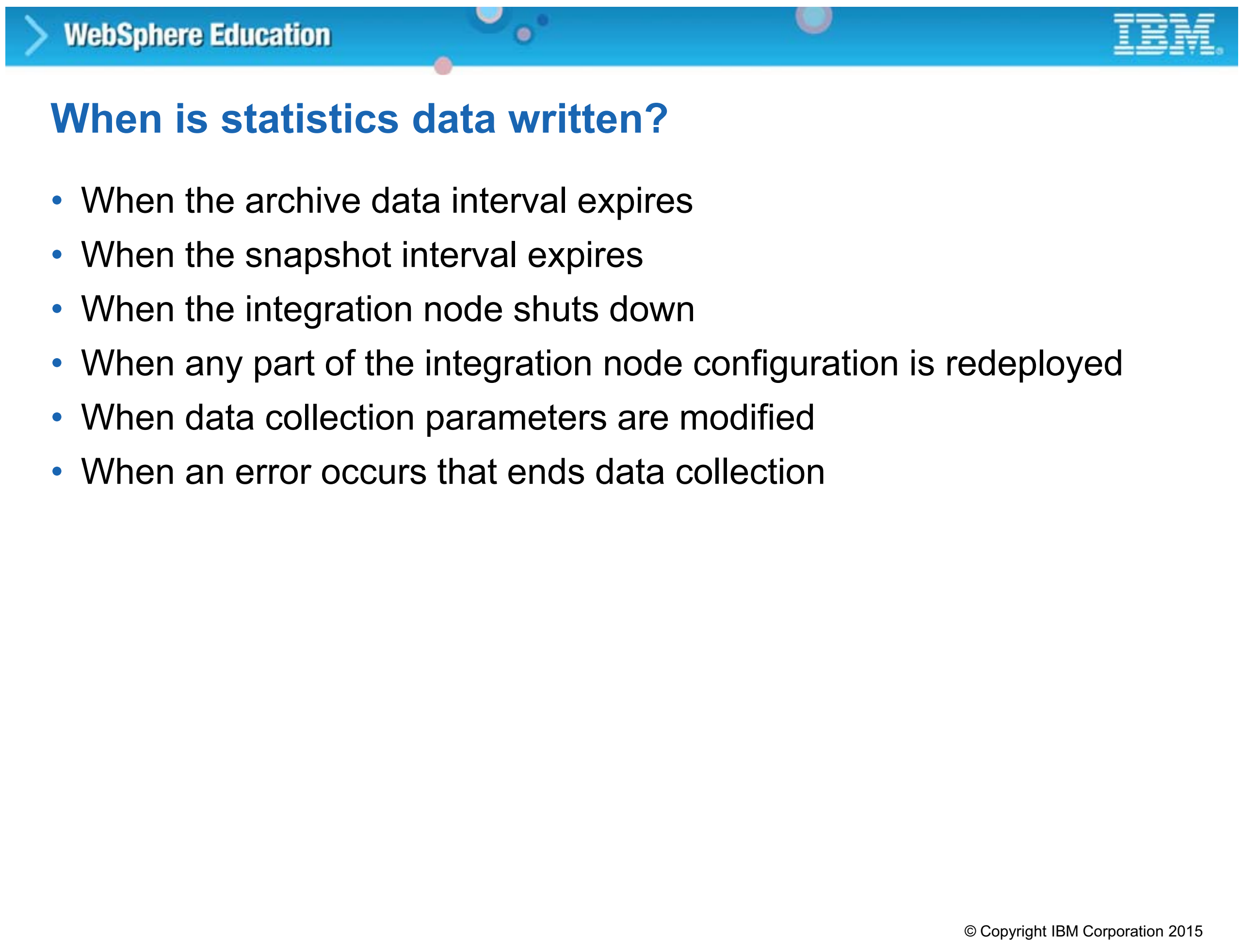
Figure 10-4. Configuring the publication of statistics event messages WM6461.0

Notes:

Integration node data collection and event messages can be published to the following publish/subscribe brokers:

- The built-in MQTT publish/subscribe broker that is supplied with Integration Bus
- An MQTT publish/subscribe broker on an external MQTT server
- An IBM MQ publish/subscribe broker
- A combination of IBM MQ and MQTT publish/subscribe brokers

The default publish/subscribe broker that is used for each group of event messages depends on your IBM Integration Bus deployment.



When is statistics data written?

- When the archive data interval expires
- When the snapshot interval expires
- When the integration node shuts down
- When any part of the integration node configuration is redeployed
- When data collection parameters are modified
- When an error occurs that ends data collection

© Copyright IBM Corporation 2015

Figure 10-5. When is statistics data written? WM6461.0

Notes:

The figure lists the circumstances under which statistics data is written to the specified output location.

There are two time periods over which statistics are available: snapshot (seconds) and archive (minutes to hours). Snapshot is usually used when there is a specific performance problem to analyze; archive statistics are used for charge-back and long-term performance monitoring.

During integration node shutdown, any data that the integration node collected but did not write to the specified output destination, is flushed to the output. Therefore, it might represent data for an incomplete interval.

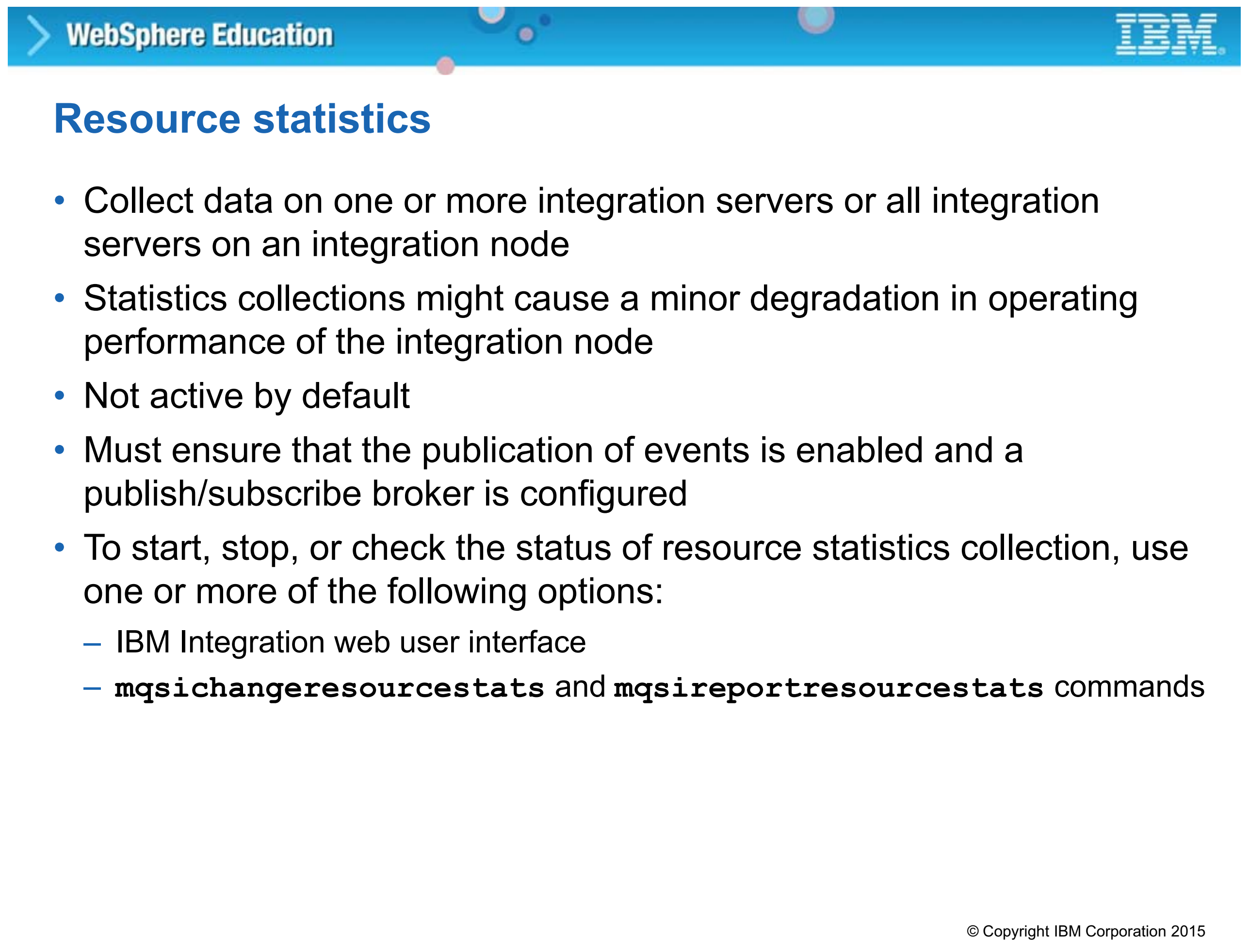
Redeployed configuration data might contain an updated configuration that is not consistent with the existing record structure (for example, a message flow might include another node). Therefore, the current data, which might represent an incomplete interval, is written to the output destination.

If you update the data collection parameters, all data that is collected for the message flow (or message flows) is written to the output destination to retain data integrity. Statistics collection is restarted according to the new parameters.

Data collection continues for the redeployed configuration until the data collection parameters are changed or data collection is stopped.

You must restart data collection manually when an error occurs that ends data collection.

10.2.Resource statistics



WebSphere Education

Resource statistics

- Collect data on one or more integration servers or all integration servers on an integration node
- Statistics collections might cause a minor degradation in operating performance of the integration node
- Not active by default
- Must ensure that the publication of events is enabled and a publish/subscribe broker is configured
- To start, stop, or check the status of resource statistics collection, use one or more of the following options:
  - IBM Integration web user interface
  - **`mqsichangeresourcestats`** and **`mqsireportresourcestats`** commands

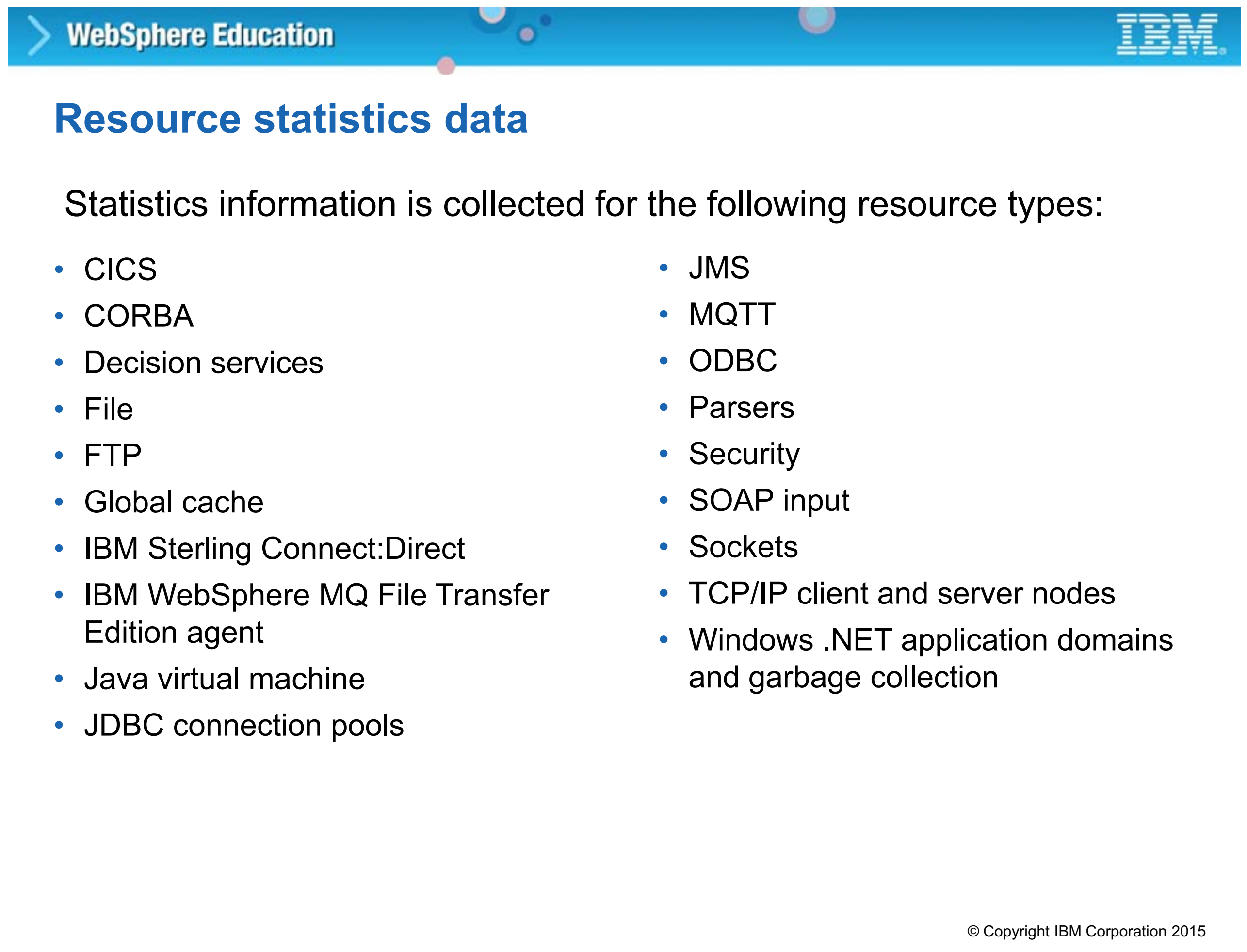
© Copyright IBM Corporation 2015

Figure 10-6. Resource statistics WM6461.0

Notes:

Integration Bus resource statistics can be used to ensure that systems are using the available resources in the most efficient manner. If you detect that system resources are under pressure, you can examine the statistics that the integration node collects to assess whether there is excessive resource consumption.

Resource statistics collection can be managed from the Integration web interface and the command interface.



WebSphere Education

Resource statistics data

Statistics information is collected for the following resource types:

- CICS
- CORBA
- Decision services
- File
- FTP
- Global cache
- IBM Sterling Connect:Direct
- IBM WebSphere MQ File Transfer Edition agent
- Java virtual machine
- JDBC connection pools
- JMS
- MQTT
- ODBC
- Parsers
- Security
- SOAP input
- Sockets
- TCP/IP client and server nodes
- Windows .NET application domains and garbage collection

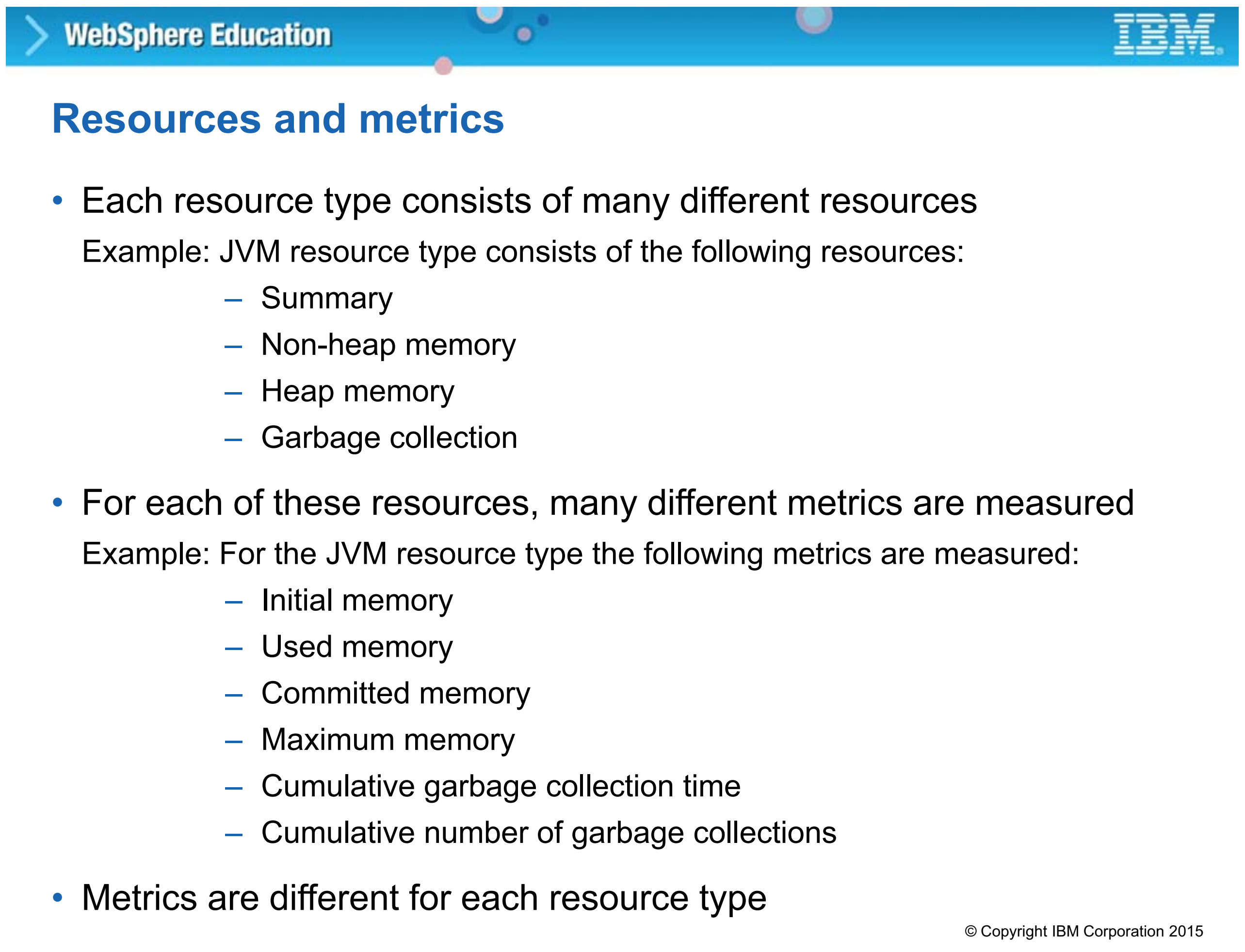
Figure 10-7. Resource statistics data WM6461.0

Notes:

The figure lists the resource types for which you can collect statistics.

You can view these statistics in the web user interface, or you can write a program that subscribes to a publication (single XML message) that returns this data.

For a detailed description of the statistics for each resource type, see the IBM Knowledge Center for IBM Integration Bus V10.



WebSphere Education IBM

Resources and metrics

- Each resource type consists of many different resources

  Example: JVM resource type consists of the following resources:
  - Summary
  - Non-heap memory
  - Heap memory
  - Garbage collection
- For each of these resources, many different metrics are measured

  Example: For the JVM resource type the following metrics are measured:
  - Initial memory
  - Used memory
  - Committed memory
  - Maximum memory
  - Cumulative garbage collection time
  - Cumulative number of garbage collections
- Metrics are different for each resource type

Figure 10-8. Resources and metrics WM6461.0

Notes:

Each resource type consists of different resources.

For example, each integration server starts its own JVM, which supports all Java activities in that integration server. The following statistics are collected for JVM resources:

- Heap memory
- Non-heap memory
- Garbage collection

Each resource has different metrics and measurements. For example, the JVM resource type provides the following measurements:

- Initial memory: The initial amount of memory that the JVM requests from the operating system for memory management during start. Its value might be undefined.
- Used memory: The amount of memory that is in use.
- Committed memory: The amount of memory that is allocated to the JVM by the operating system.

- Maximum memory: The maximum amount of memory that can be used for memory management. Its value might be undefined.
- Cumulative garbage collection time: The accumulated garbage collection elapsed time in seconds for this instance of the JVM.
- Cumulative number of garbage collections: The total number of garbage collections that occurred for this instance of the JVM.

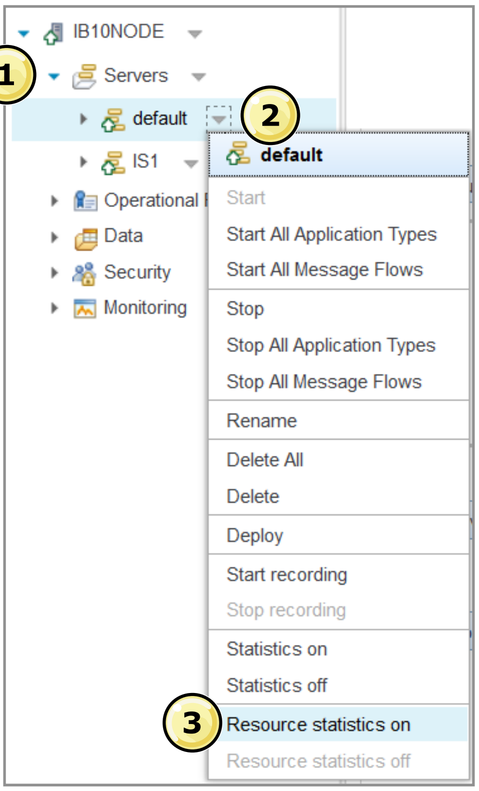
WebSphere Education IBM

Controlling resource collection in the Integration web interface

- To start resource statistics collection in the Integration web interface:
  1. Expand **Servers**
  2. Click the down arrow next to the integration server to display the menu
  3. In the menu, click **Resource statistics on**
- A success message indicates that resource statistics collection is active

  ✓ Resource statistics successfully turned on
- To stop resource statistics collection, click **Resource statistics off** in the menu

  ✓ Resource statistics successfully turned off



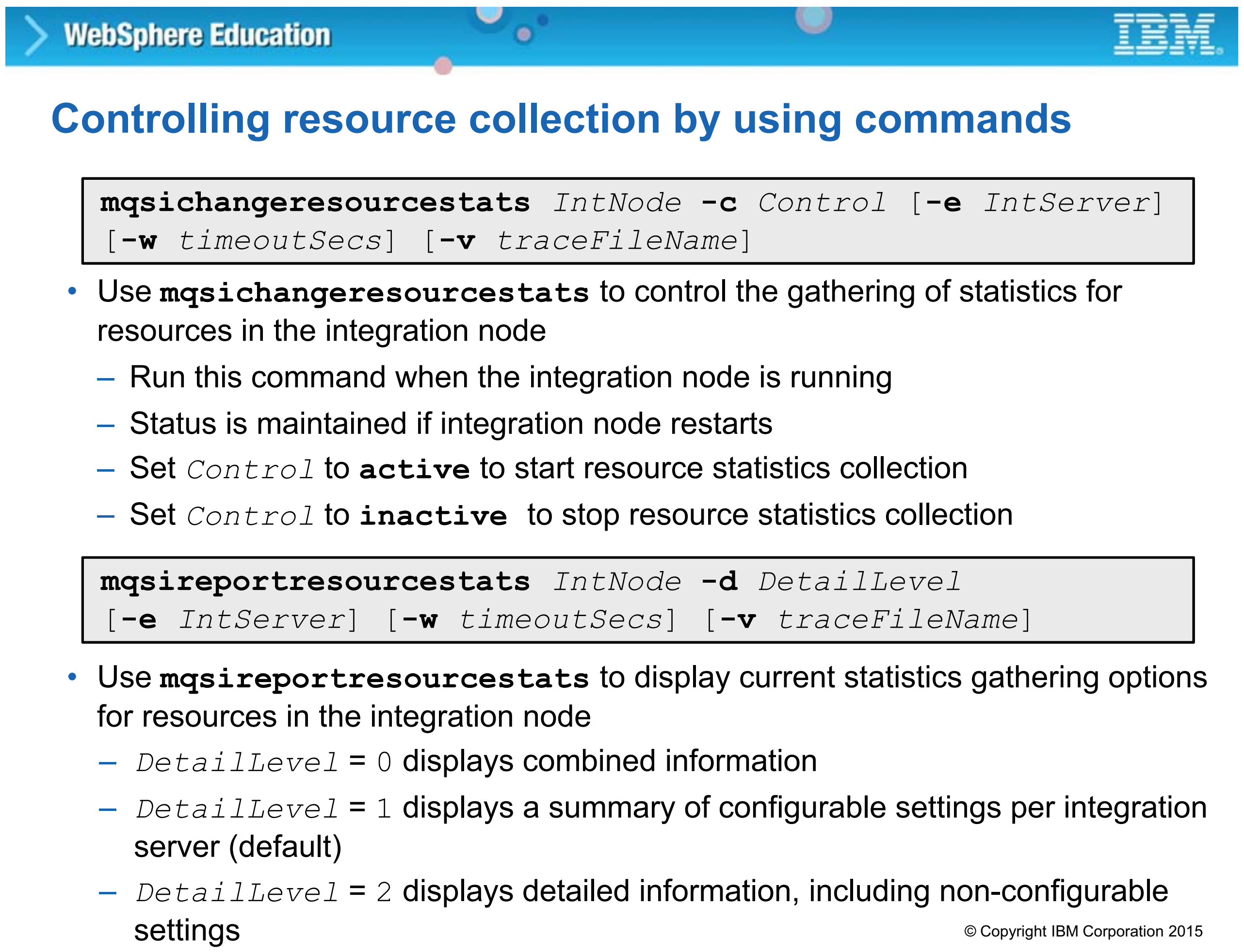
© Copyright IBM Corporation 2015

Figure 10-9. Controlling resource collection in the Integration web interface WM6461.0

Notes:

Before you can view resource statistics, you can start resource statistics collection in the Integration web user interface, as described in the figure.

When you click **Resource statistics on**, a message is sent to the integration node to start collecting data for the selected integration server. A message is displayed in the integration server properties window to notify you that resource statistics is turned on.



WebSphere Education

Controlling resource collection by using commands

```
mqsichangeresourcestats IntNode -c Control [-e IntServer]
 [-w timeoutSecs] [-v traceFileName]
```

- Use **`mqsichangeresourcestats`** to control the gathering of statistics for resources in the integration node
  - Run this command when the integration node is running
  - Status is maintained if integration node restarts
  - Set *`Control`* to **`active`** to start resource statistics collection
  - Set *`Control`* to **`inactive`** to stop resource statistics collection

```
mqsireportresourcestats IntNode -d DetailLevel
 [-e IntServer] [-w timeoutSecs] [-v traceFileName]
```

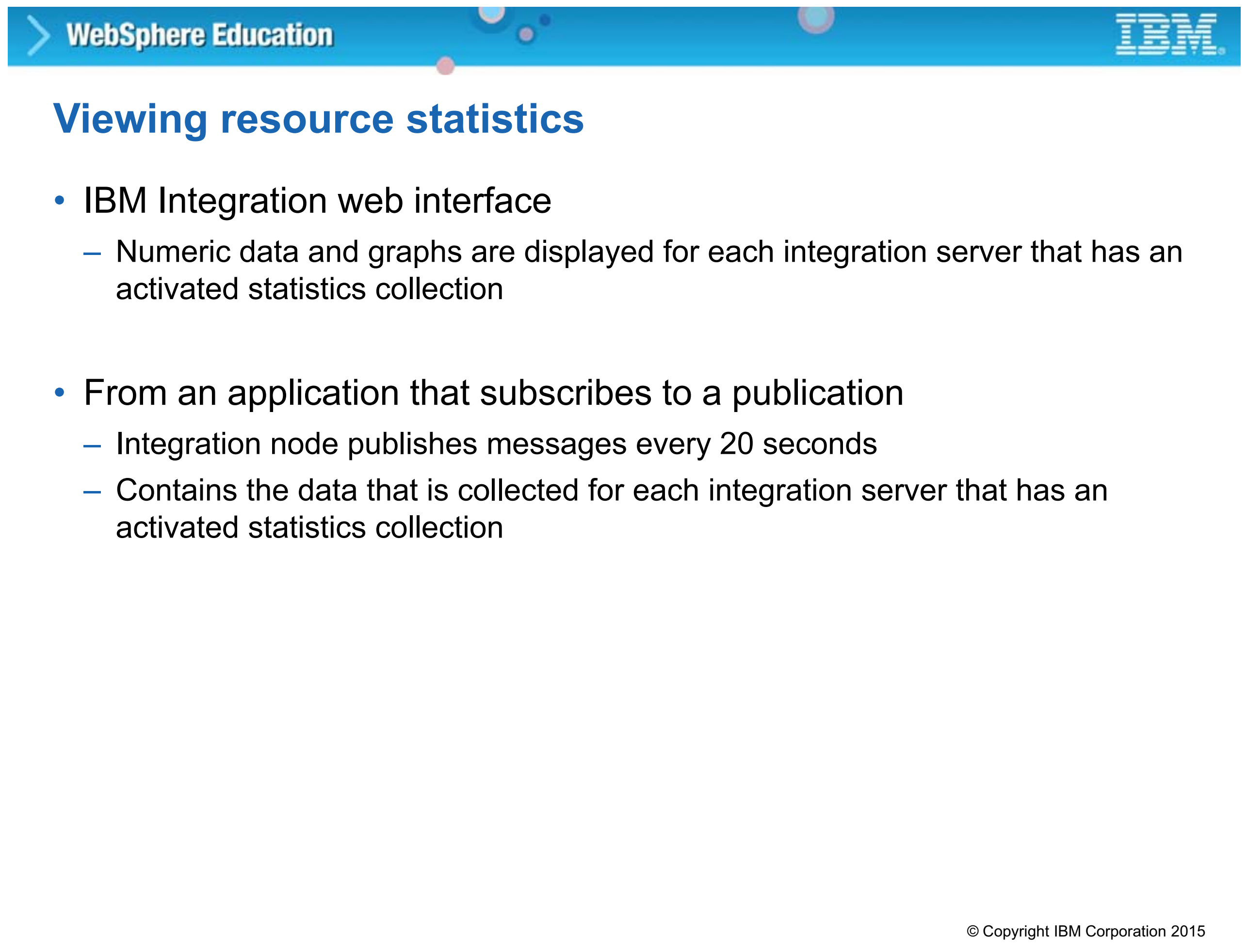
- Use **`mqsireportresourcestats`** to display current statistics gathering options for resources in the integration node
  - *`DetailLevel`* = `0` displays combined information
  - *`DetailLevel`* = `1` displays a summary of configurable settings per integration server (default)
  - *`DetailLevel`* = `2` displays detailed information, including non-configurable settings

© Copyright IBM Corporation 2015

Figure 10-10. Controlling resource collection by using commands WM6461.0

Notes:

This figure describes the commands and command syntax for enabling resource statistics and showing the status of resource status collection.



WebSphere Education

Viewing resource statistics

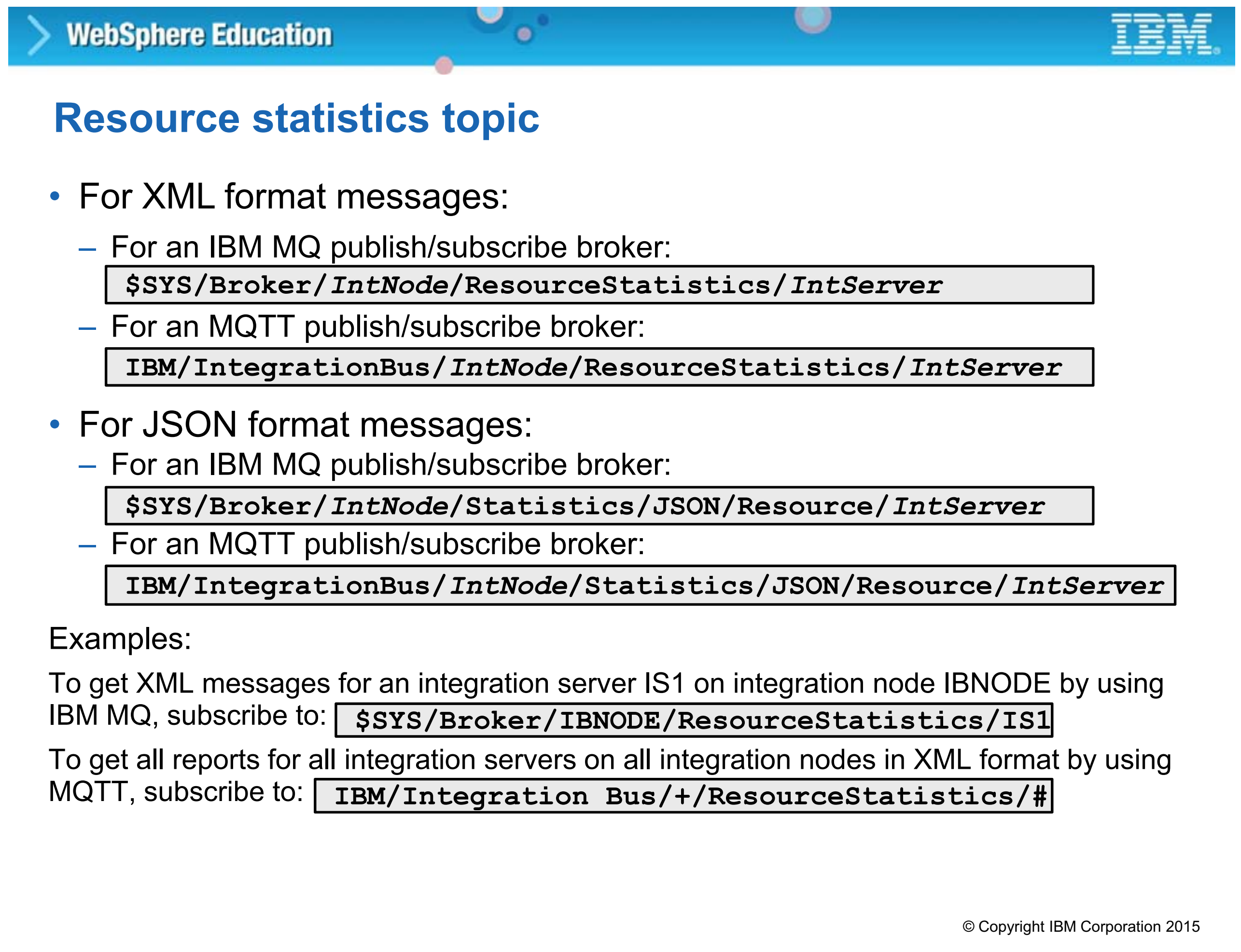
- IBM Integration web interface
  - Numeric data and graphs are displayed for each integration server that has an activated statistics collection
- From an application that subscribes to a publication
  - Integration node publishes messages every 20 seconds
  - Contains the data that is collected for each integration server that has an activated statistics collection

Figure 10-11. Viewing resource statistics WM6461.0

Notes:

You can examine the statistics that the integration node collects to assess whether the cause of the concern is the use of those resources by processes in Integration Bus.

The integration node publishes statistics every 20 seconds. You can view the statistics in Integration web interface or from an application that subscribes to that publication.



WebSphere Education

Resource statistics topic

- For XML format messages:
  - For an IBM MQ publish/subscribe broker:
    $SYS/Broker/*IntNode*/ResourceStatistics/*IntServer*
  - For an MQTT publish/subscribe broker:
    IBM/IntegrationBus/*IntNode*/ResourceStatistics/*IntServer*
- For JSON format messages:
  - For an IBM MQ publish/subscribe broker:
    $SYS/Broker/*IntNode*/Statistics/JSON/Resource/*IntServer*
  - For an MQTT publish/subscribe broker:
    IBM/IntegrationBus/*IntNode*/Statistics/JSON/Resource/*IntServer*

Examples:

To get XML messages for an integration server IS1 on integration node IBNODE by using IBM MQ, subscribe to: `$SYS/Broker/IBNODE/ResourceStatistics/IS1`

To get all reports for all integration servers on all integration nodes in XML format by using MQTT, subscribe to: `IBM/Integration Bus/+/ResourceStatistics/#`

Figure 10-12. Resource statistics topic WM6461.0

Notes:

The resource statistics topics are created in the folder `WMQIStatisticsAccounting`, which contains subfolders that provide more detailed information. All folders are present in the publication even if you set current data collection parameters to specify that the relevant data is not collected.

Snapshot and archive data is published to the following topics for JSON messages:

- For publications on an IBM MQ publish/subscribe broker:

  $SYS/Broker/*IntNode*/Statistics/JSON/SnapShot/
  $SYS/Broker/*IntNode*/Statistics/JSON/Archive/

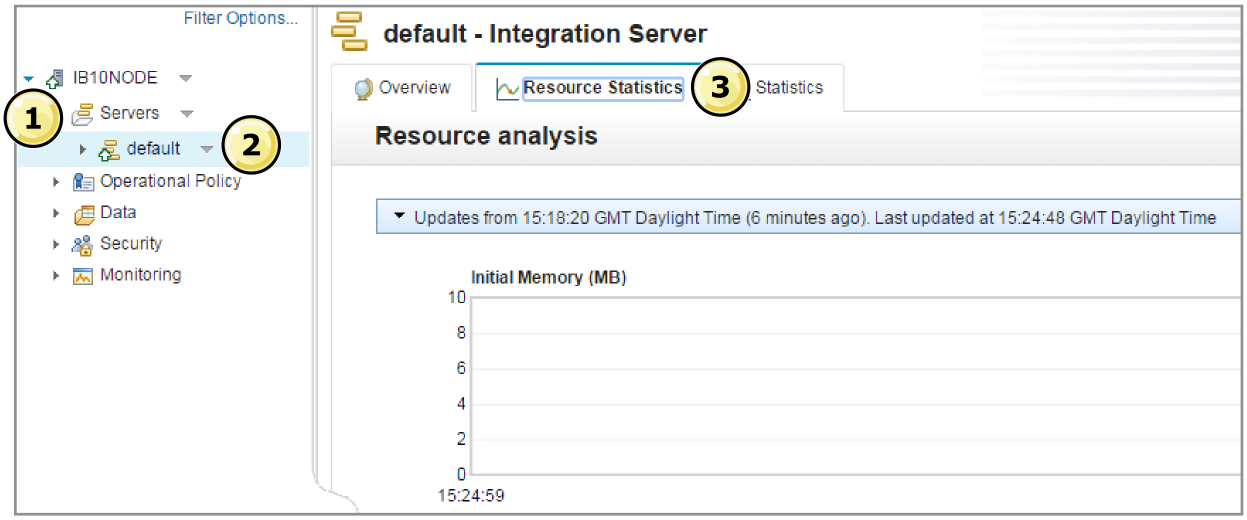
- For publications on an MQTT publish/subscribe broker:

  IBM/IntegrationBus/*IntNode*/Statistics/JSON/SnapShot/
  IBM/IntegrationBus/*IntNode*/Statistics/JSON/Archive/

You can use ESQL filters to further refine your subscriptions to examine the content of accounting and statistics messages.

View resource statistics in the Integration web interface

1. Expand the **Servers** section
2. Click the integration server for which you want to view resource statistics
3. Click the **Resource Statistics** tab



Updates from 15:18:20 GMT Daylight Time (6 minutes ago). Last updated at 15:24:48 GMT Daylight Time

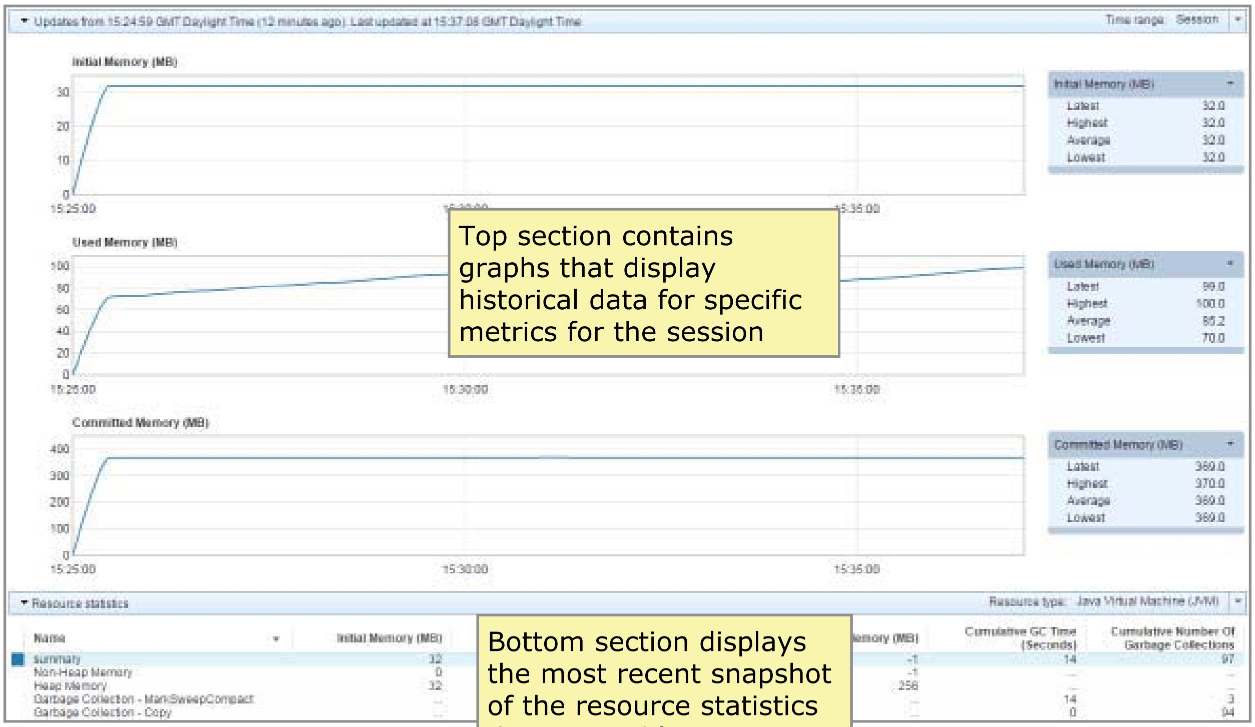
Figure 10-13. View resource statistics in the Integration web interface WM6461.0

Notes:

You can view resource statistics in the Integration web interface.

WebSphere Education IBM

Resource statistics view



Updates from 15:24:59 GMT Daylight Time (12 minutes ago). Last updated at 15:37:08 GMT Daylight Time

Time range: Session

Initial Memory (MB)	
Latest	32.0
Highest	32.0
Average	32.0
Lowest	32.0

Used Memory (MB)	
Latest	99.0
Highest	100.0
Average	85.2
Lowest	70.0

Committed Memory (MB)	
Latest	369.0
Highest	370.0
Average	369.0
Lowest	369.0

Resource statistics

Resource type: Java Virtual Machine (JVM)

Name	Initial Memory (MB)	emory (MB)	Cumulative GC Time (Seconds)	Cumulative Number Of Garbage Collections
summary	32	-1	14	97
Non-Heap Memory	0	-1	--	--
Heap Memory	32	256	--	--
Garbage Collection - MarkSweepCompact	--	--	14	3
Garbage Collection - Copy	--	--	0	94

Top section contains graphs that display historical data for specific metrics for the session

Bottom section displays the most recent snapshot of the resource statistics data in a table

Figure 10-14. Resource statistics view WM6461.0

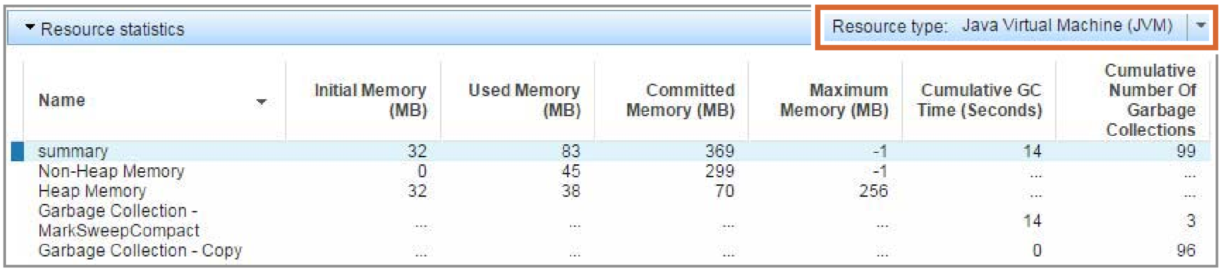
Notes:

The **Resource Statistics** tab shows three graphs and one table. By default the data is updated every 20 seconds. The data for the JVM is also displayed by default.

WebSphere Education

Change the resource type

- It is only possible to view the resource statistics data for a single resource type at any specific time
- By default, resource statistics for the **Java Virtual Machine (JVM)** are displayed when the **Resource Statistics** view is first displayed
- To change the resource type, select the resource type from the **Resource type** drop-down



Name	Initial Memory (MB)	Used Memory (MB)	Committed Memory (MB)	Maximum Memory (MB)	Cumulative GC Time (Seconds)	Cumulative Number Of Garbage Collections
summary	32	83	369	-1	14	99
Non-Heap Memory	0	45	299	-1	...	...
Heap Memory	32	38	70	256	...	...
Garbage Collection - MarkSweepCompact	...	...	...	...	14	3
Garbage Collection - Copy	...	...	...	...	0	96

Figure 10-15. Change the resource type WM6461.0

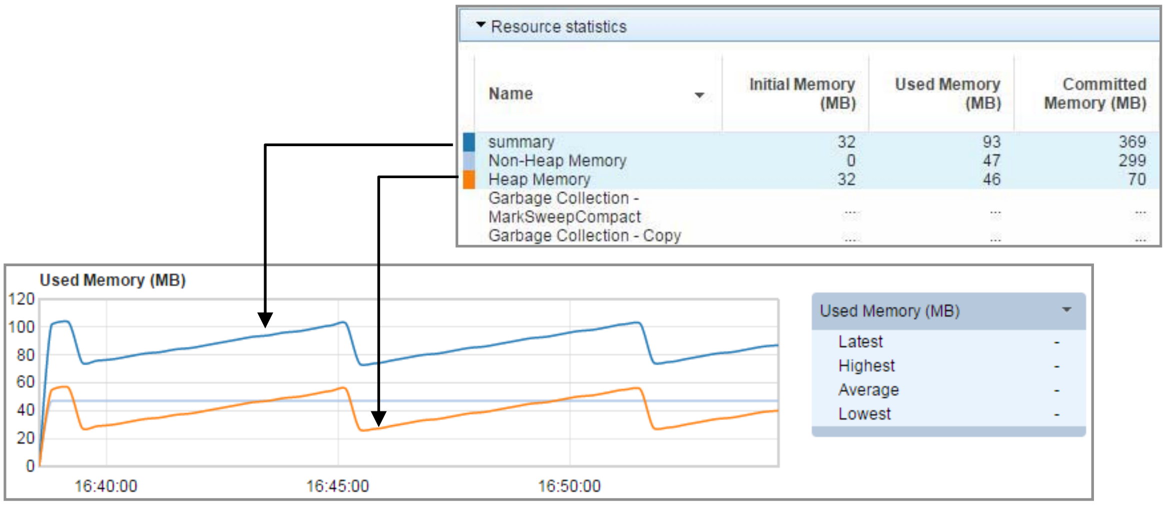
Notes:

The table consists of a row for each subcategory of the selected statistic and a number of columns, one for each measurement; some statistics have only a summary row. For example, if you select JVM, the table includes rows for heap and non-heap memory and columns for initial, used, committed, and maximum memory, plus a summary row.

If you want to view the statistics for a different resource, click the down arrow beside **Resource type** on the table to display the menu, and select the required resource. The statistics that are associated with this resource are then displayed in the table and graphs.

Display more resource graphs

- By default, the **summary** resource is selected in the table and a corresponding line is drawn on the graphs
- Some resource types display the **summary** resource only
- If multiple resources are available for the selected resource type, click the resource name in the table to add it to the graph
- Colors distinguish the resources on the graph



Resource statistics

Name	Initial Memory (MB)	Used Memory (MB)	Committed Memory (MB)
summary	32	93	369
Non-Heap Memory	0	47	299
Heap Memory	32	46	70
Garbage Collection - MarkSweepCompact	...	...	...
Garbage Collection - Copy	...	...	...

Figure 10-16. Display more resource graphs WM6461.0

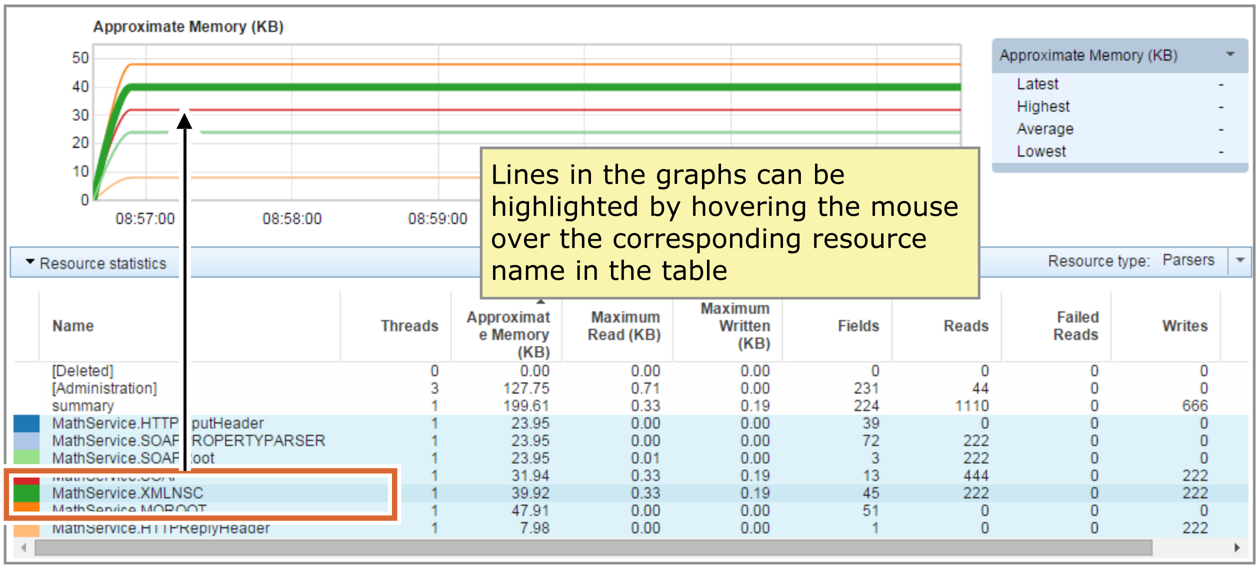
Notes:

By default, the three graphs display the data from the first three columns of the summary row. If the table contains more than one statistic, you can click a statistic name in the table to add it to the graph.

Different colors are used to distinguish the statistics on the graph.

WebSphere Education

Highlight resources in the graphs



Lines in the graphs can be highlighted by hovering the mouse over the corresponding resource name in the table

	Name	Threads	Approximate Memory (KB)	Maximum Read (KB)	Maximum Written (KB)	Fields	Reads	Failed Reads	Writes
	[Deleted]	0	0.00	0.00	0.00	0	0	0	0
	[Administration]	3	127.75	0.71	0.00	231	44	0	0
	summary	1	199.61	0.33	0.19	224	1110	0	666
🟦	MathService.HTTP putHeader	1	23.95	0.00	0.00	39	0	0	0
🟦	MathService.SOAP ROPERTYPARSER	1	23.95	0.00	0.00	72	222	0	0
🟩	MathService.SOAP oot	1	23.95	0.01	0.00	3	222	0	0
🟥	MathService.SOAP	1	31.94	0.33	0.19	13	444	0	222
🟩	MathService.XMLNSC	1	39.92	0.33	0.19	45	222	0	222
🟧	MathService.MQROOT	1	47.91	0.00	0.00	51	0	0	0
🟧	MathService.HTTPReplyHeader	1	7.98	0.00	0.00	1	0	0	222

Figure 10-17. Highlight resources in the graphs WM6461.0

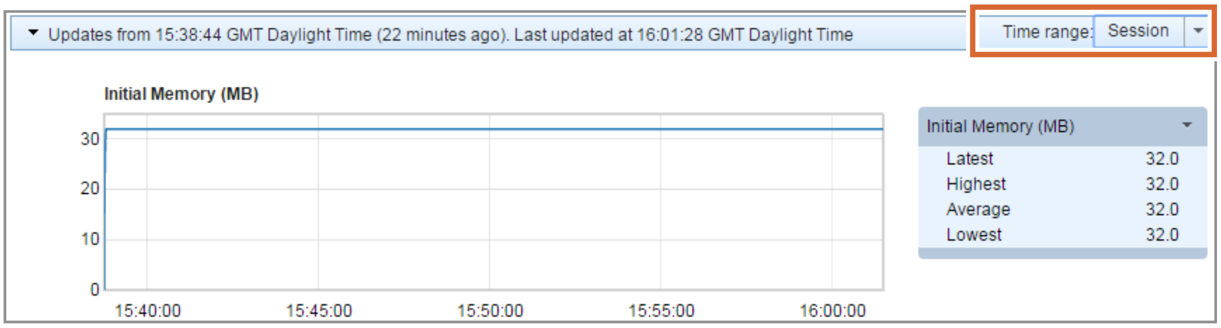
Notes:

If you hover over a statistic name in the table, the associated line in the graph is highlighted, as shown in the example in the figure.

WebSphere Education

Change the displayed time range

- By default, resource statistics for the entire session are displayed in the graphs
- To change the time range, select the required range by using the **Time range** menu
- Graphs redraw so that only data for the selected time range is displayed
- Available time ranges are:
  - Session
  - 5 minutes
  - 15 minutes
  - 30 minutes
  - 60 minutes



Updates from 15:38:44 GMT Daylight Time (22 minutes ago). Last updated at 16:01:28 GMT Daylight Time

Time range: Session

Initial Memory (MB)

Initial Memory (MB)	
Latest	32.0
Highest	32.0
Average	32.0
Lowest	32.0

Figure 10-18. Change the displayed time range WM6461.0

Notes:

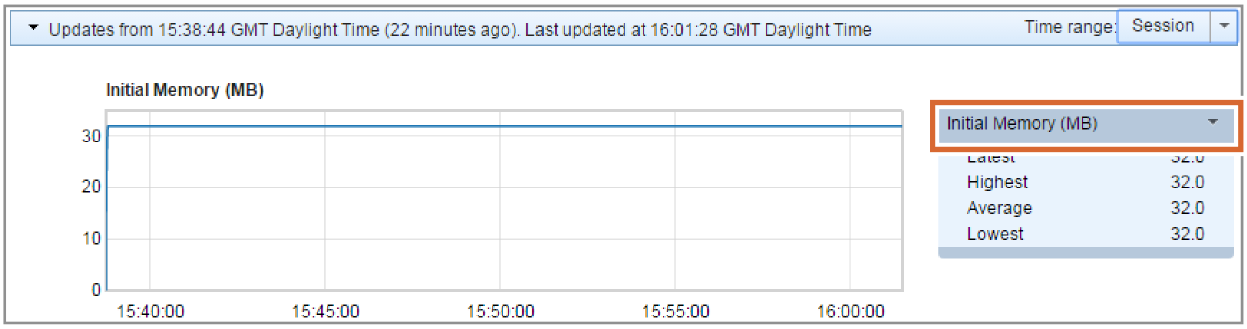
You can change the time period that is covered by the graph by clicking the arrow next to **Time range** in the graph header and selecting the time period that you want to view.

For example, to see the resource statistics that were collected in the last hour, select **60 mins**. To see all the resource statistics that were collected since resource statistic collection started, select **Session**.

WebSphere Education IBM

Change the metric that is displayed by a graph

- Each graph displays historical data for a single metric
- By default, the first three metrics for the current resource type are displayed in the graphs
- To change the metric, select the required metric in the table next to the graph
  – The graph redraws to display the data for the selected metric


Updates from 15:38:44 GMT Daylight Time (22 minutes ago). Last updated at 16:01:28 GMT Daylight Time

Time range: Session

Initial Memory (MB)

Initial Memory (MB)	
Latest	32.0
Highest	32.0
Average	32.0
Lowest	32.0

Figure 10-19. Change the metric that is displayed by a graph WM6461.0

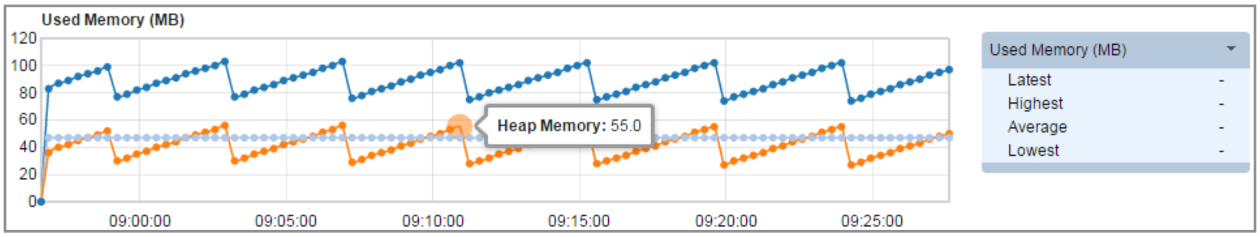
Notes:

You can display the data from different columns of the table by clicking the arrow beside a graph and selecting a column name from the menu, as shown in the example in the figure.

WebSphere Education IBM

Display the data points in a graph

- The data points that are collected for each resource can be displayed on a graph by hovering the mouse over the graph
- Hovering the mouse over a data point displays the actual measurement that was collected then



Used Memory (MB)	
Latest	-
Highest	-
Average	-
Lowest	-

Figure 10-20. Display the data points in a graph WM6461.0

Notes:

If you hover over a graph, you can see the collection points. If you hover over a collection point, you can see the value of the measurement that was collected then, as shown in the following image.

WebSphere Education IBM

Download resource statistics

- Download the resource statistics data for the selected resource type by clicking **Download data**
  - Browser prompts the user to save the data as a **CSV** file
- Name of the file has the following format:
  `ResourceStatistics_`*`ResourceType`*`_`*`Timestamp`*`.csv`

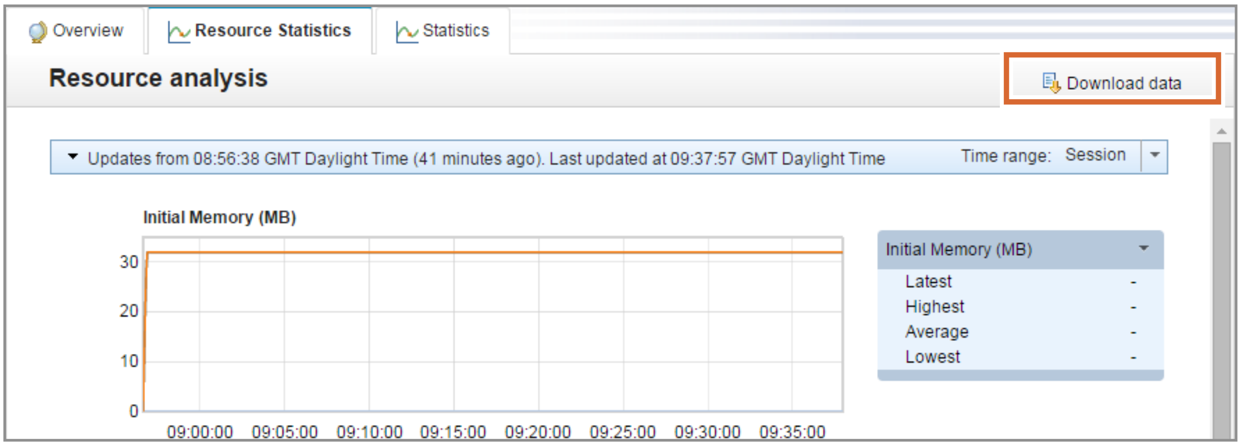


Figure 10-21. Download resource statistics WM6461.0

Notes:

You can download resource statistics by clicking **Download data** on the **Resource Statistics** tab.

WebSphere Education

Example: Downloaded resource statistics

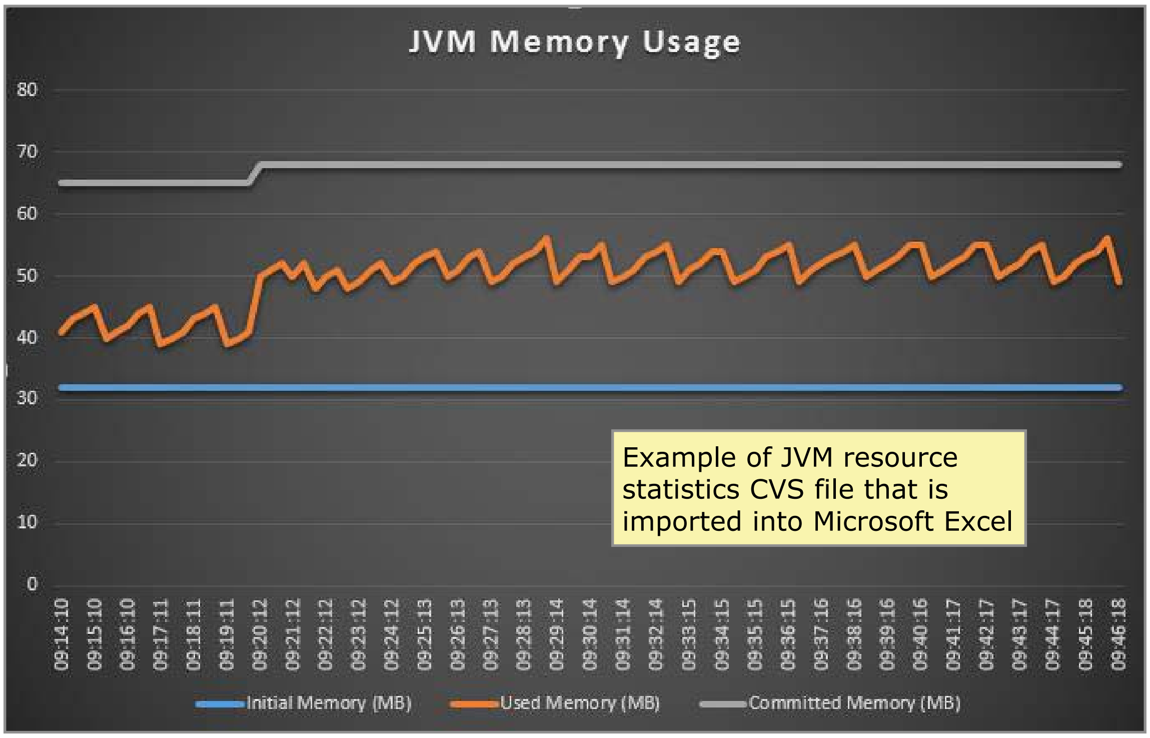
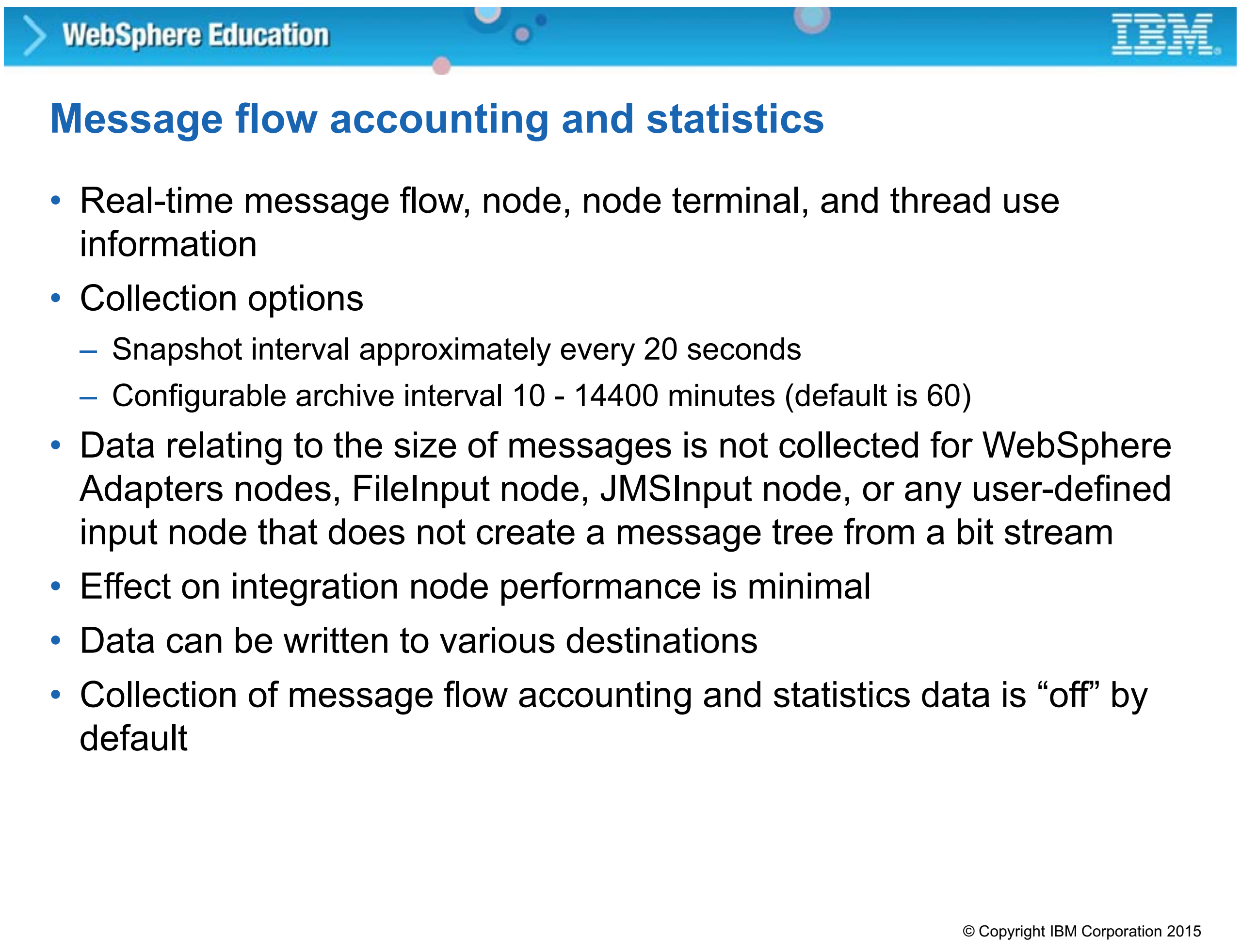


Figure 10-22. Example: Downloaded resource statistics WM6461.0

Notes:

The CSV format of the resource statistics download file allows the data to be used easily within tools such as Microsoft Excel.

10.3.Message flow accounting and statistics data



WebSphere Education IBM.

Message flow accounting and statistics

- Real-time message flow, node, node terminal, and thread use information
- Collection options
  - Snapshot interval approximately every 20 seconds
  - Configurable archive interval 10 - 14400 minutes (default is 60)
- Data relating to the size of messages is not collected for WebSphere Adapters nodes, FileInput node, JMSInput node, or any user-defined input node that does not create a message tree from a bit stream
- Effect on integration node performance is minimal
- Data can be written to various destinations
- Collection of message flow accounting and statistics data is “off” by default

© Copyright IBM Corporation 2015

Figure 10-23. Message flow accounting and statistics WM6461.0

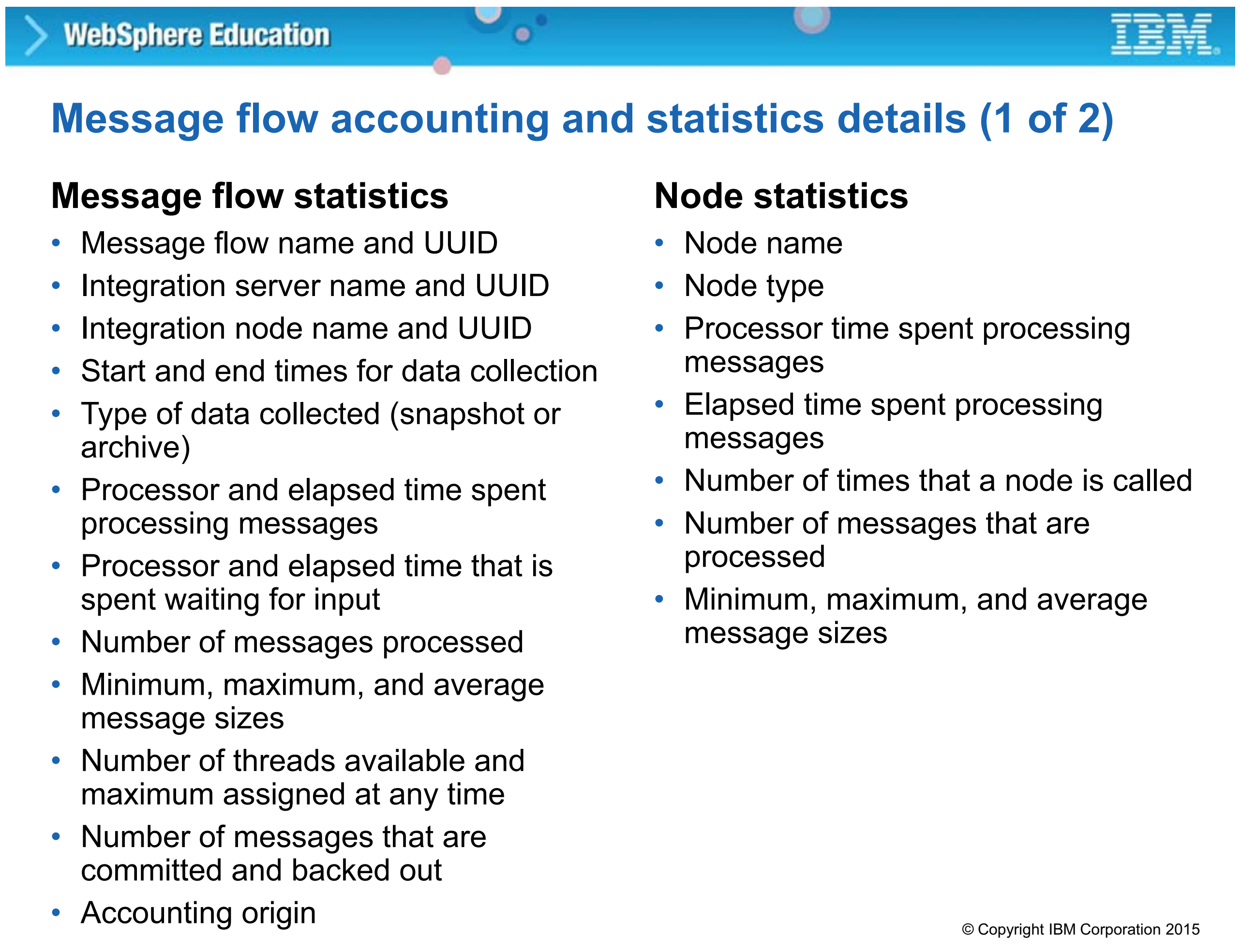
Notes:

You can configure your message flow to collect statistics about the operation and behavior of your message flows. Use this information to assess the performance of your message flow.

Note

The integration node takes information about statistics and accounting from the operating system. On some operating systems, such as Windows, Linux, and UNIX, rounding can occur because the system calls that are used to determine the processor times are not sufficiently granular. This rounding might affect the accuracy of the data.

The effect of monitoring on integration node performance is minimal. However, collecting more data than the default message flow statistics can generate high volumes of report data that might affect performance slightly.



WebSphere Education
IBM

Message flow accounting and statistics details (1 of 2)

Message flow statistics

- Message flow name and UUID
- Integration server name and UUID
- Integration node name and UUID
- Start and end times for data collection
- Type of data collected (snapshot or archive)
- Processor and elapsed time spent processing messages
- Processor and elapsed time that is spent waiting for input
- Number of messages processed
- Minimum, maximum, and average message sizes
- Number of threads available and maximum assigned at any time
- Number of messages that are committed and backed out
- Accounting origin

Node statistics

- Node name
- Node type
- Processor time spent processing messages
- Elapsed time spent processing messages
- Number of times that a node is called
- Number of messages that are processed
- Minimum, maximum, and average message sizes

© Copyright IBM Corporation 2015

Figure 10-24. Message flow accounting and statistics details (1 of 2) WM6461.0

Notes:

Integration Bus accounting and statistics facility collects four types of data records. These records correspond to message flows, processing nodes, node terminals, and threads (message flow instances).

Message flow statistics include the integration server name, integration node name, and message flow name. You also receive the message flow labels and UUIDs. Message flow statistics also include:

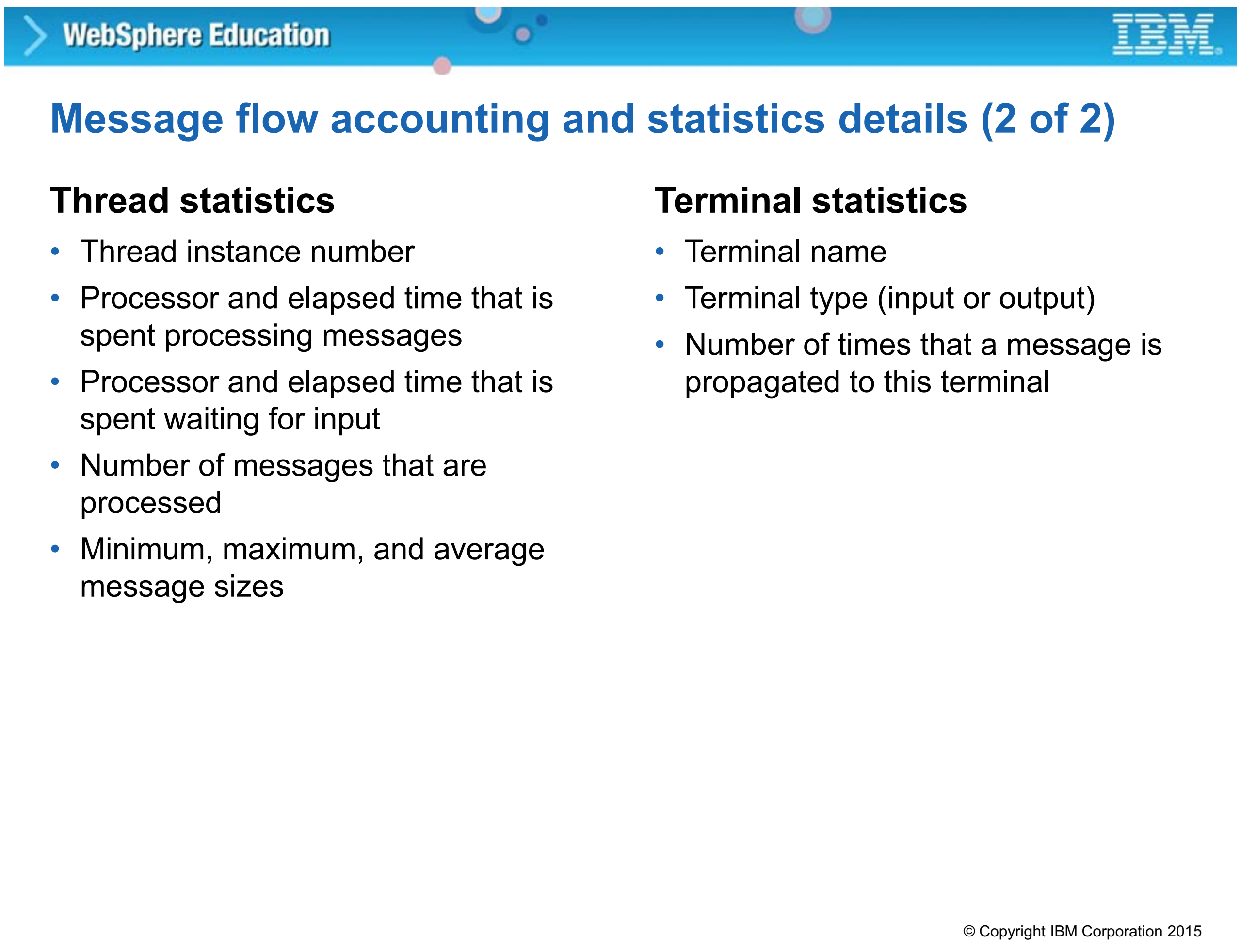
- A **Type of data collected** field, which is contained in each record of message flow statistics data. This field indicates whether the data was collected during an archive (long-term) or snapshot (short-term) interval. Since it is possible to record both during the same interval, it is up to you to make sure that events are not counted twice during analysis.
- The number of messages and their minimum, maximum, and average size during the interval. Also provided is the total number of messages and bytes processed in the interval by a message flow.

- Elapsed, processor, and wait times. These statistics show how much processor and elapsed time the flow that is needed to process messages. For cases where I/O is an important factor, the wait times might prove more interesting.
- Time that a message flow is waiting for a message to process. This metric gives you an idea of the activity level of message flows.
- Total and maximum threads the flow was assigned, and the maximum in use during the interval. Use the thread statistics to determine whether the BAR file variable, **Additional Instances,** is set appropriately.
- How many messages were backed out or committed during the interval. This metric is a useful statistic to verify that messages are being processed as expected.

Node statistics include one statistical record per node. Node statistics include:

- Elapsed, processor, and wait times. Standard statistical times to give node level granularity of processing. Allows isolation of hot spots, or charging according to specific processing.
- The number of times the node was traversed. This metric is useful to better understand, or identify, a critical path in a message flow.
- Message size, number of messages that are processed, and bytes that are processed. Various statistical sizes of messages that the thread processed.

Data that relates to message size is not collected for certain types of nodes that do not create a message tree from a bit stream. These nodes include WebSphere Adapters nodes (for example, the SAPInput node), the FileInput node, the JMSInput node, or any user-defined input node.



Message flow accounting and statistics details (2 of 2)

Thread statistics

- Thread instance number
- Processor and elapsed time that is spent processing messages
- Processor and elapsed time that is spent waiting for input
- Number of messages that are processed
- Minimum, maximum, and average message sizes

Terminal statistics

- Terminal name
- Terminal type (input or output)
- Number of times that a message is propagated to this terminal

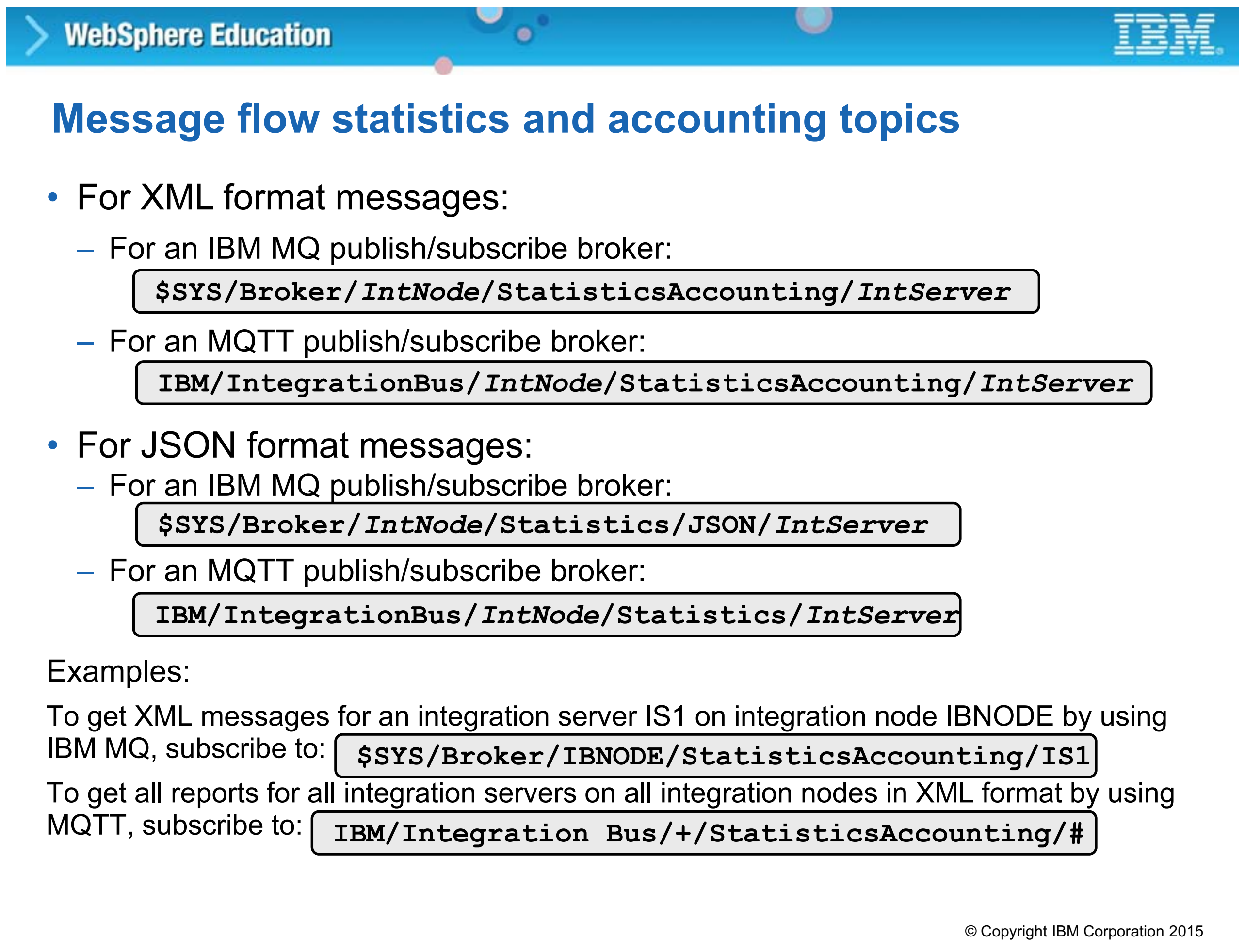
Figure 10-25. Message flow accounting and statistics details (2 of 2) WM6461.0

Notes:

Thread statistics include:

- Arbitrary thread number. This metric has no significance other than identification. One record is produced per thread.
- Total messages that the thread processed during the interval.
- Elapsed, processor, wait, and input wait times. Processing times are similar to the times recorded for message flows, but you can see how effectively the (number of) threads are (or are not) supporting the message flow processing needs.
- Minimum and maximum message input size. Various statistical sizes of messages that the thread processed.

Terminals statistics provide one record for each node terminal. Statistics include labels of each terminal within a node and count of the number of times traversed. This metric is useful for identifying a critical path and exception processing frequency.



Message flow statistics and accounting topics

- For XML format messages:
  - For an IBM MQ publish/subscribe broker:
    `$SYS/Broker/IntNode/StatisticsAccounting/IntServer`
  - For an MQTT publish/subscribe broker:
    `IBM/IntegrationBus/IntNode/StatisticsAccounting/IntServer`
- For JSON format messages:
  - For an IBM MQ publish/subscribe broker:
    `$SYS/Broker/IntNode/Statistics/JSON/IntServer`
  - For an MQTT publish/subscribe broker:
    `IBM/IntegrationBus/IntNode/Statistics/IntServer`

Examples:

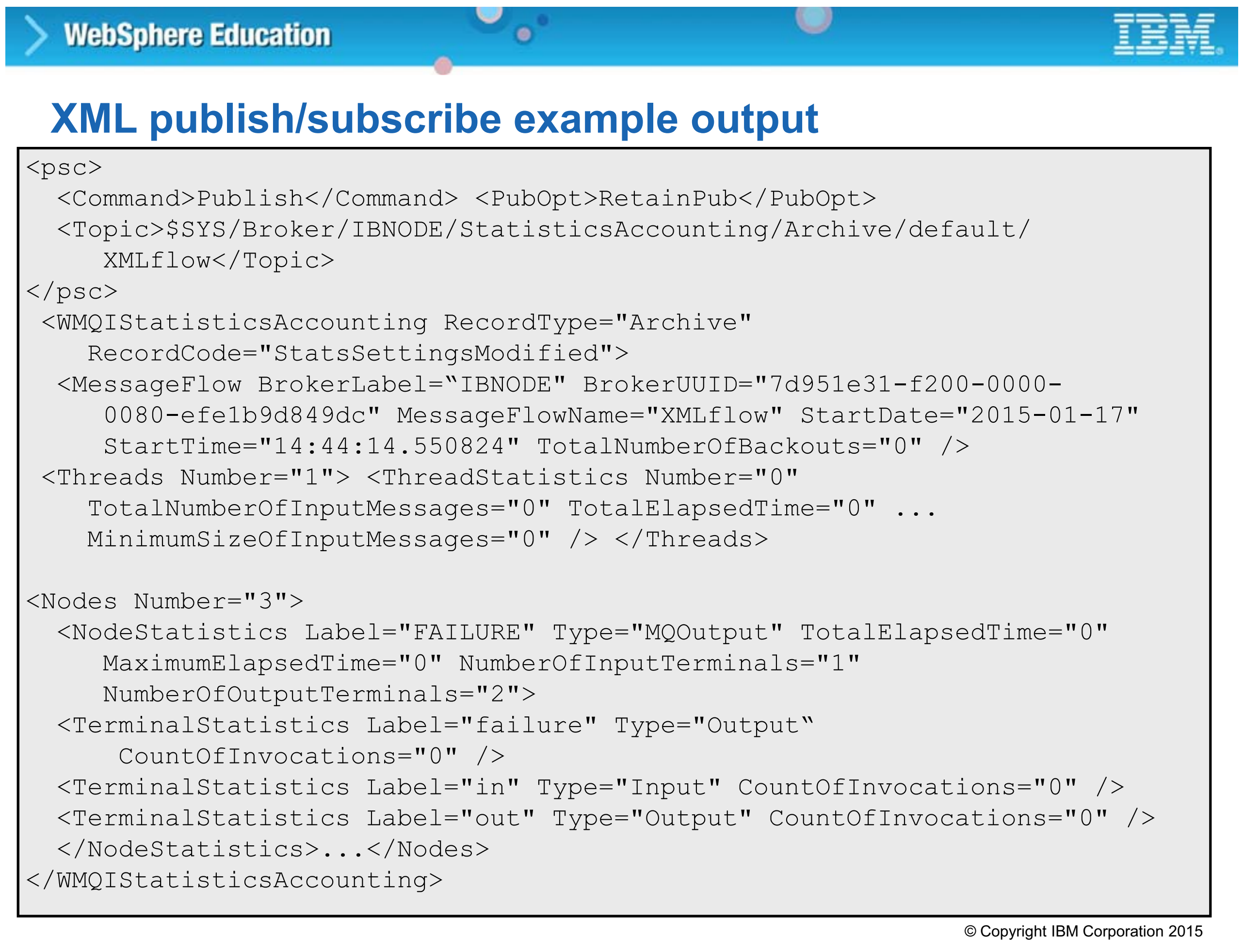
To get XML messages for an integration server IS1 on integration node IBNODE by using IBM MQ, subscribe to: `$SYS/Broker/IBNODE/StatisticsAccounting/IS1`

To get all reports for all integration servers on all integration nodes in XML format by using MQTT, subscribe to: `IBM/Integration Bus/+/StatisticsAccounting/#`

Figure 10-26. Message flow statistics and accounting topics WM6461.0

Notes:

This figure describes the topic strings that Integration Bus generates for message flow statistics and accounting topics for each type of publish/subscribe broker.



WebSphere Education

IBM

XML publish/subscribe example output

```
<psc>
  <Command>Publish</Command> <PubOpt>RetainPub</PubOpt>
  <Topic>$SYS/Broker/IBNODE/StatisticsAccounting/Archive/default/
     XMLflow</Topic>
</psc>
 <WMQIStatisticsAccounting RecordType="Archive"
    RecordCode="StatsSettingsModified">
  <MessageFlow BrokerLabel=“IBNODE" BrokerUUID="7d951e31-f200-0000-
     0080-efe1b9d849dc" MessageFlowName="XMLflow" StartDate="2015-01-17"
     StartTime="14:44:14.550824" TotalNumberOfBackouts="0" />
 <Threads Number="1"> <ThreadStatistics Number="0"
    TotalNumberOfInputMessages="0" TotalElapsedTime="0" ...
    MinimumSizeOfInputMessages="0" /> </Threads>

<Nodes Number="3">
  <NodeStatistics Label="FAILURE" Type="MQOutput" TotalElapsedTime="0"
     MaximumElapsedTime="0" NumberOfInputTerminals="1"
     NumberOfOutputTerminals="2">
  <TerminalStatistics Label="failure" Type="Output“
      CountOfInvocations="0" />
  <TerminalStatistics Label="in" Type="Input" CountOfInvocations="0" />
  <TerminalStatistics Label="out" Type="Output" CountOfInvocations="0" />
  </NodeStatistics>...</Nodes>
</WMQIStatisticsAccounting>
```

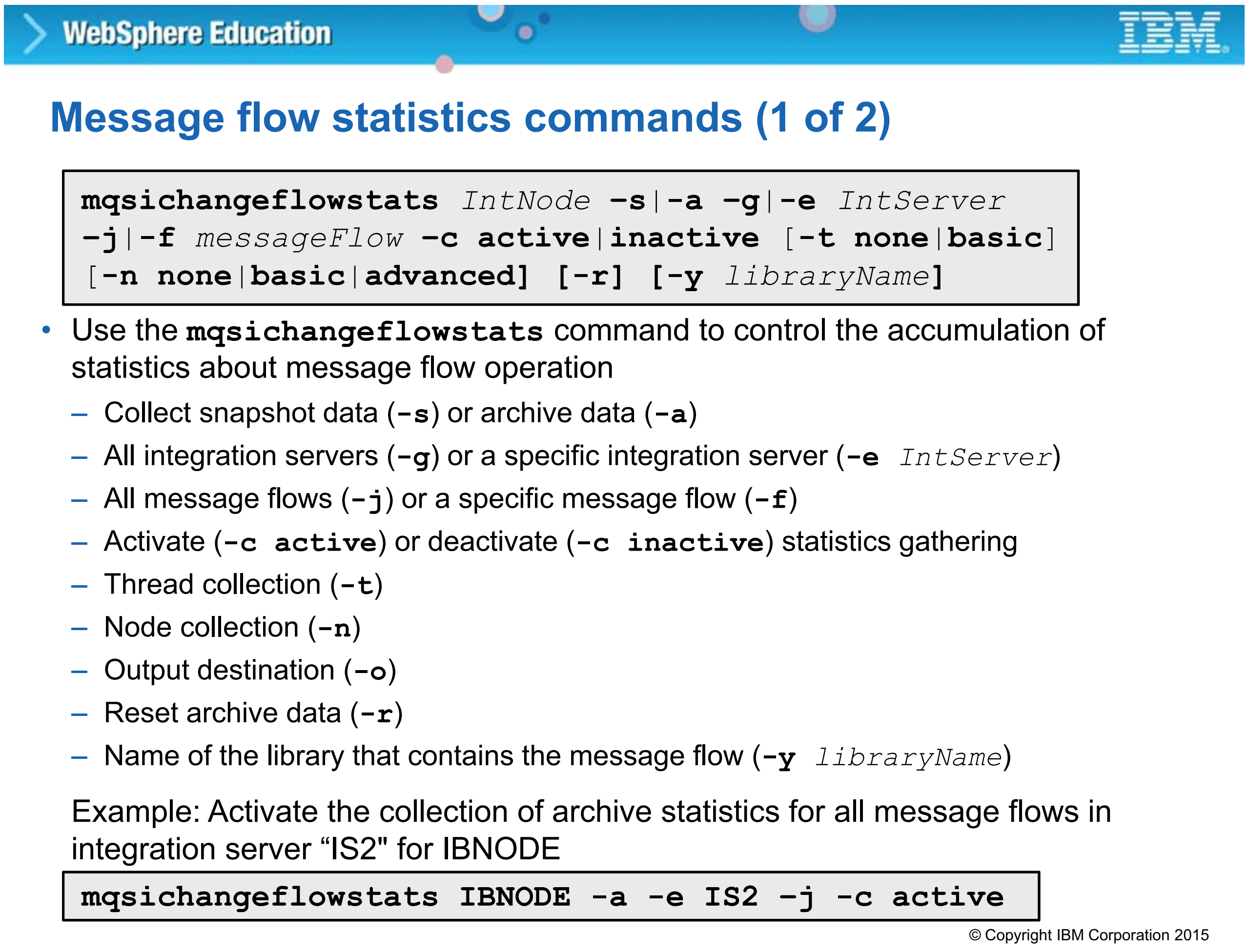
Figure 10-27. XML publish/subscribe example output WM6461.0

Notes:

This figure shows an example of a simple statistics message that is published in XML format. There are five sections in the statistics message:

- The <psc> tag denotes publish/subscribe elements in the MQRFH2.
- The <Message Flow> tag denotes the message flow attributes.
- The <Threads> tag denotes the thread attributes.
- The <Nodes> tag denotes the nodes attributes. This folder can contain several terminal folders, if this level of granularity of reporting was requested.

Both short-term snapshot and longer term archive data messages are published. Since the snapshot data is intended to be used for performance analysis (being sent to a real-time monitor, where, for example, it is graphically displayed), it is published as a retained and nonpersistent message. When archive data is used for accounting and long-term performance, an audit trail is more important. It is published persistently and retained so that late subscribers can get the most recent update.



WebSphere Education
IBM

Message flow statistics commands (1 of 2)

```
mqsichangeflowstats IntNode -s|-a -g|-e IntServer
-j|-f messageFlow -c active|inactive [-t none|basic]
[-n none|basic|advanced] [-r] [-y libraryName]
```

- Use the **`mqsichangeflowstats`** command to control the accumulation of statistics about message flow operation
  - Collect snapshot data (**`-s`**) or archive data (**`-a`**)
  - All integration servers (**`-g`**) or a specific integration server (**`-e`** *`IntServer`*)
  - All message flows (**`-j`**) or a specific message flow (**`-f`**)
  - Activate (**`-c active`**) or deactivate (**`-c inactive`**) statistics gathering
  - Thread collection (**`-t`**)
  - Node collection (**`-n`**)
  - Output destination (**`-o`**)
  - Reset archive data (**`-r`**)
  - Name of the library that contains the message flow (**`-y`** *`libraryName`*)

  Example: Activate the collection of archive statistics for all message flows in integration server “IS2" for IBNODE

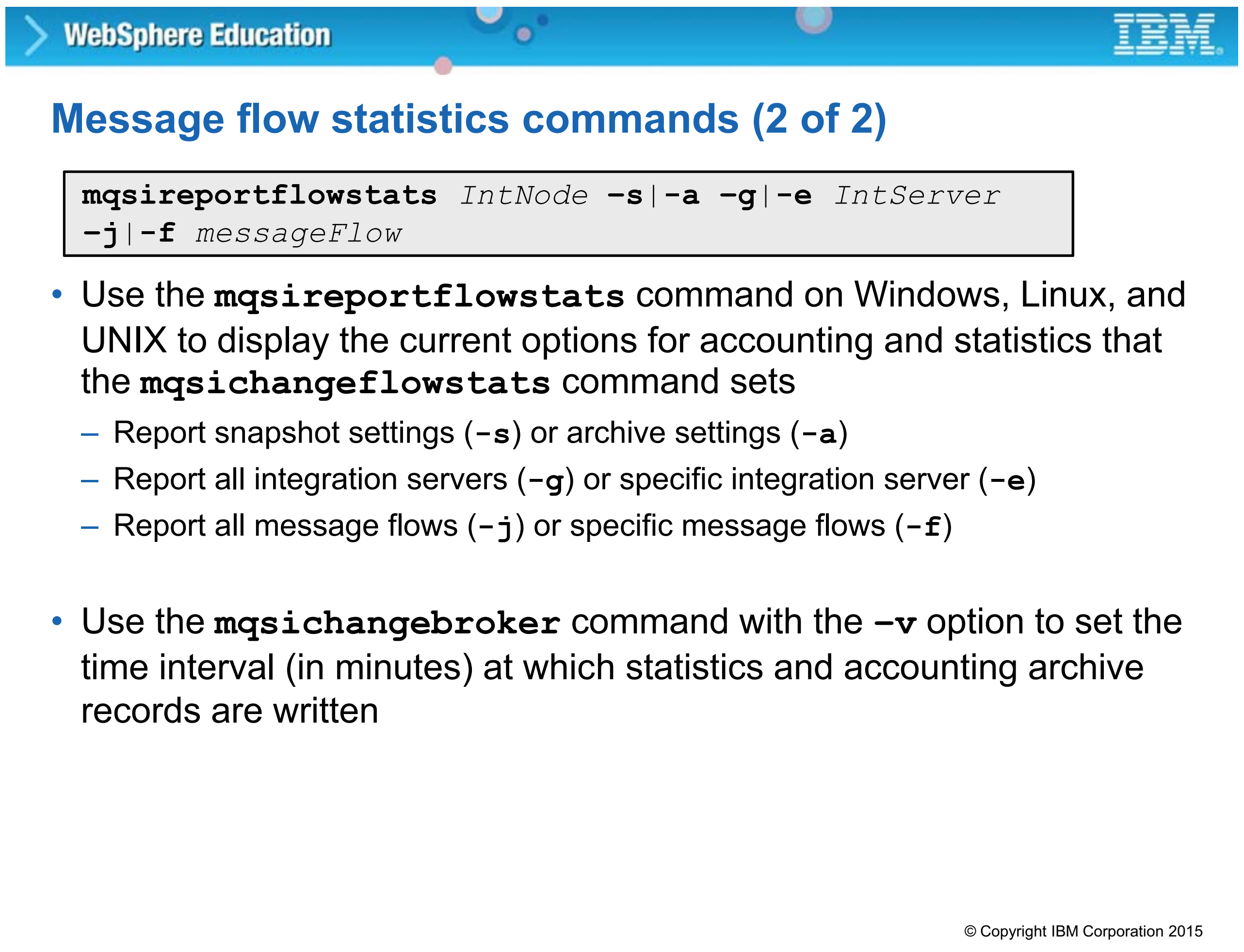
```
mqsichangeflowstats IBNODE -a -e IS2 -j -c active
```

Figure 10-28. Message flow statistics commands (1 of 2) WM6461.0

Notes:

Use the `mqsichangeflowstats` command to complete the following actions:

- Turn on or off accounting and statistics snapshot publication, or archive record output.
- Specify that the command is applied to a specific message flow, or all flows in an integration server, or all integration servers that belong to an integration node.
- Modify the granularity of the data that is collected in addition to the standard message flow accounting and statistics. This extra data can include thread-related data, node-related data, node terminal-related data, or a mixture of this data.
- Specify the output destination. User trace is the default output destination on all operating systems.



WebSphere Education

Message flow statistics commands (2 of 2)

```
mqsireportflowstats IntNode -s|-a -g|-e IntServer
-j|-f messageFlow
```

- Use the **`mqsireportflowstats`** command on Windows, Linux, and UNIX to display the current options for accounting and statistics that the **`mqsichangeflowstats`** command sets
  - Report snapshot settings (**`-s`**) or archive settings (**`-a`**)
  - Report all integration servers (**`-g`**) or specific integration server (**`-e`**)
  - Report all message flows (**`-j`**) or specific message flows (**`-f`**)
- Use the **`mqsichangebroker`** command with the **`–v`** option to set the time interval (in minutes) at which statistics and accounting archive records are written

Figure 10-29. Message flow statistics commands (2 of 2) WM6461.0

Notes:

Use the `mqsireportflowstats` command to display the current options for accounting and statistics that were set with the `mqsichangeflowstats` command. The `mqsichangeflowstats` command has the following syntax requirements:

- You must specify at least one option for snapshot settings (`-a`, `-s`, or both).
- You must either specify a specific integration server (`-e`) or all integration servers (`-g`).
- You must either specify a specific message flow (`-f`) or all message flows (`-j`).

When defining or modifying an integration node by using a command, use the `-v` flag to specify the time interval (in minutes) at which statistics and accounting archive records are written. For internal accounting, the valid range is 10 - 14400 minutes; the default value is 60. An interval of zero minutes indicates that the operating system has an external method of notification (the ENF timer), and is not using an internal timer within Integration Bus.

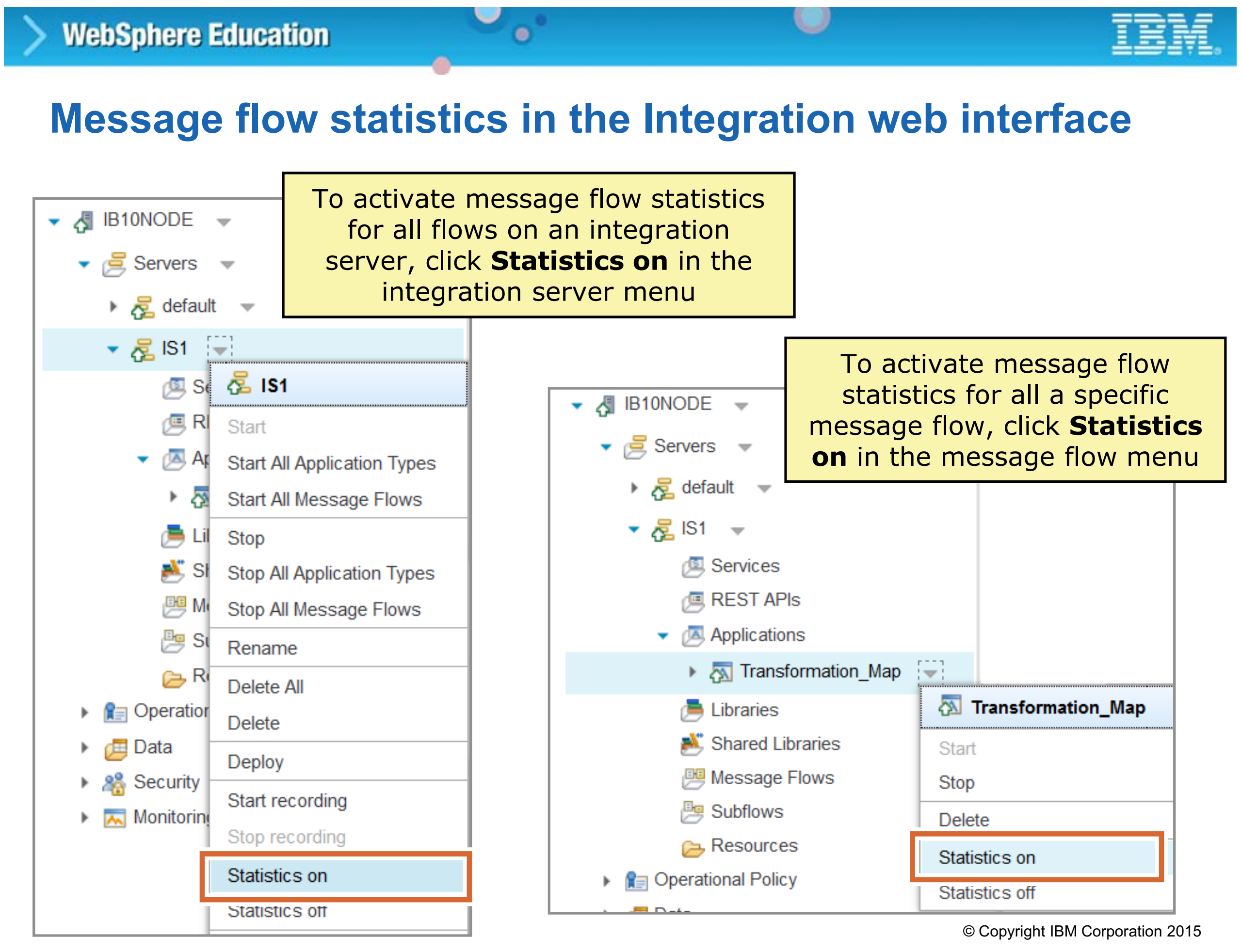

WebSphere Education
IBM
Message flow statistics in the Integration web interface
To activate message flow statistics for all flows on an integration server, click Statistics on in the integration server menu
To activate message flow statistics for all a specific message flow, click Statistics on in the message flow menu
IB10NODE
Servers
default
IS1
Start
Start All Application Types
Start All Message Flows
Stop
Stop All Application Types
Stop All Message Flows
Rename
Delete All
Delete
Deploy
Start recording
Stop recording
Statistics on
Statistics off
Operational Policy
Data
Security
Monitoring
Services
REST APIs
Applications
Transformation_Map
Libraries
Shared Libraries
Message Flows
Subflows
Resources
© Copyright IBM Corporation 2015

Figure 10-30. Message flow statistics in the Integration web interface WM6461.0

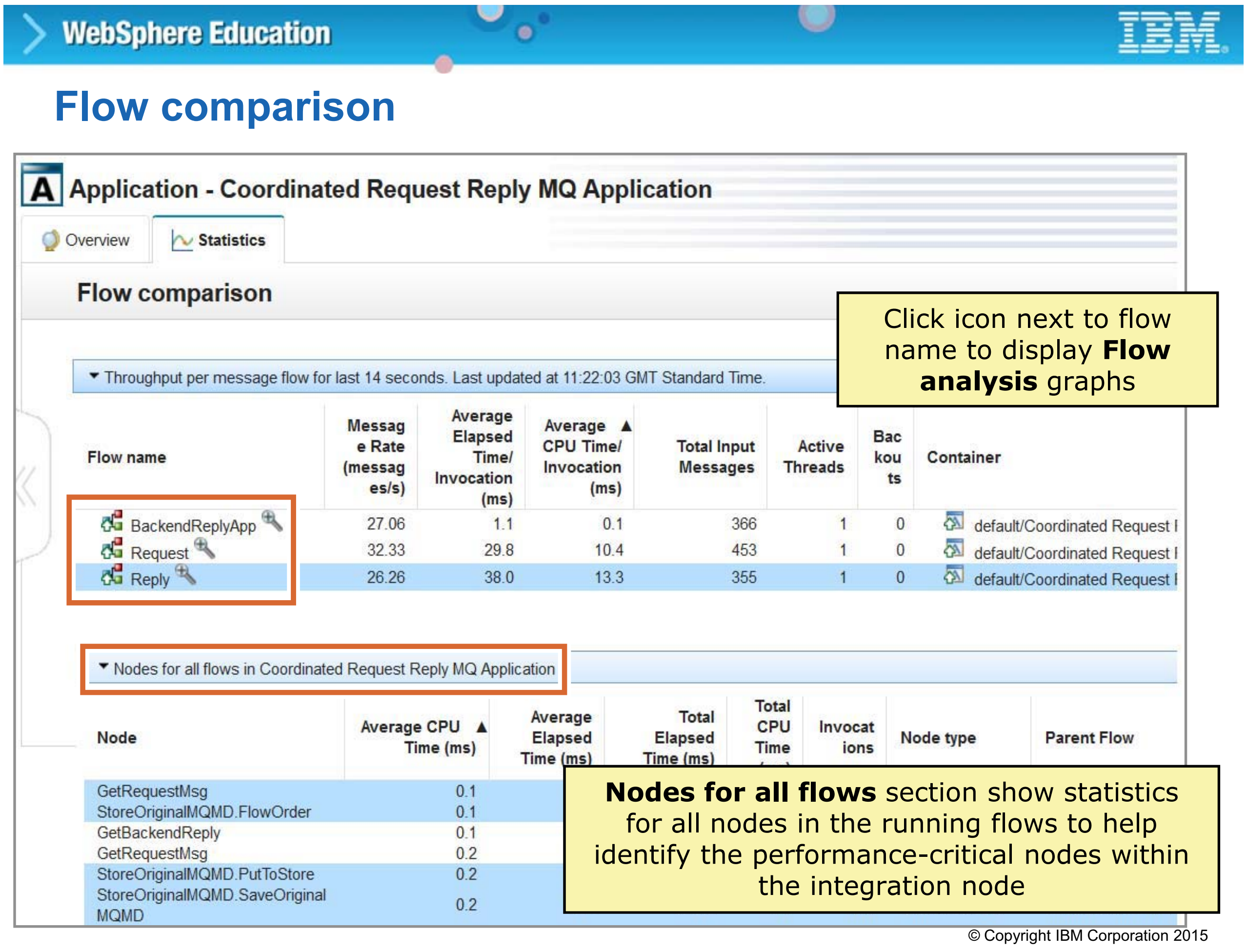
Notes:

You can use the Integration web interface to start collecting snapshot accounting and statistics data for your message flows. You can then view the accounting and statistics data in the Integration web interface.

To start collecting snapshot message flow accounting and statistics data by using the web console, click the arrow next to the required message flow in the navigation tree, and then click **Statistics on** in the menu.

You can start the collection of statistics data for all message flows in an integration node, integration server, or application, by clicking the arrow next to the required resource and then clicking **Statistics on** in the menu.

Statistics collection for the selected message flows is turned on, and an up-to-date snapshot of information is collected every 20 seconds.



Flow comparison

A Application - Coordinated Request Reply MQ Application

Overview | Statistics

Flow comparison

▾ Throughput per message flow for last 14 seconds. Last updated at 11:22:03 GMT Standard Time.

Flow name	Message Rate (messages/s)	Average Elapsed Time/ Invocation (ms)	Average CPU Time/ Invocation (ms) ▲	Total Input Messages	Active Threads	Backouts	Container
BackendReplyApp	27.06	1.1	0.1	366	1	0	default/Coordinated Request I
Request	32.33	29.8	10.4	453	1	0	default/Coordinated Request I
Reply	26.26	38.0	13.3	355	1	0	default/Coordinated Request I

Click icon next to flow name to display **Flow analysis** graphs

▾ Nodes for all flows in Coordinated Request Reply MQ Application

Node	Average CPU Time (ms) ▲	Average Elapsed Time (ms)	Total Elapsed Time (ms)	Total CPU Time	Invocations	Node type	Parent Flow
GetRequestMsg	0.1						
StoreOriginalMQMD.FlowOrder	0.1						
GetBackendReply	0.1						
GetRequestMsg	0.2						
StoreOriginalMQMD.PutToStore	0.2						
StoreOriginalMQMD.SaveOriginalMQMD	0.2						

Nodes for all flows section show statistics for all nodes in the running flows to help identify the performance-critical nodes within the integration node

© Copyright IBM Corporation 2015

Figure 10-31. Flow comparison WM6461.0

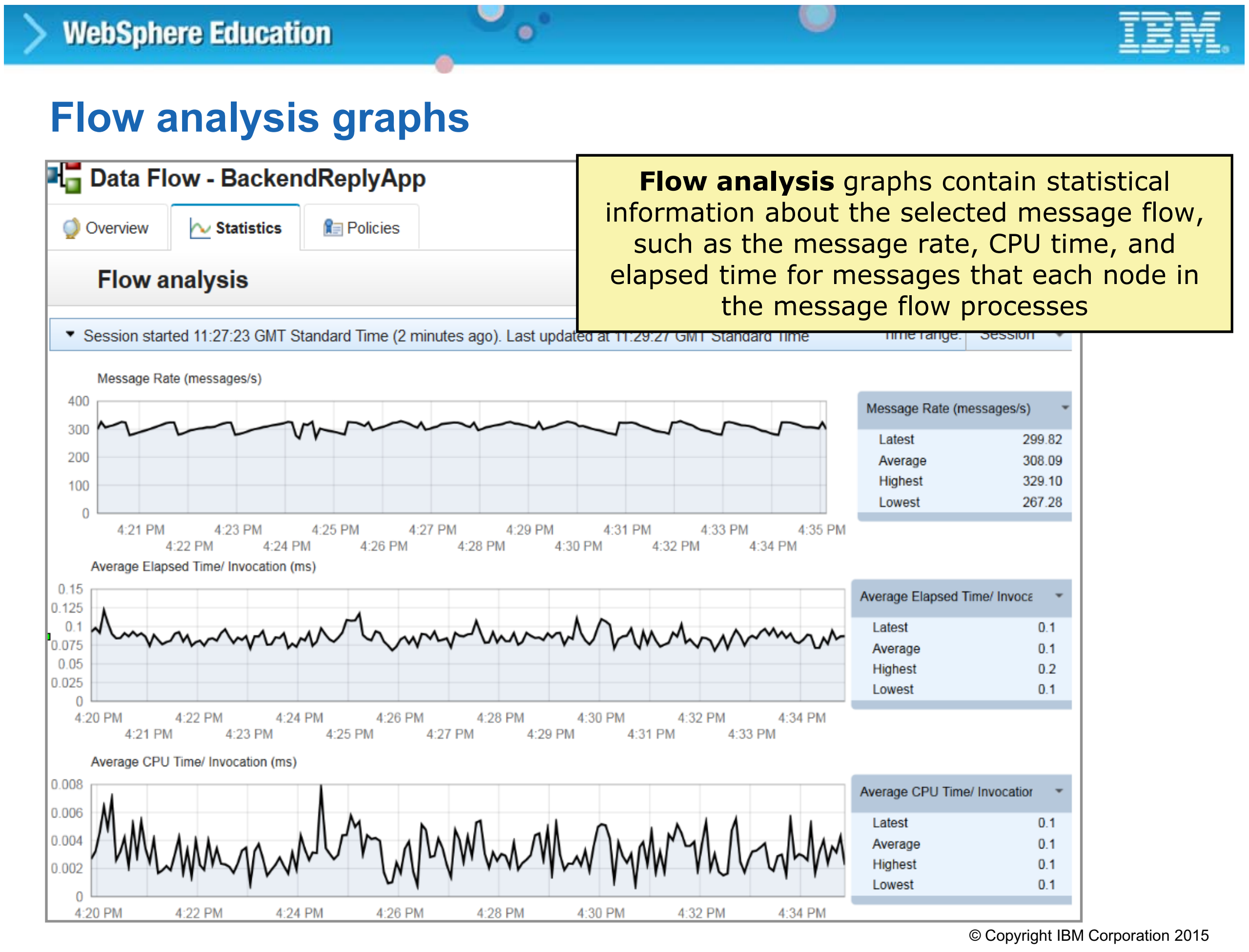
Notes:

To see an overview of statistical information for all message flows in an integration node, integration server, application, library, or service, click the name of the resource in the navigation tree, and then click the **Statistics** tab.

You can use the information on the **Flow comparison** section of the **Statistics** tab to help you identify aspects of the message flows that might benefit from optimization.

The **Throughput per message flow** section shows statistics for the throughput of messages in all message flows in the selected integration node, integration server, application, or library, during the last 20-second reporting period.

The **Node for all flows** section shows statistical information about all the message flow nodes in the running message flows that have statistics collection turned on, for the selected integration node, integration server, application, or library.



Flow analysis graphs

Data Flow - BackendReplyApp

Overview | Statistics | Policies

Flow analysis

Flow analysis graphs contain statistical information about the selected message flow, such as the message rate, CPU time, and elapsed time for messages that each node in the message flow processes

Session started 11:27:23 GMT Standard Time (2 minutes ago). Last updated at 11:29:27 GMT Standard Time

Message Rate (messages/s)

Message Rate (messages/s)	
Latest	299.82
Average	308.09
Highest	329.10
Lowest	267.28

Average Elapsed Time/ Invocation (ms)

Average Elapsed Time/ Invoca	
Latest	0.1
Average	0.1
Highest	0.2
Lowest	0.1

Average CPU Time/ Invocation (ms)

Average CPU Time/ Invocatior	
Latest	0.1
Average	0.1
Highest	0.1
Lowest	0.1

© Copyright IBM Corporation 2015

Figure 10-32. Flow analysis graphs WM6461.0

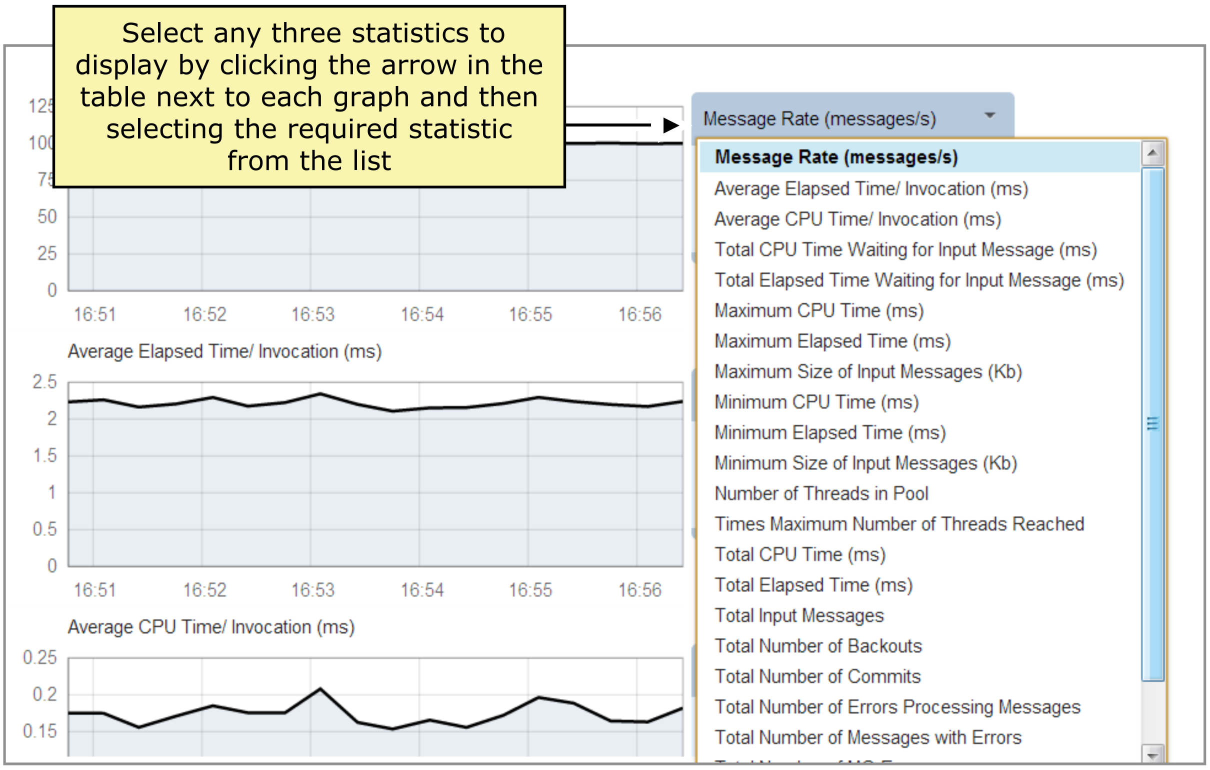
Notes:

Use the information in the **Flow analysis** section to assess the performance of a message flow. This view contains statistical information about the selected message flow, including data such as the message rate, processor time, and elapsed time for messages that each node in the message flow processes.

The charts in this view show data for three statistics at a time. By default the charts show the message rate, average elapsed time, and average processor time for the selected message flow.

WebSphere Education

Flow analysis graph content


Select any three statistics to display by clicking the arrow in the table next to each graph and then selecting the required statistic from the list
Average Elapsed Time/ Invocation (ms)
Average CPU Time/ Invocation (ms)
16:51
16:52
16:53
16:54
16:55
16:56
Message Rate (messages/s)
Message Rate (messages/s)
Average Elapsed Time/ Invocation (ms)
Average CPU Time/ Invocation (ms)
Total CPU Time Waiting for Input Message (ms)
Total Elapsed Time Waiting for Input Message (ms)
Maximum CPU Time (ms)
Maximum Elapsed Time (ms)
Maximum Size of Input Messages (Kb)
Minimum CPU Time (ms)
Minimum Elapsed Time (ms)
Minimum Size of Input Messages (Kb)
Number of Threads in Pool
Times Maximum Number of Threads Reached
Total CPU Time (ms)
Total Elapsed Time (ms)
Total Input Messages
Total Number of Backouts
Total Number of Commits
Total Number of Errors Processing Messages
Total Number of Messages with Errors

© Copyright IBM Corporation 2015

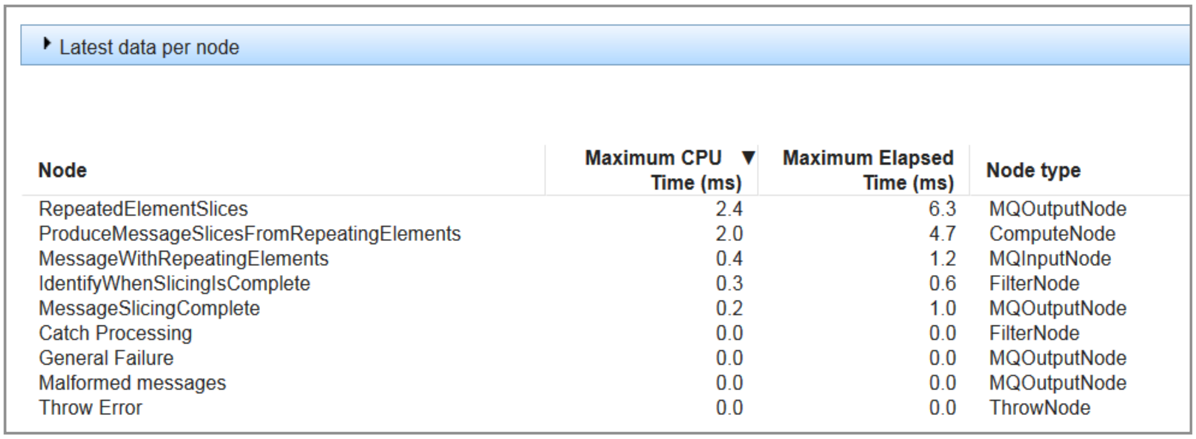
Figure 10-33. Flow analysis graph content WM6461.0

Notes:

You can select any three statistics to display in the charts by clicking the arrow in the table next to each graph and then selecting the required statistic from the list. The tables show the highest, lowest, and average data for the whole of the current browser session, and the data for the last 20-second reporting period.

WebSphere Education

Flow analysis data per node



Latest data per node

Node	Maximum CPU Time (ms) ▼	Maximum Elapsed Time (ms)	Node type
RepeatedElementSlices	2.4	6.3	MQOutputNode
ProduceMessageSlicesFromRepeatingElements	2.0	4.7	ComputeNode
MessageWithRepeatingElements	0.4	1.2	MQInputNode
IdentifyWhenSlicingIsComplete	0.3	0.6	FilterNode
MessageSlicingComplete	0.2	1.0	MQOutputNode
Catch Processing	0.0	0.0	FilterNode
General Failure	0.0	0.0	MQOutputNode
Malformed messages	0.0	0.0	MQOutputNode
Throw Error	0.0	0.0	ThrowNode

- Shows elapsed and CPU time per message that the node processes
- Can choose between minimum, average, or maximum per message
- Also shows the type of each node, which can help to identify the potential cause of any performance problems

Figure 10-34. Flow analysis data per node WM6461.0

Notes:

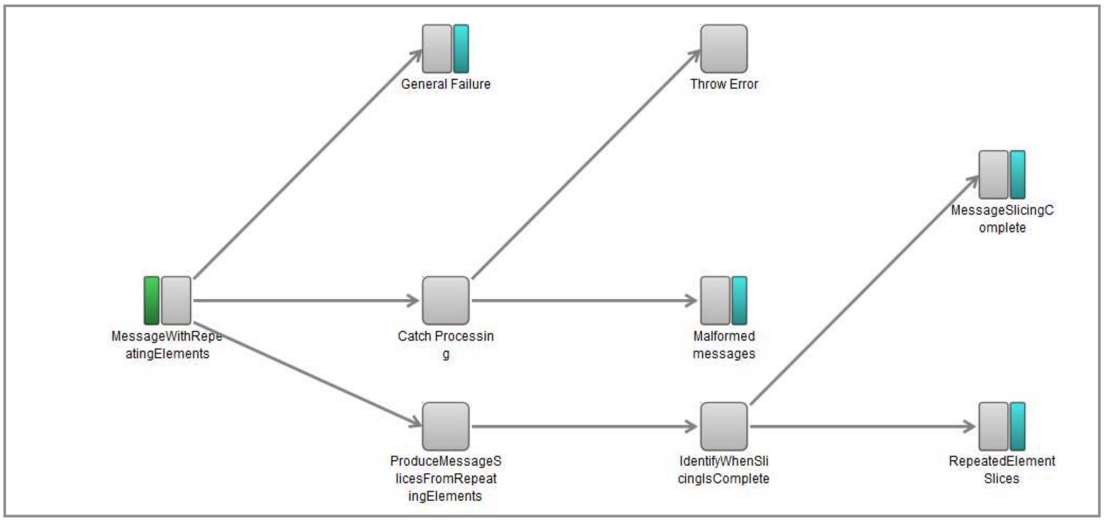
The **Latest data per node** section shows statistical data for each node in the selected message flow, including the elapsed and processor time per message that the node processes.

You can choose to display the minimum, average, or maximum processor and elapsed times per message, by clicking **Minimum**, **Average**, or **Maximum** in the section heading.

You can also sort the data in the table by clicking the heading of a column; for example, you can sort the data by processor time by clicking the heading **CPU time (ms)**.

This table also shows the type of each node, which can help you to identify the potential cause of any performance problem.

Flow analysis flow profile



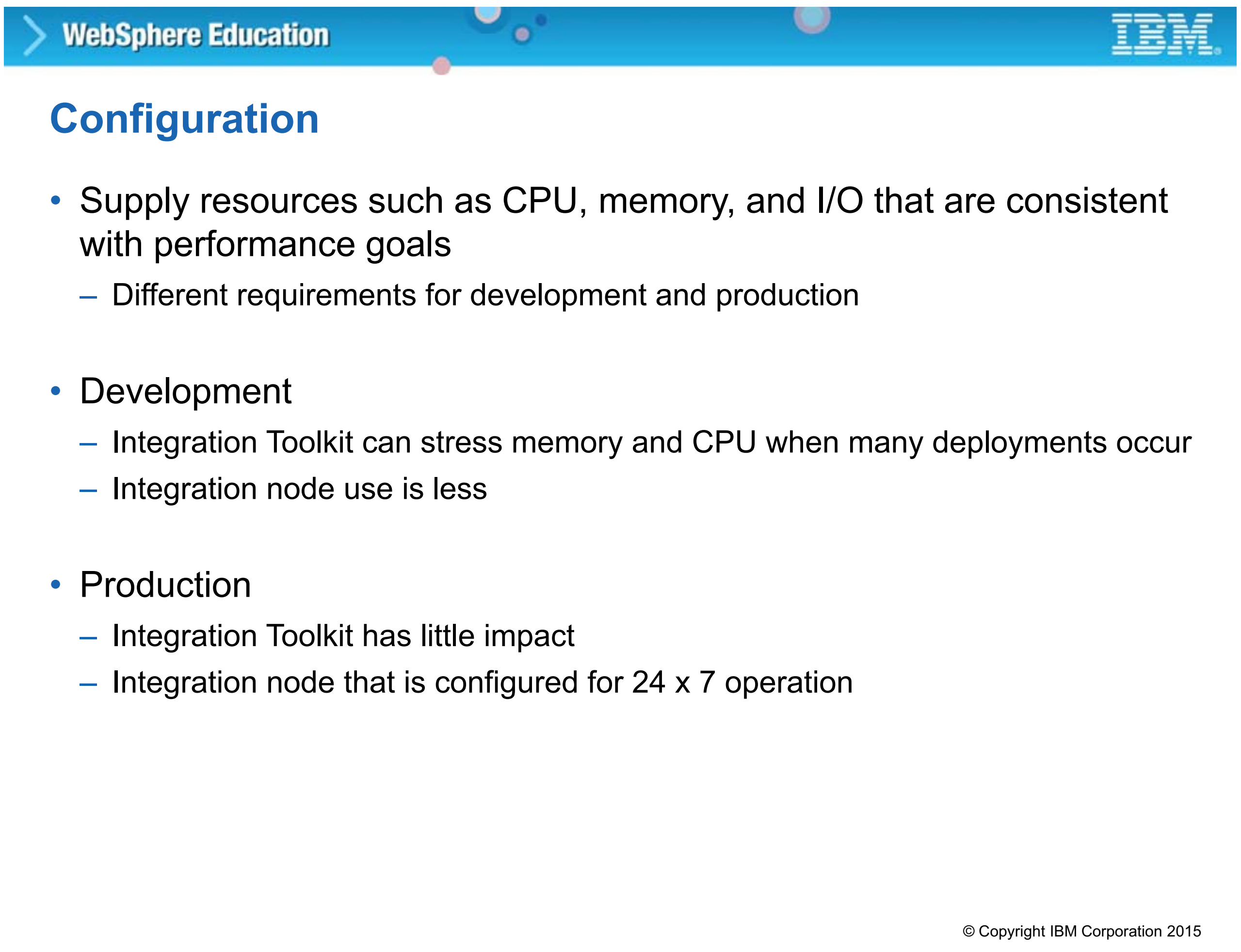
- Graphical representation of the selected flow, displaying all nodes in the flow
- Helps you understand the message flow performance statistics

Figure 10-35. Flow analysis flow profile WM6461.0

Notes:

The **Flow profile** section on the Integration web interface **Statistics** tab shows a graphical representation of the selected message flow, displaying all nodes in the flow.

10.4.Performance tuning



Configuration

- Supply resources such as CPU, memory, and I/O that are consistent with performance goals
  - Different requirements for development and production
- Development
  - Integration Toolkit can stress memory and CPU when many deployments occur
  - Integration node use is less
- Production
  - Integration Toolkit has little impact
  - Integration node that is configured for 24 x 7 operation

Figure 10-36. Configuration WM6461.0

Notes:

To get the level of performance that you expect, it is essential to have sufficient resources that are allocated. These include processor speed, the number of processors, memory, and the I/O subsystem.

It is important to consider the difference in resource demands in a development environment and a production environment. A development environment might host the IBM Integration Toolkit and a test integration node, while a production environment might have many flows on many integration nodes and integration servers.

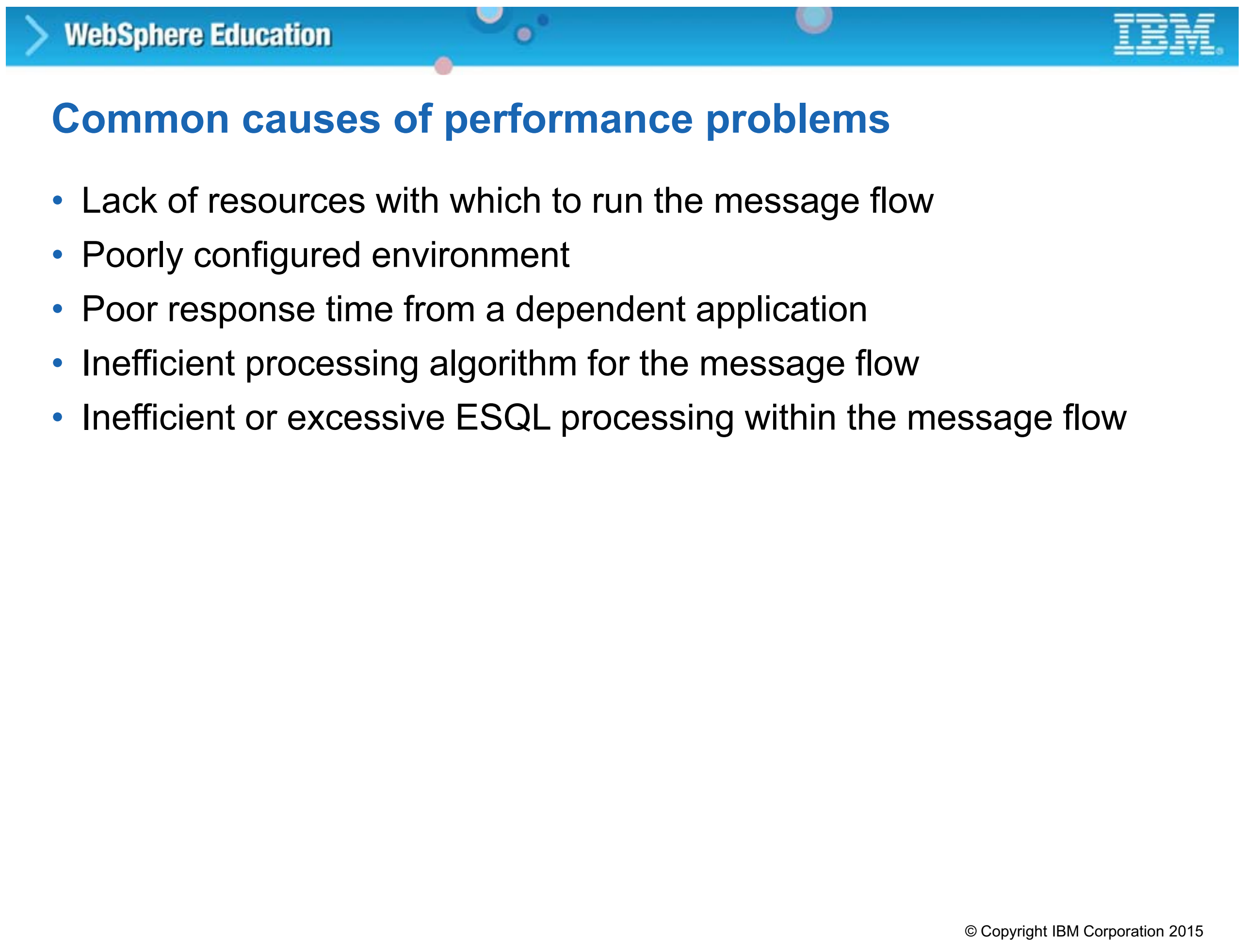
There is often a tendency to concentrate on just the processors and memory and pay little attention to I/O. You can achieve significant gains in performance by using disks with a fast write nonvolatile cache for the queue manager log. Doing so can lead to a substantial increase in the persistent message rate that you are able to achieve.

Think about the different levels of resource that you are likely to need for developing message flows compared to running them in production.

In the development environment, a user’s time is spent with the workbench, creating, and testing (deploying) message flows. The volume of messages that the integration node handles is low. The integration node has much lower requirements.

When running in a production environment (presumably 24 hours per day, 7 days per week), it is highly likely that the workbench has little use. The integration node is the busy component as it is continually processing messages.

The key to good configuration is to match the capabilities of the equipment with the needs of the job.



WebSphere Education

Common causes of performance problems

- Lack of resources with which to run the message flow
- Poorly configured environment
- Poor response time from a dependent application
- Inefficient processing algorithm for the message flow
- Inefficient or excessive ESQL processing within the message flow

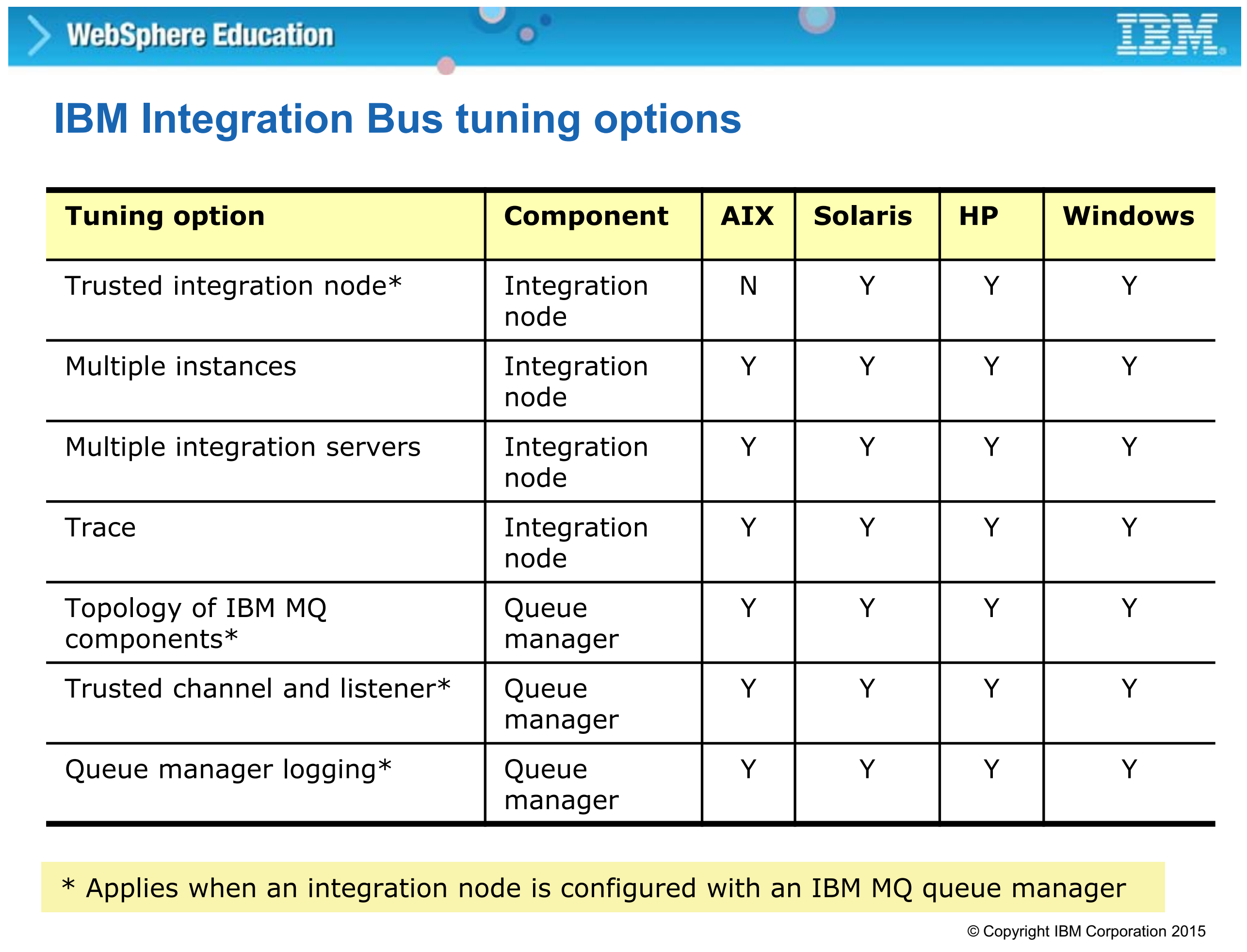
© Copyright IBM Corporation 2015

Figure 10-37. Common causes of performance problems WM6461.0

Notes:

The figure lists some of the most common causes of performance problems.

Some of these problems, such as resource allocation and environment are under the administrator’s control; other problems, such as inefficient ESQL is not.



WebSphere Education
IBM

IBM Integration Bus tuning options

Tuning option	Component	AIX	Solaris	HP	Windows
Trusted integration node*	Integration node	N	Y	Y	Y
Multiple instances	Integration node	Y	Y	Y	Y
Multiple integration servers	Integration node	Y	Y	Y	Y
Trace	Integration node	Y	Y	Y	Y
Topology of IBM MQ components*	Queue manager	Y	Y	Y	Y
Trusted channel and listener*	Queue manager	Y	Y	Y	Y
Queue manager logging*	Queue manager	Y	Y	Y	Y

* Applies when an integration node is configured with an IBM MQ queue manager

Figure 10-38. IBM Integration Bus tuning options WM6461.0

Notes:

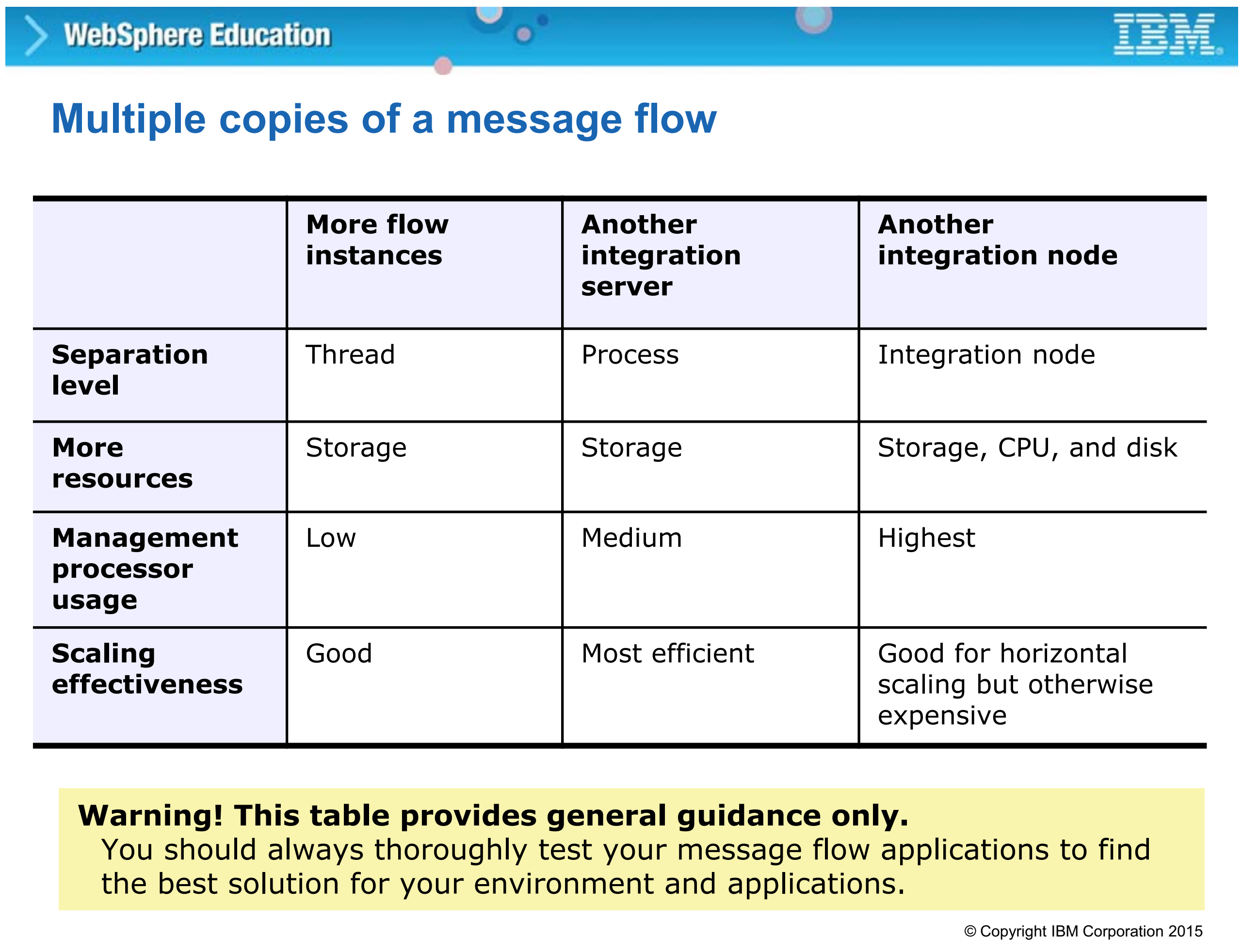
Tuning options cover the integration node, its associated queue manager (if present), and any databases that contain business data across each of the operating systems on which Integration Bus is implemented.

In the figure, a value of Y indicates that the option applies or is available. It is not the default value but it can be enabled. A value of N indicates that the option is not available or is not recommended. You look at each of these options in more detail later in this topic.

On some operating systems, it is possible to run an integration node as an IBM MQ trusted application to streamline MQPUT and MQGET calls from the integration node. Performance is improved because the integration node interacts directly with the queue manager rather than an intermediary agent process. The increase in performance is at the cost of not isolating the queue manager from errant program behavior, such as poorly coded plug-in nodes, which can cause the integration node queue manager to fail. An integration node can be modified to run as a trusted IBM MQ application by using the `mqsichangebroker` command with the `-t` flag.

The benefit that is obtained from running an integration node in trusted mode depends on the processing within the message flows. Different message flows running in the same integration node might see different results. A message flow that has a high proportion of MQPUT or MQGET processing of nonpersistent messages experiences more benefit. This processing is equivalent to a simple MQInput to MQOutput test.

Message flows in which most of the processing is manipulating the contents of a message might not benefit from using a trusted integration node.



WebSphere Education

Multiple copies of a message flow

	More flow instances	Another integration server	Another integration node
Separation level	Thread	Process	Integration node
More resources	Storage	Storage	Storage, CPU, and disk
Management processor usage	Low	Medium	Highest
Scaling effectiveness	Good	Most efficient	Good for horizontal scaling but otherwise expensive

Warning! This table provides general guidance only.
You should always thoroughly test your message flow applications to find the best solution for your environment and applications.

Figure 10-39. Multiple copies of a message flow WM6461.0

Notes:

Each message flow instance (within an integration server) requires another thread to run in that integration server process. In some situations, running multiple instances within an integration server does not provide sufficient isolation between message flows. Different integration servers provide separation at the process or address space level, as enforced by the operating system. With multiple integration nodes that are running, it would be possible to run a common message flow on different servers. This scenario provides the maximum level of separation but is the most expensive in terms of hardware resources.

In the figure, the **Management processor usages** row indicates how much extra resources are involved with each of the techniques. The amount of work to add instances is low, since it is only the number of copies in the integration server, which is changed. If there are many integration servers in existence, managing them in the Integration Toolkit can become an issue.

Creating multiple integration nodes has the highest impact on performance. Each integration node queue manager must be connected into the existing topology. As an integration node and integration node queue manager are created, more memory, processor, and disk space are required on the integration node server.

When you must add more copies of a message flow, first increase the **Additional instances** property to increase message throughput. If adding more instances is not an option, try multiple integration servers to fully use the server on which the integration node is running, or to use as much of it as possible. If further increases in message throughput are needed, use more integration nodes. Establish the best use of extra instances and integration servers before replicating the integration node.

When assigning message flows to integration servers, there might be a requirement to not run specific message flows in the same integration server. This restriction might affect the policy that you use to assign message flows to integration servers.

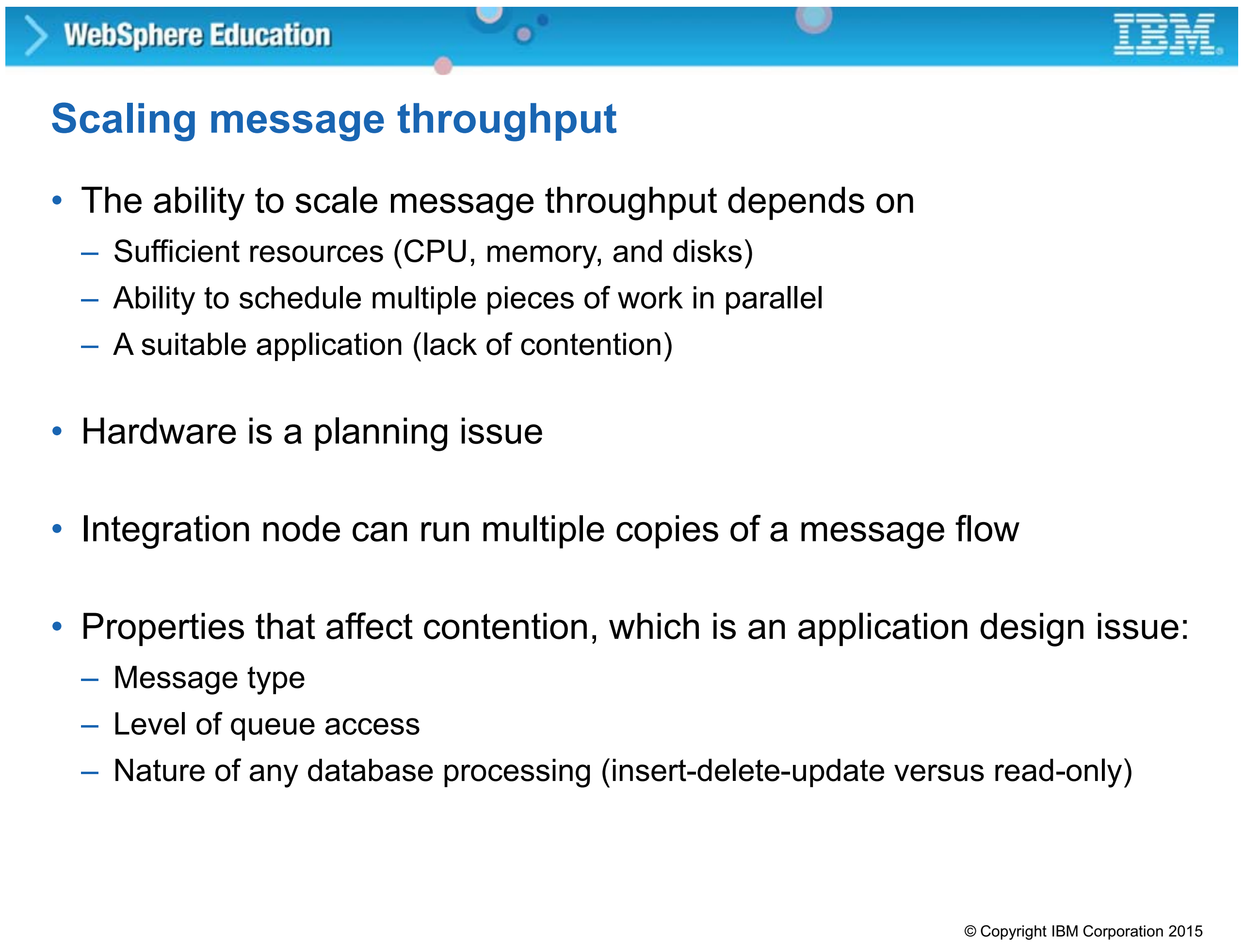
In some situations, a message flow might have heavy resource requirements, such as the need for a large amount of processor or memory usage. By assigning such flows to a restricted number of integration servers, the effects can be limited. This consideration can be true where a message flow has a large high water mark for virtual memory use. If such a flow is assigned to all integration servers, the total virtual memory requirements are large, which places a greater demand on the memory of the server. This scenario can lead to a noticeable increase in paging or swapping. If the message flow is only assigned to a limited number of integration servers, the effect is contained.

To determine how many copies of the message flow are required, you must know (or estimate) what the target message rate is and compare that to the rate achieved with one copy of the message flow.

Next, you must establish an overall goal. Do you want to achieve a particular message rate, reach a particular processor use, or run in the most efficient way (although that might not lead to the maximum message rate)?

Run a single copy of the message flow with a realistic message. If the message flow processes multiple types of messages, make sure that all of the required message types are present and in their correct proportions. Continuously load the input queues to get the message flow to run in a steady state. Measure the message throughput with tools, such as `vmstat`, to also determine the relative state of the system.

From these initial measurements, you should be able to see whether the message flow is processor-bound or I/O-bound. Both of these conditions affect how the message rate increases if you deploy more copies of the message flow. It might also mean that you must change the hardware configuration to overcome an I/O constraint.



WebSphere Education

Scaling message throughput

- The ability to scale message throughput depends on
  - Sufficient resources (CPU, memory, and disks)
  - Ability to schedule multiple pieces of work in parallel
  - A suitable application (lack of contention)
- Hardware is a planning issue
- Integration node can run multiple copies of a message flow
- Properties that affect contention, which is an application design issue:
  - Message type
  - Level of queue access
  - Nature of any database processing (insert-delete-update versus read-only)

Figure 10-40. Scaling message throughput WM6461.0

Notes:

It is difficult to accurately predict the precise benefits that are obtained when running multiple copies of a message flow since it depends on the design of the message flow. However, message flows must have specific characteristics so that they are able to scale well.

- Message flow should do minimal I/O, either queue or database related. I/O normally imposes a lower maximum message rate than is the case when processing is bound to a processor.
- Message flows should do minimal queue access.
- Message flows should do a minimal amount of database insert, delete, and update activity. Read processing is far more scalable.

While the ideal is to produce message flows that give the level of required performance with little tuning of hardware, more benefit might be obtained by upgrading to faster processors or disks.

By adding more copies of a message flow, your aim is to balance throughput and resources. It is often difficult to follow this ideal for a number of reasons. What can often happen is that you reach the limitation of a resource, perhaps the speed at which you can insert messages into a database. In such a case, the improvement of more copies reduces and you start to reach a plateau.

A more extreme case of reducing benefit is where you reduce overall throughput by adding more copies of the message flow. This scenario might occur where you started to experience deadlocks for key resources. The task in this case, is to identify the contention and eliminate it. At a minimum, you should reduce the number of copies of running message flows so that you get the maximum throughput.

The ability to increase message throughput by running more copies of the message flow is an important consideration for many users. They must know that there are no inherent limitations, which prevent them from growing vertically (within a server) and horizontally (across servers) as they require.

To be able to increase throughput with any piece of transaction or messaging software, a number of fundamentals must be in place.

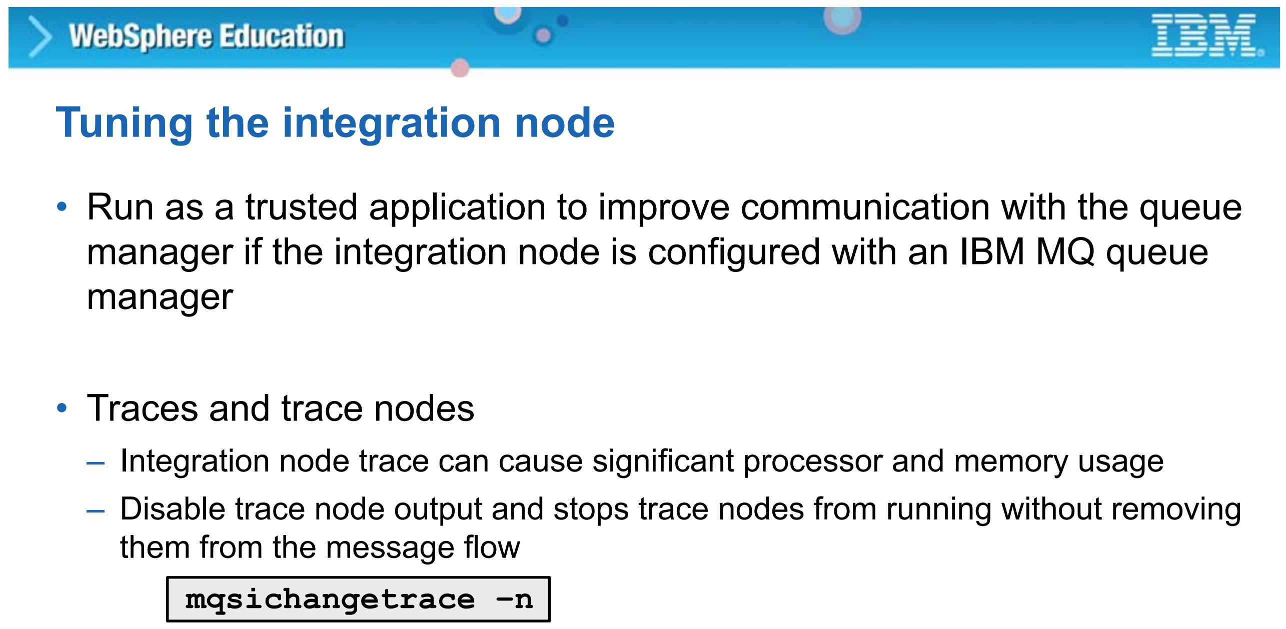
- Physical resources must be in place. That is, processor, memory, and disks.
- There must be the ability to schedule concurrent pieces of work (multiprogramming).
- There must be a minimum of contention within the application so that there is no conflict between running multiple copies concurrently. If each processing element needs access to a central control record, this restriction would be a point of contention and so would limit throughput gains from running multiple copies.

Ensuring that sufficient hardware is available is really a case of recognizing what is required. To do so, you must implement the message flows, test, and measure to adequately determine the requirements.

Contention between message flows is not something you can predict. It depends on how the message flows are coded and the data that they reference. Items to be aware of are high levels of access to few queues and database processing. If messages are located over multiple input queues, there is less chance of contention than if a single queue is used for all message processing.

Message type is also a factor. Persistent messages are locked for a longer time than nonpersistent messages, as I/O takes place in the log. In addition, the maximum message rate that is achieved with persistent messages is lower than for nonpersistent messages as there is an obvious I/O limitation.

Consider carefully any database processing. Avoid control records that multiple message flows must access. Realize that insert, delete, and update processing is more intensive than read processing. With insert, delete, and update activity, changes must be logged.



Tuning the integration node

- Run as a trusted application to improve communication with the queue manager if the integration node is configured with an IBM MQ queue manager
- Traces and trace nodes
  - Integration node trace can cause significant processor and memory usage
  - Disable trace node output and stops trace nodes from running without removing them from the message flow

```
mqsichangetrace -n
```

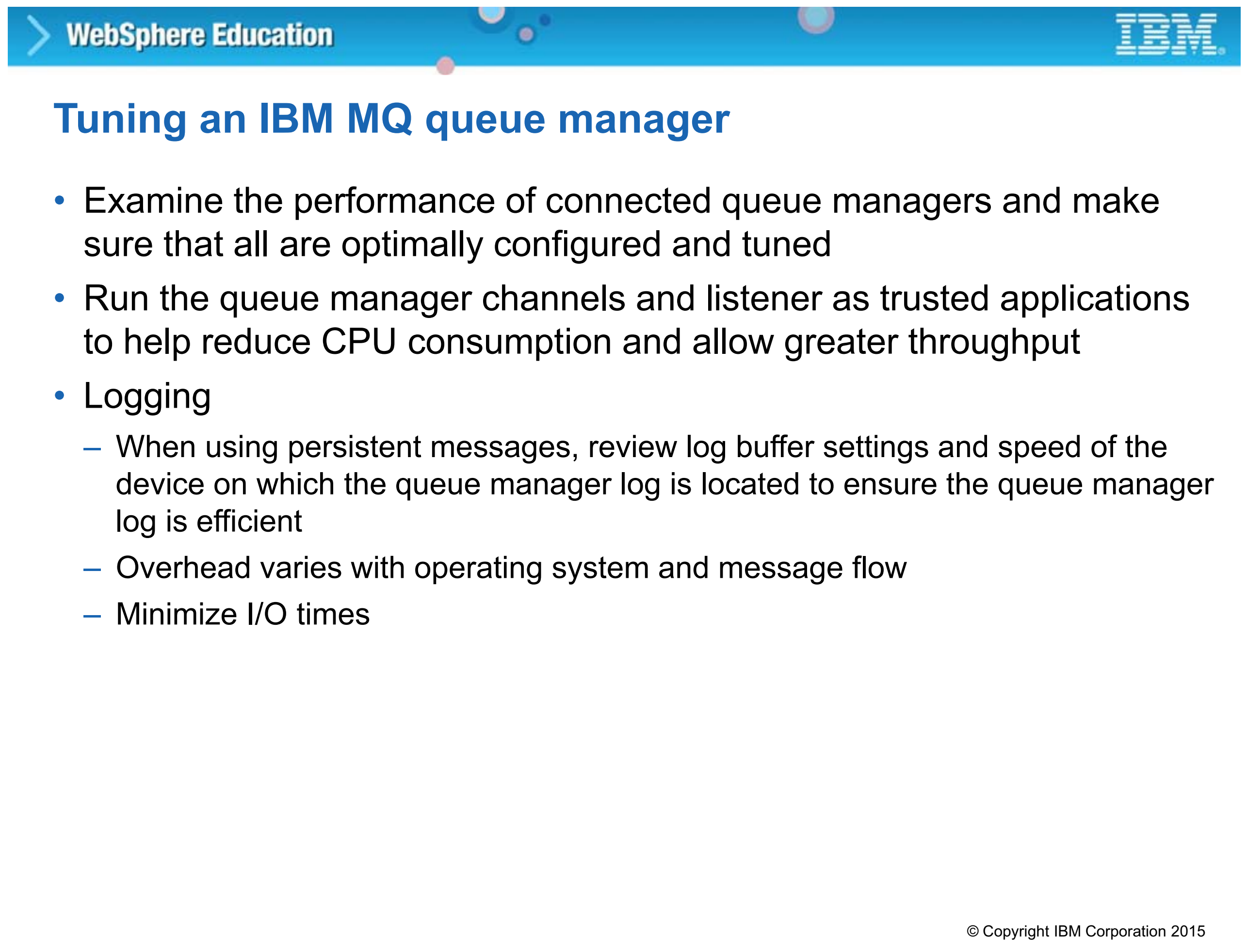
© Copyright IBM Corporation 2015

Figure 10-41. Tuning the integration node WM6461.0

Notes:

Turn all tracing off within the integration node before tuning the integration node environment, including user and service level trace.

Disable trace node output by using the `mqsichangetrace` command. With this command, you can suppress trace node output without deleting the trace nodes from the message flow. There is no performance impact when the trace nodes are deactivated.



Tuning an IBM MQ queue manager

- Examine the performance of connected queue managers and make sure that all are optimally configured and tuned
- Run the queue manager channels and listener as trusted applications to help reduce CPU consumption and allow greater throughput
- Logging
  - When using persistent messages, review log buffer settings and speed of the device on which the queue manager log is located to ensure the queue manager log is efficient
  - Overhead varies with operating system and message flow
  - Minimize I/O times

Figure 10-42. Tuning an IBM MQ queue manager WM6461.0

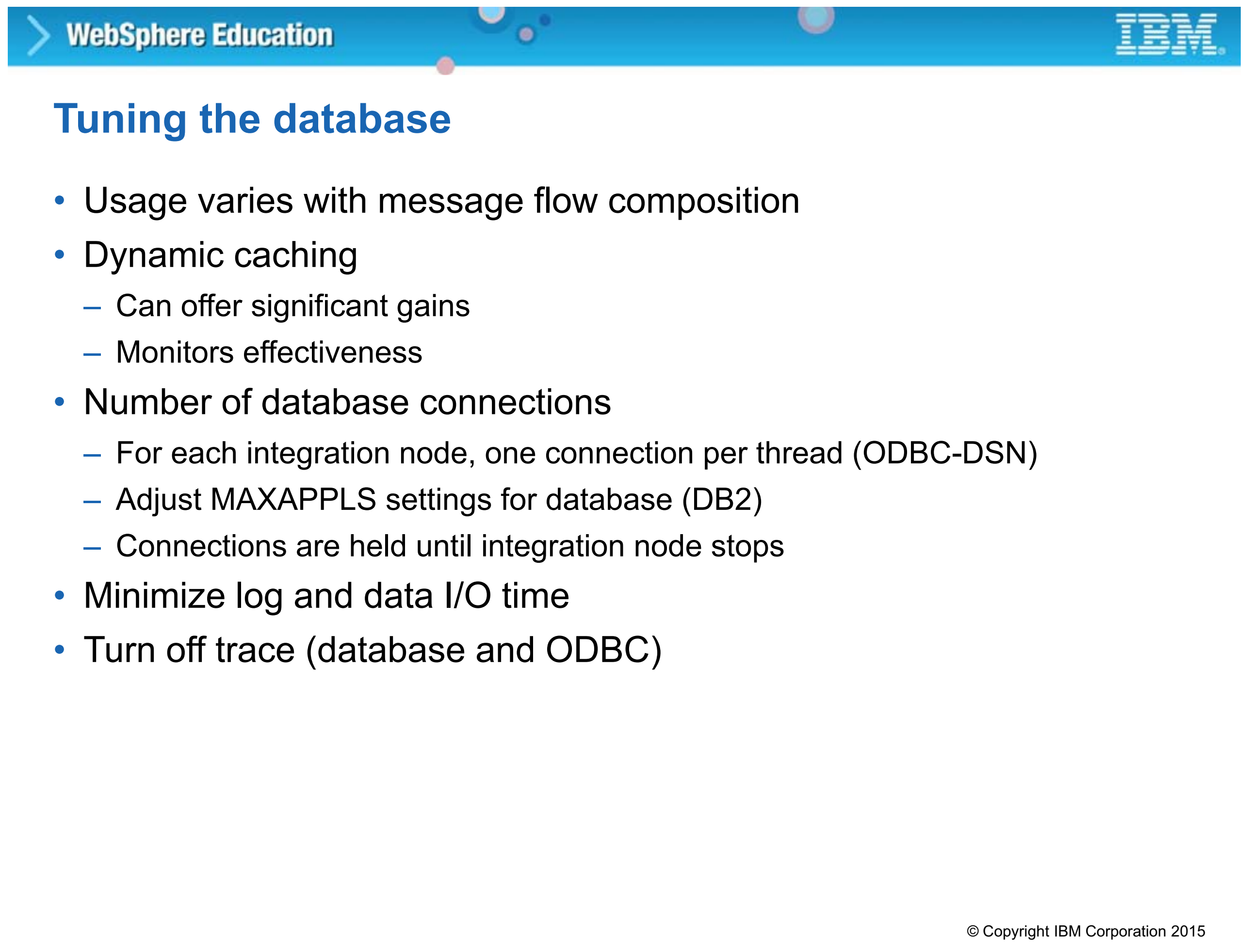
Notes:

Where possible, ensure that the integration node is located as close as possible to the incoming messages. If output messages must be moved to a remote queue manager, processing cost increases for those messages. This situation is true if the messages are persistent. With persistent messages, the performance of logging over multiple queue managers contributes to slower throughput. Each message must be logged multiple times as it passes from one queue manager to another.

When persistent messages are used, the queue manager must successfully generate a synchronous write at commit time to the log before a unit of work is complete. This restriction can significantly add to the amount of processor resources that are required for message processing. This situation becomes apparent when you compare the rates at which persistent and nonpersistent messages can be processed.

When tuning logging parameters, also consider changing those parameters that are related to log buffer and extent size. Large log extents mean less frequent switching between extents. The performance impact that logging causes also varies with operating system.

In a system in which queue managers are interconnected. It is important that both queue managers are configured for efficient logging. The log on both queue managers is used as the messages move across the IBM MQ channel that connects the two queue managers.



WebSphere Education

Tuning the database

- Usage varies with message flow composition
- Dynamic caching
  - Can offer significant gains
  - Monitors effectiveness
- Number of database connections
  - For each integration node, one connection per thread (ODBC-DSN)
  - Adjust MAXAPPLS settings for database (DB2)
  - Connections are held until integration node stops
- Minimize log and data I/O time
- Turn off trace (database and ODBC)

© Copyright IBM Corporation 2015

Figure 10-43. Tuning the database WM6461.0

Notes:

The extent to which a message flow accesses a database depends on the composition of the message flow. Obviously if there is no access, no tuning is required. The use of business data can vary from read-only activity through update-insert-delete. The type of processing that takes place determines where you must place your tuning effort.

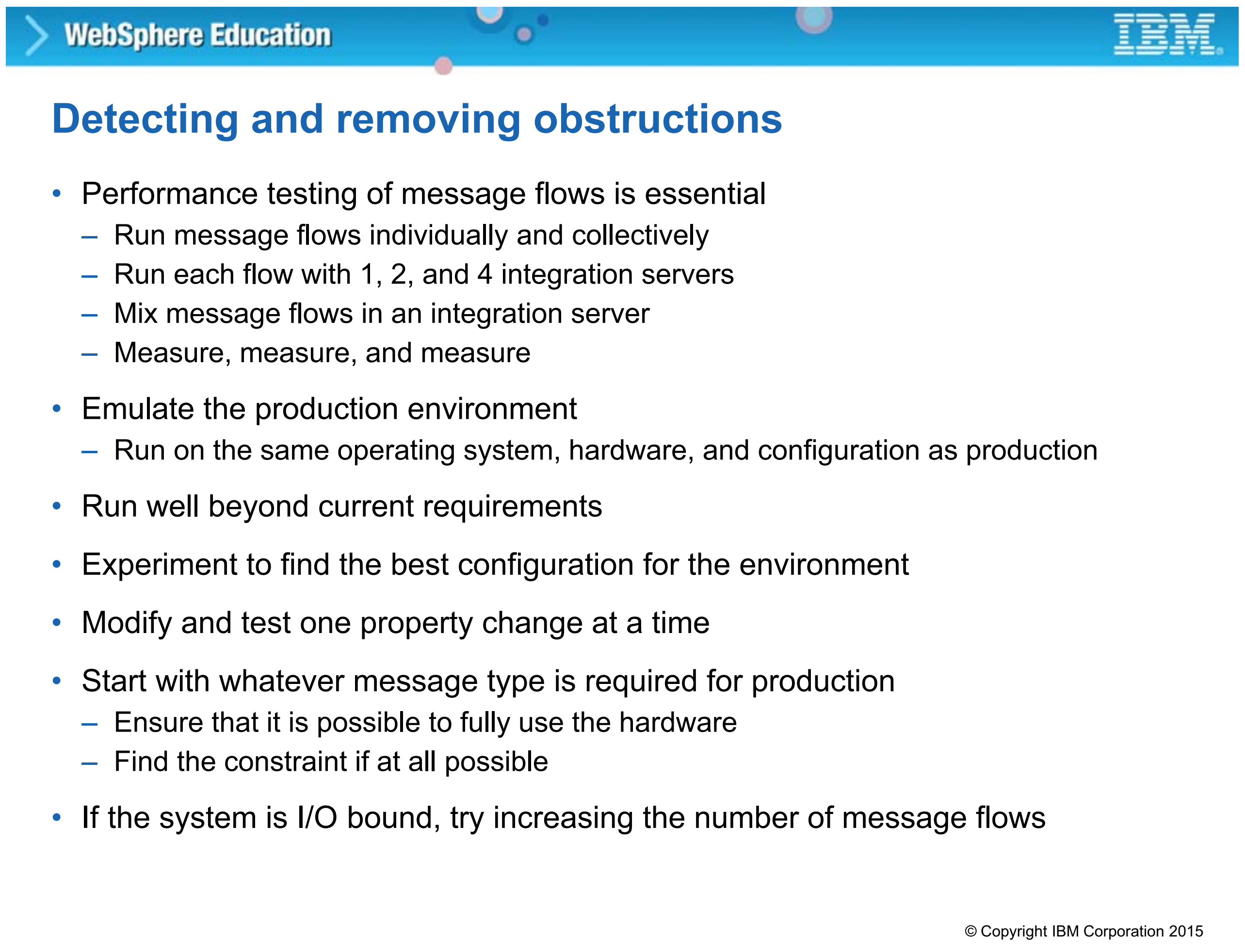
Ensure that both database and ODBC trace are turned off. ODBC tracing is controlled differently on each of the different operating systems.

Where message flows frequently access a database, you can obtain a significant performance benefit by caching the dynamic SQL, which is generated when an access error occurs. Without caching, each data access must be compiled with an `SQL PREPARE` statement and then run by using an `SQL EXECUTE` statement. The compilation is relatively expensive. The aim is to significantly reduce the number of times it takes place.

Ensure that each database is configured with sufficient connections. The connection from the message flow to the database manager is made on the first database request in the flow. Depending on the number of connections that are defined within the database manager and the number of instances of the message flow, not all requests can be connected, which obviously reduces the throughput potential.

To achieve maximum throughput, ensure that the database log is on a fast device.

A BLOB written to a database is an immediate write to disk. It is not buffered like other data. If BLOB I/O is frequent, it is beneficial to locate the data portion of the database on a fast device.



WebSphere Education

Detecting and removing obstructions

- Performance testing of message flows is essential
  - Run message flows individually and collectively
  - Run each flow with 1, 2, and 4 integration servers
  - Mix message flows in an integration server
  - Measure, measure, and measure
- Emulate the production environment
  - Run on the same operating system, hardware, and configuration as production
- Run well beyond current requirements
- Experiment to find the best configuration for the environment
- Modify and test one property change at a time
- Start with whatever message type is required for production
  - Ensure that it is possible to fully use the hardware
  - Find the constraint if at all possible
- If the system is I/O bound, try increasing the number of message flows

Figure 10-44. Detecting and removing obstructions WM6461.0

Notes:

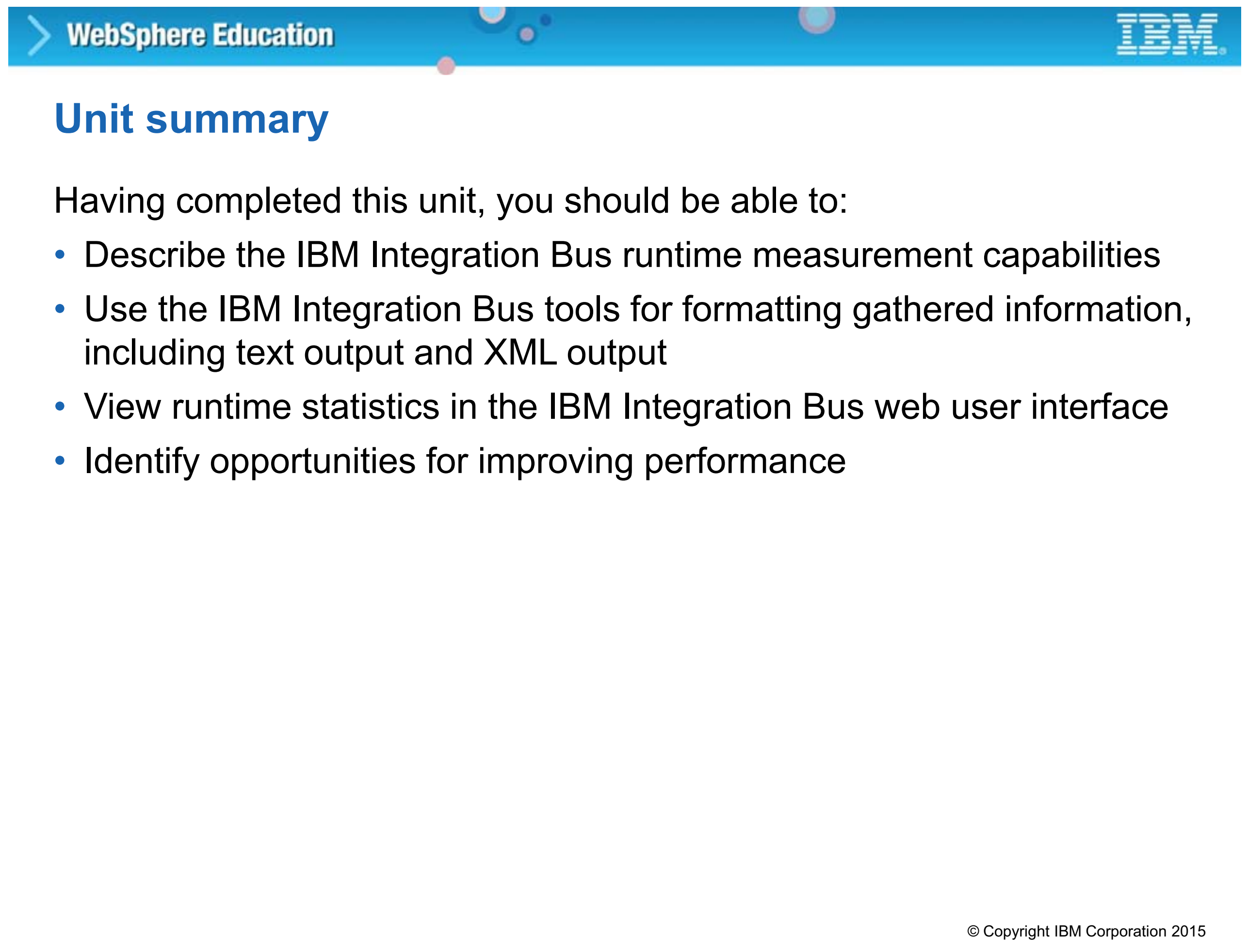
It is essential to undertake performance testing of message flows before production. You might observe that all is well when running few copies of a message flow but that things change when more copies are running. It is essential to assess this situation before moving to production.

First, determine the effects of running an increasing number of copies of each message flow. Run message flows in isolation, and in combination, to see whether there are any adverse effects.

Be sure to test this situation in the same environment as production, ideally by using the same data.

Test with nonpersistent messages at first, even if you intend to use persistent messages in production. Using nonpersistent messages removes the limit (I/O impact) that the use of persistent messages imposes. Such testing can help to identify a constraint, which would not otherwise be apparent.

In performance testing, it is important to reflect the production environment as fully as possible. In most cases, it is typical to have a continual flow of messages through the system, rather than processing messages in batches. What often happens in testing is that many messages are loaded onto a queue and then processed. This approach produces different results from a continual topping up of the input queues.



Unit summary

Having completed this unit, you should be able to:

- Describe the IBM Integration Bus runtime measurement capabilities
- Use the IBM Integration Bus tools for formatting gathered information, including text output and XML output
- View runtime statistics in the IBM Integration Bus web user interface
- Identify opportunities for improving performance

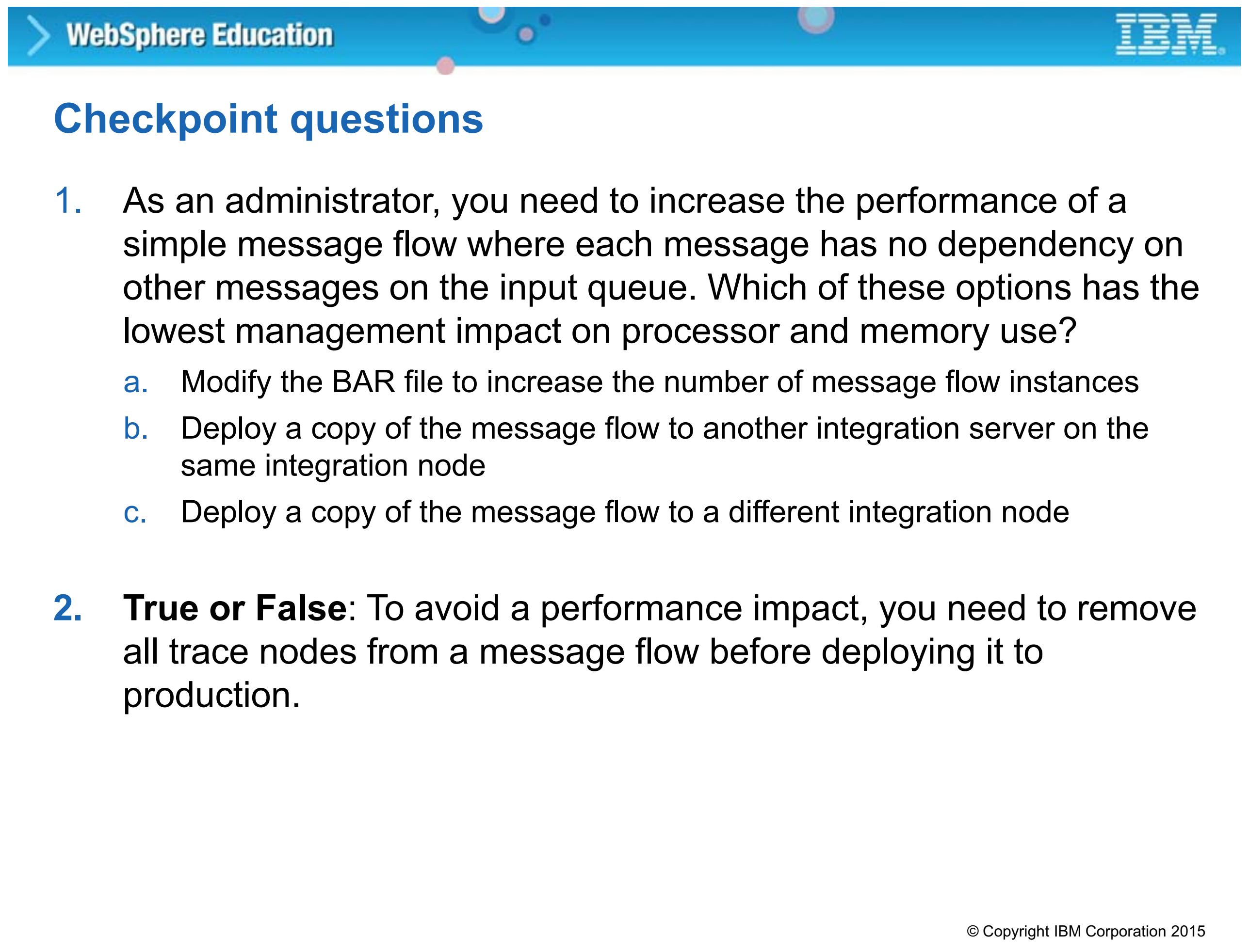
Figure 10-45. Unit summary WM6461.0

Notes:

There are multiple aspects to improving performance. It is rare to change one parameter and have a dramatic effect on message throughput. In practice, the large changes come from improvements in a number of areas. It is important to have a consistent and reliable view of message throughput and resource usage as you tune a system. Without consistency, you cannot be sure of the benefits from changing a property.

Tuning is an iterative process. Early changes tend to focus on the most significant metrics. With experience, later changes become less significant as do the returns. However, it is still worth addressing all areas to make the best possible improvement to your configuration.

Finally, you saw how it is possible to estimate message throughput for a message flow that is not yet developed, if you have a good understanding of the high-level requirements. Such estimates can be useful and can help to influence early design decisions when it becomes clear that a certain implementation choice is not going to meet objectives. This approach makes it possible to influence and improve design without coding the message flow.



WebSphere Education IBM.

Checkpoint questions

1. As an administrator, you need to increase the performance of a simple message flow where each message has no dependency on other messages on the input queue. Which of these options has the lowest management impact on processor and memory use?
   a. Modify the BAR file to increase the number of message flow instances
   b. Deploy a copy of the message flow to another integration server on the same integration node
   c. Deploy a copy of the message flow to a different integration node

2. **True or False**: To avoid a performance impact, you need to remove all trace nodes from a message flow before deploying it to production.

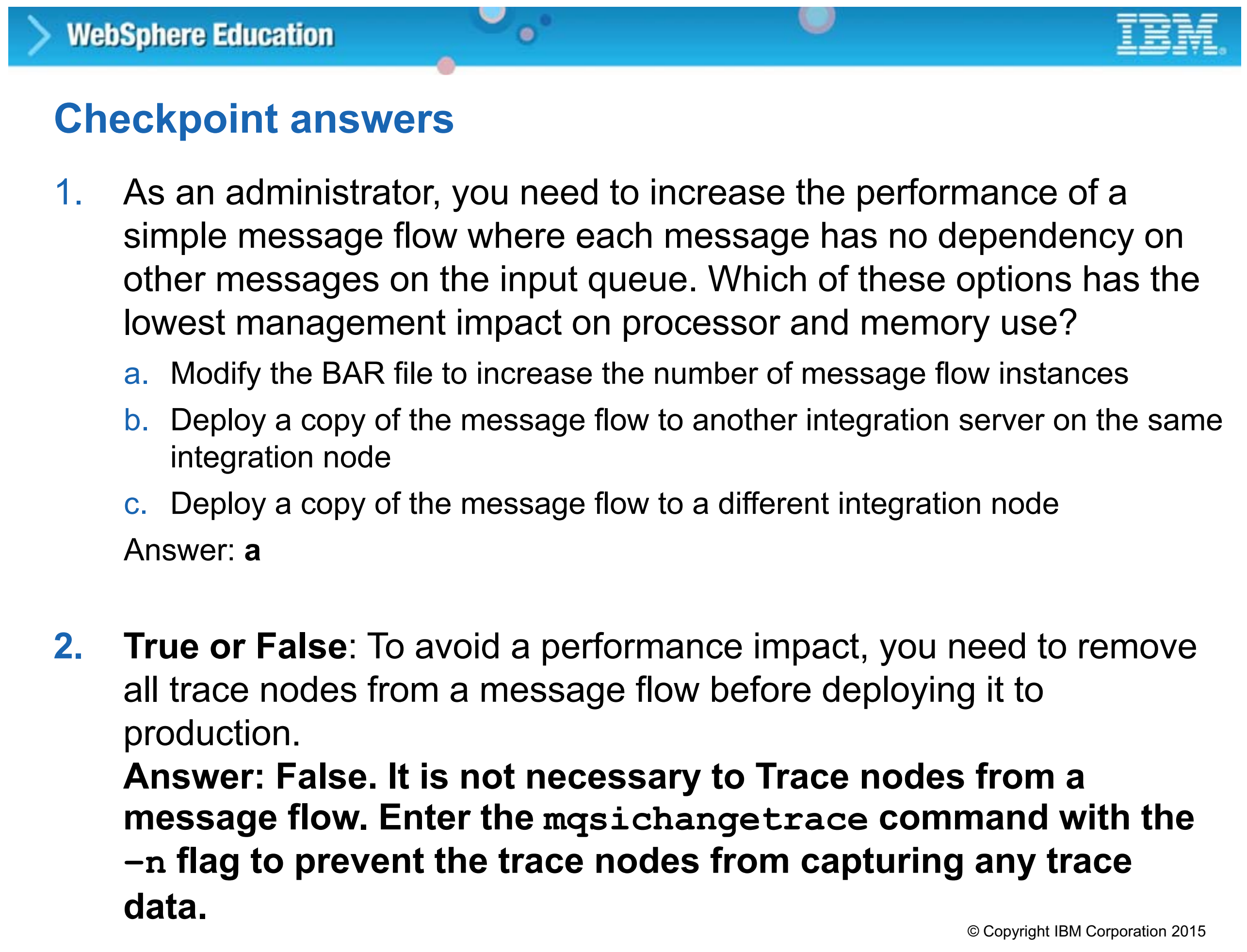
Figure 10-46. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education IBM

Checkpoint answers

1. As an administrator, you need to increase the performance of a simple message flow where each message has no dependency on other messages on the input queue. Which of these options has the lowest management impact on processor and memory use?
   a. Modify the BAR file to increase the number of message flow instances
   b. Deploy a copy of the message flow to another integration server on the same integration node
   c. Deploy a copy of the message flow to a different integration node

   Answer: **a**

2. **True or False**: To avoid a performance impact, you need to remove all trace nodes from a message flow before deploying it to production.
   Answer: False. It is not necessary to Trace nodes from a message flow. Enter the `mqsichangetrace` command with the `–n` flag to prevent the trace nodes from capturing any trace data.

Figure 10-47. Checkpoint answers WM6461.0

Notes:

Unit 11. Publish/subscribe implementation overview

What this unit is about

In this unit, you learn how to use IBM Integration Bus with IBM MQ and MQTT to implement publish/subscribe messaging.

What you should be able to do

After completing this unit, you should be able to:

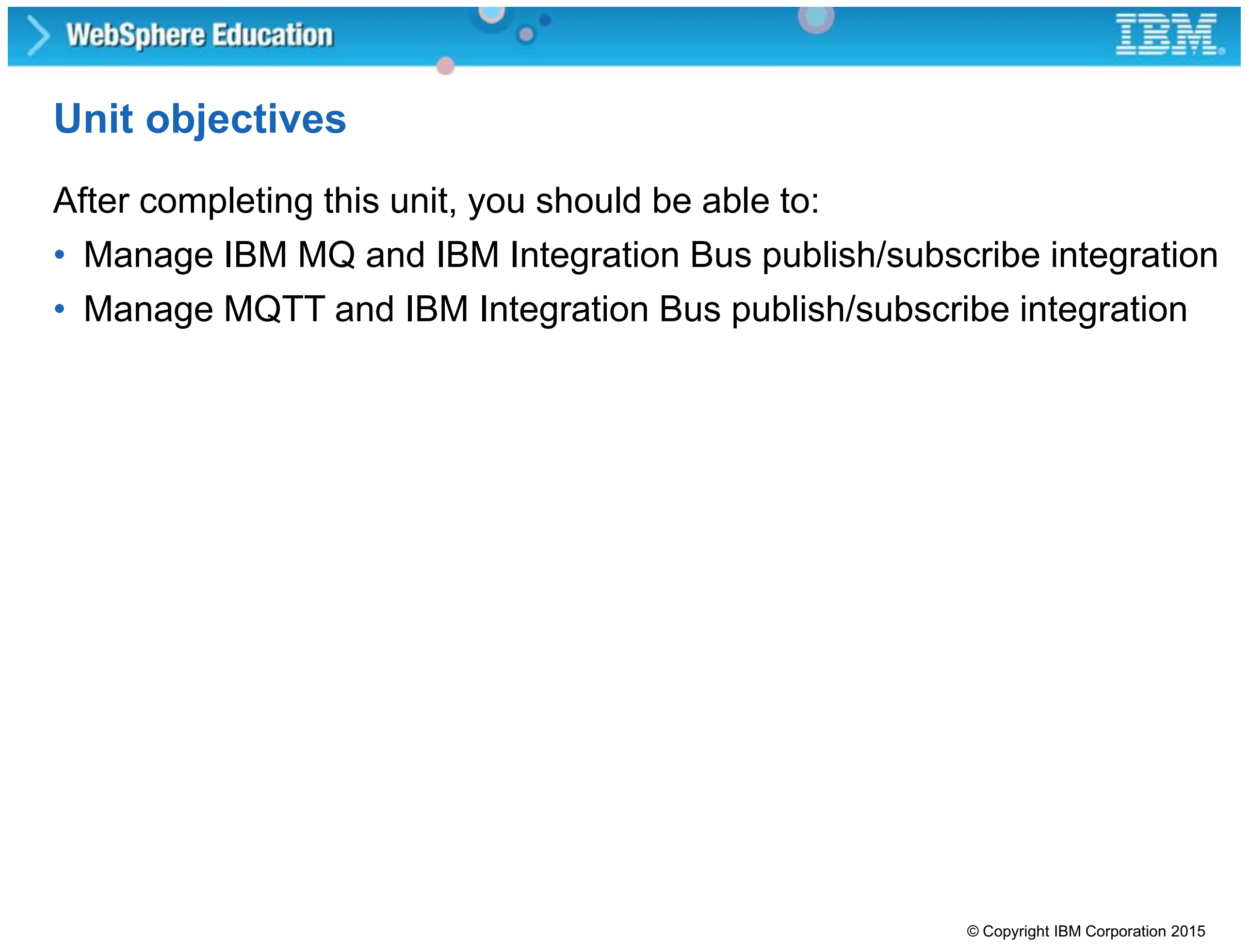
- Manage IBM MQ and IBM Integration Bus publish/subscribe integration
- Manage MQTT and IBM Integration Bus publish/subscribe integration

How you will check your progress

- Checkpoint questions
- Lab exercise

References

IBM Knowledge Center for IBM Integration Bus V10



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Manage IBM MQ and IBM Integration Bus publish/subscribe integration
- Manage MQTT and IBM Integration Bus publish/subscribe integration

© Copyright IBM Corporation 2015

Figure 11-1. Unit objectives WM6461.0

Notes:

11.1. Publish/subscribe overview

WebSphere Education

What is publish/subscribe?

- Provides independence between senders (publishers) and receivers (subscribers)
  - Applications can join or leave without affecting others or configuration actions
  - Administrator can set authorities
  - Administrative application can set subscriptions if required
  - Publish/subscribe broker filters messages based on subject or topic

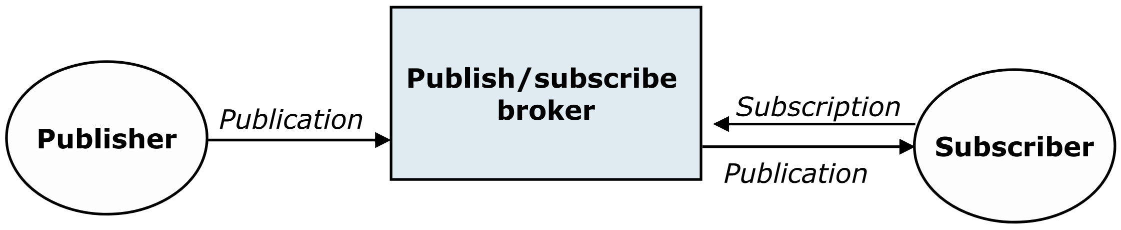


Figure 11-2. What is publish/subscribe? WM6461.0

Notes:

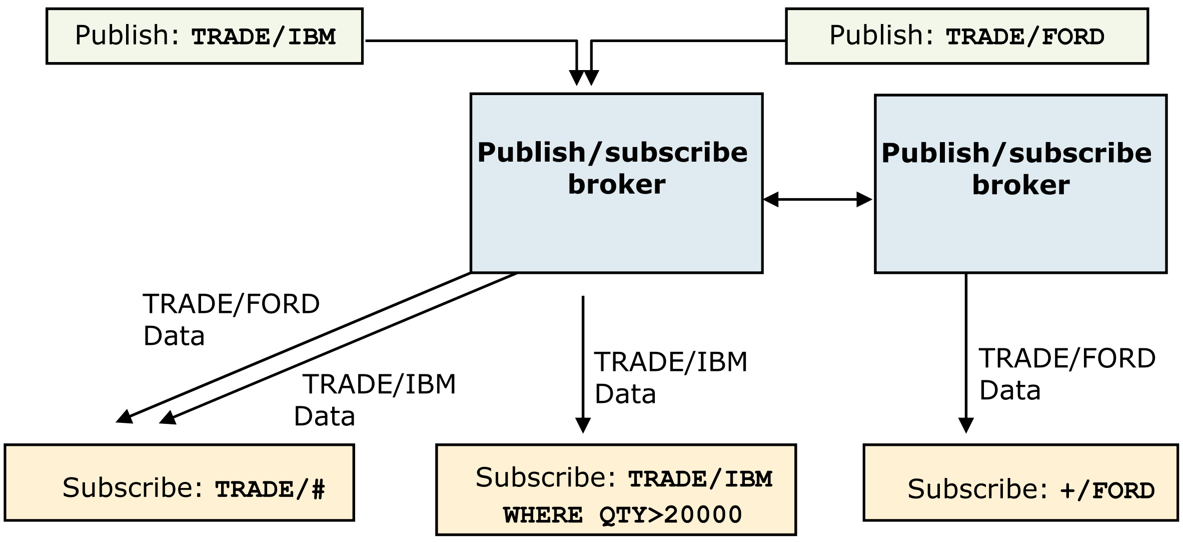
A publish/subscribe topology decouples the sender (publisher or producer) and the receiver (subscriber or consumer) of a message. Publishers create topics. Applications can subscribe and unsubscribe to a topic.

IBM MQ and MQTT servers support publish/subscribe networks. Integration Bus supplements this support by providing more transformation or routing, and filtering based on topic. A *filter* is an expression that is applied to the content of a publication message to determine whether it matches a subscription.

The advantage of a publish/subscribe system is that it can remove the unmanageable aspects of a point-to-point network and replace them with a simple network of a publisher, an integration node, and all the subscribers. New subscription clients or services can be added without any effect or interruption in the service to other users. This capability provides a superior means of providing streamlined and efficient integration and growth across an enterprise. When IBM MQ is used as the backbone for message delivery, all of the benefits and features of IBM MQ are inherited.

WebSphere Education

Topics and filters



- Topic identifies the subject matter of the publication
- Publisher can specify multiple topics for the same publication
- Subscribers can use wildcards

Figure 11-3. Topics and filters WM6461.0

Notes:

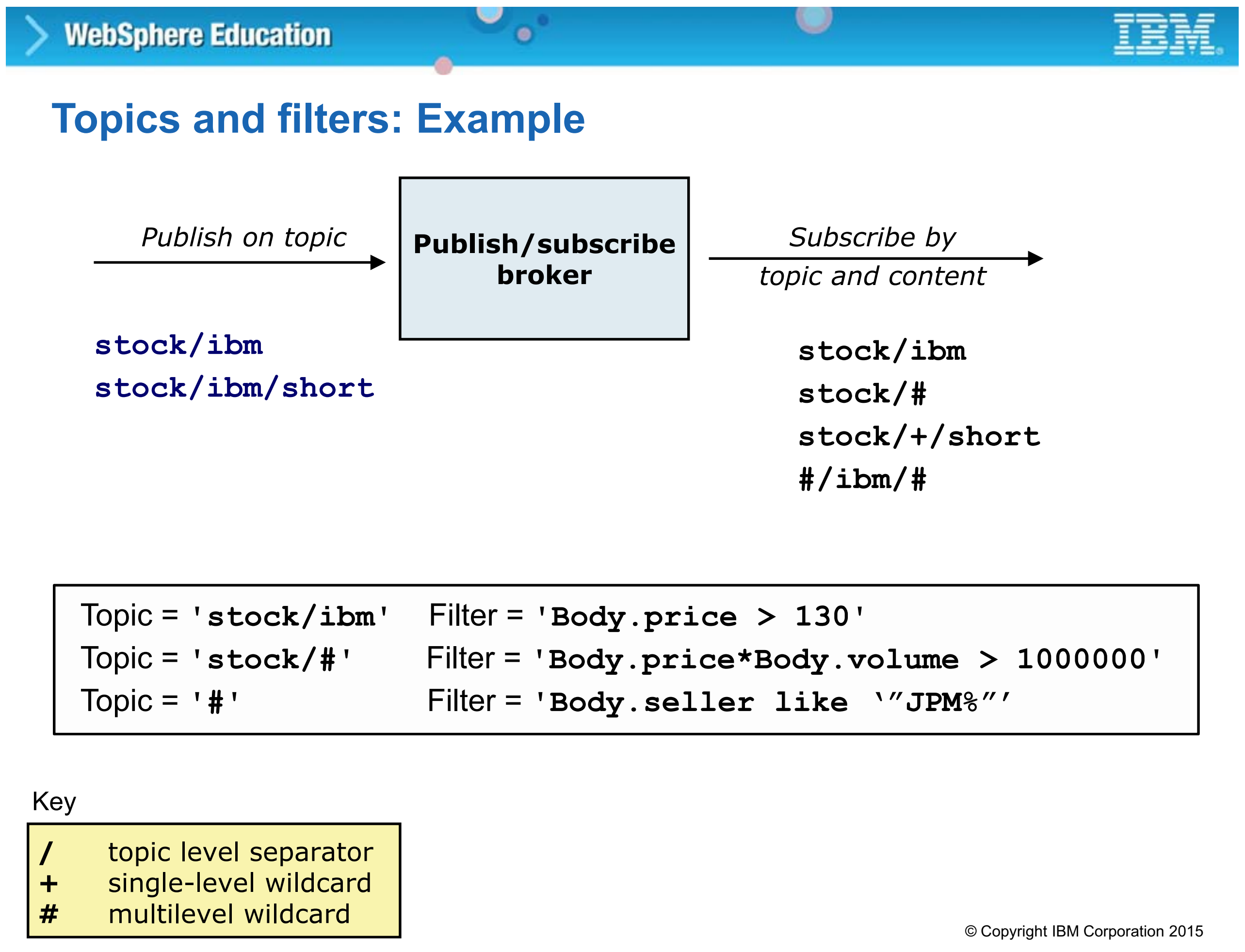
The topic identifier represents the subject matter of a publication, which the publisher specifies. The topic string can be any length. A topic tree structure can be built by qualifying the topic with the slash (/) character, which Integration Bus recognizes, but is not apparent to (base) IBM MQ publish/subscribe. Publishers can specify multiple topics for the same publication.

Subscribers subscribe to topics and can use wildcards:

+ matches with exactly one topic level (between / characters).

# matches with zero or more topic levels (at the beginning or end of a topic).

Subscribers can optionally include a content filter to further refine a request for matching publications. Wildcards are also allowed when specifying a content filter expression. In this figure, the “WHERE QTY > 20000” represents a content filter.



Topics and filters: Example

Publish on topic:

```
stock/ibm
stock/ibm/short
```

Subscribe by topic and content:

```
stock/ibm
stock/#
stock/+/short
#/ibm/#
```

Topic = `'stock/ibm'`	Filter = `'Body.price > 130'`
Topic = `'stock/#'`	Filter = `'Body.price*Body.volume > 1000000'`
Topic = `'#'`	Filter = `'Body.seller like ‘”JPM%”’`

Key

/	topic level separator
+	single-level wildcard
#	multilevel wildcard

Figure 11-4. Topics and filters: Example WM6461.0

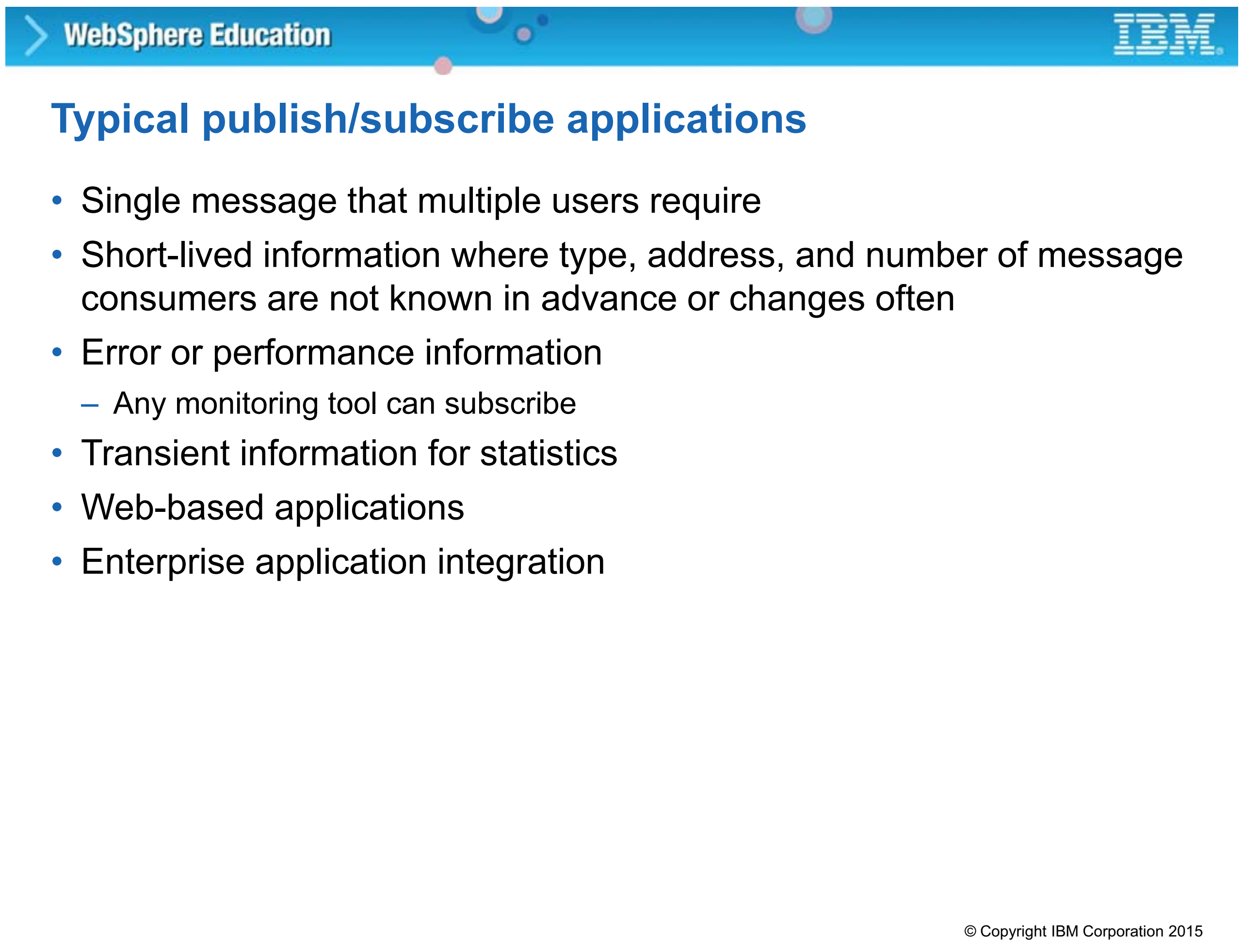
Notes:

Subscribers subscribe to topics and can use wildcard characters. A subscription must include a topic. A hashtag (#) denotes a generic topic. Filters are optional.

The figure shows examples of valid topic and filter strings by using topic level separators, single-level wildcards, and multilevel wildcards. The plus sign (+) matches with exactly one topic level, between forward slash (/) characters. The hashtag (#) matches with zero or more topic levels, at the beginning or end of the topic.

Note

Use only single quotation marks (‘) in filters.



WebSphere Education
IBM

Typical publish/subscribe applications

- Single message that multiple users require
- Short-lived information where type, address, and number of message consumers are not known in advance or changes often
- Error or performance information
  - Any monitoring tool can subscribe
- Transient information for statistics
- Web-based applications
- Enterprise application integration

© Copyright IBM Corporation 2015

Figure 11-5. Typical publish/subscribe applications WM6461.0

Notes:

Publish/subscribe messaging offers many benefits when applied to the correct environment. It is most useful when there are many dynamic receiving applications, that is, applications that subscribe to varying and changing topics. Also, no link is required between publishers and subscribers.

If there is a requirement to link applications, for example, to bill a subscriber according to the type and source of information, then it might be necessary to add more function to the publish/subscribe application. Doing so might reduce the benefit of low central administration (an inherent benefit of publish/subscribe).

If all the subscribers are known and remain relatively static and the type of information they receive is also a known quantity, it might be better to use point-to-point messaging. In this scenario, the integration node examines messages for content and routes it to the relevant destination. In this case, there would be no requirement to formulate topic strings and for the receiving applications subscribe to these topics. This situation is especially true if the information that the subscribers request (that is, the topic) does not change and they require the same information regularly.

WebSphere Education

Integration Bus and publish/subscribe support (1 of 2)

- MQTT or IBM MQ publish/subscribe brokers notify applications of significant events that occur in integration nodes
  - Choose the publish/subscribe broker based on processing requirements and your existing architecture
- Enhances publish/subscribe applications
  - Provides extra transformation or routing function, or both, at publication time
  - Filters messages based on the content of the body of the message

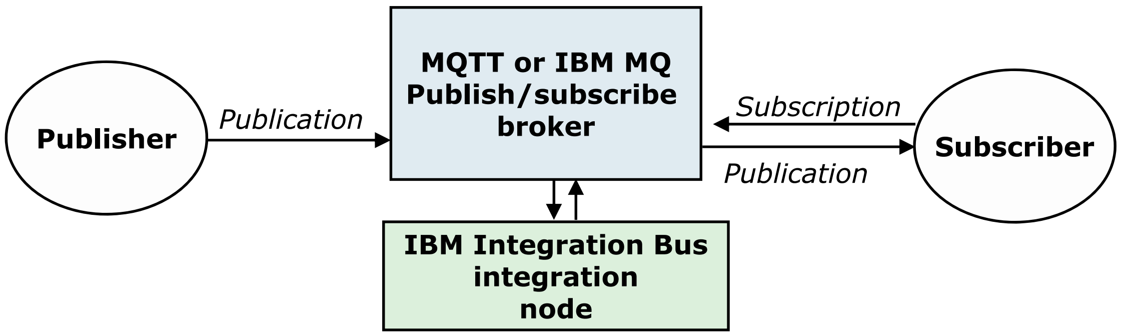

Publisher
Publication
MQTT or IBM MQ Publish/subscribe broker
Subscription
Publication
Subscriber
IBM Integration Bus integration node

Figure 11-6. Integration Bus and publish/subscribe support (1 of 2) WM6461.0

Notes:

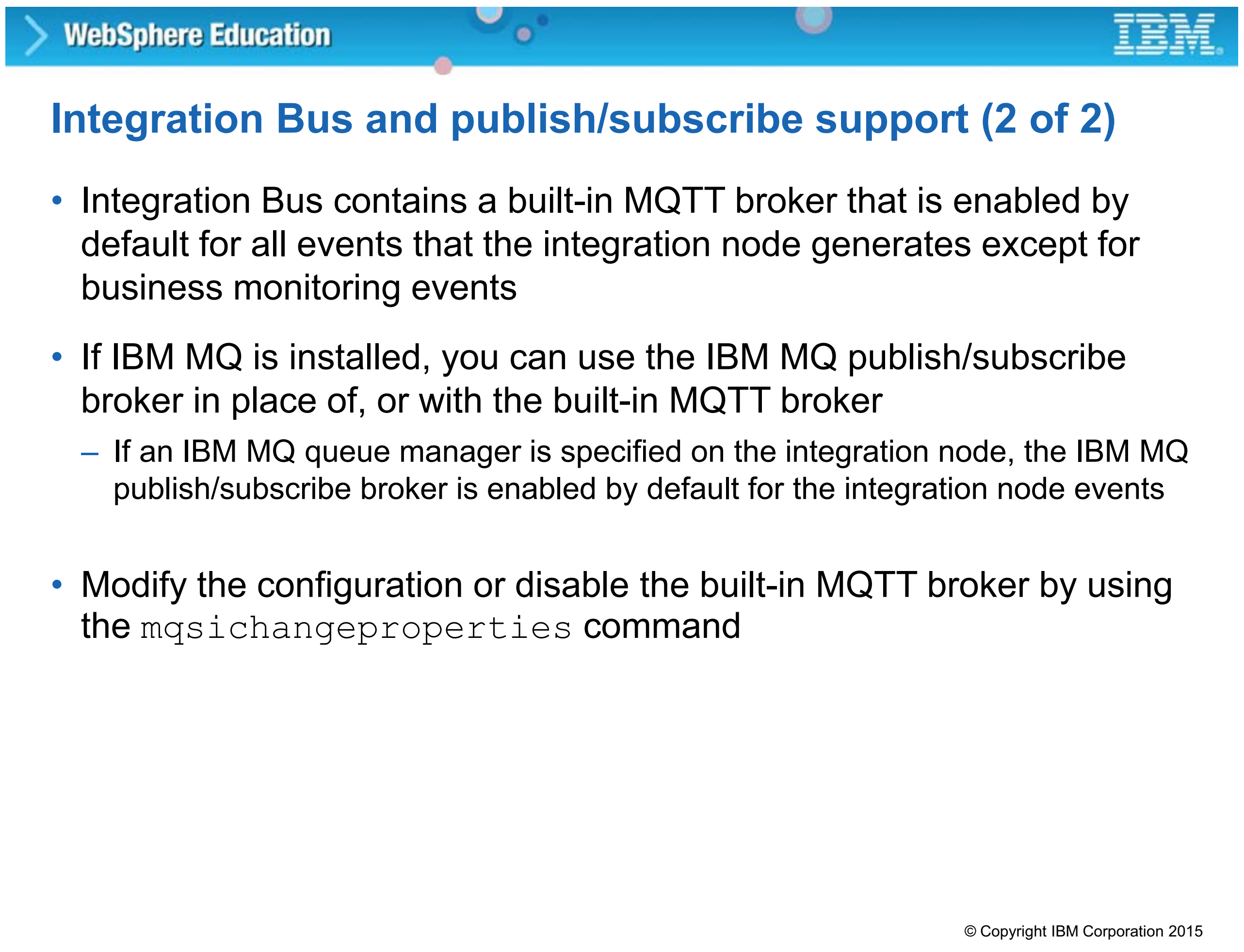
MQTT is a lightweight publish/subscribe messaging protocol. You can use IBM Integration Bus to connect to applications and devices that send and receive messages by using the MQTT messaging protocol.

Integration Bus and IBM MQ share a common publish/subscribe domain for topic and content-based operation. As an option, you can use IBM MQ for publish/subscribe with Integration Bus.

There are two situations where you can use publish/subscribe in Integration Bus:

- To provide more transformation or routing function, or both, at publication time.
- To filter messages based on the content of the body of the message.

As shown in the figure, messages are supplied to the message flow by applications that publish messages. Messages are retrieved from the message flow by applications that register a subscription with an integration node. A *subscription* defines the interest that a subscriber has in published messages.



WebSphere Education IBM

Integration Bus and publish/subscribe support (2 of 2)

- Integration Bus contains a built-in MQTT broker that is enabled by default for all events that the integration node generates except for business monitoring events
- If IBM MQ is installed, you can use the IBM MQ publish/subscribe broker in place of, or with the built-in MQTT broker
  - If an IBM MQ queue manager is specified on the integration node, the IBM MQ publish/subscribe broker is enabled by default for the integration node events
- Modify the configuration or disable the built-in MQTT broker by using the `mqsichangeproperties` command

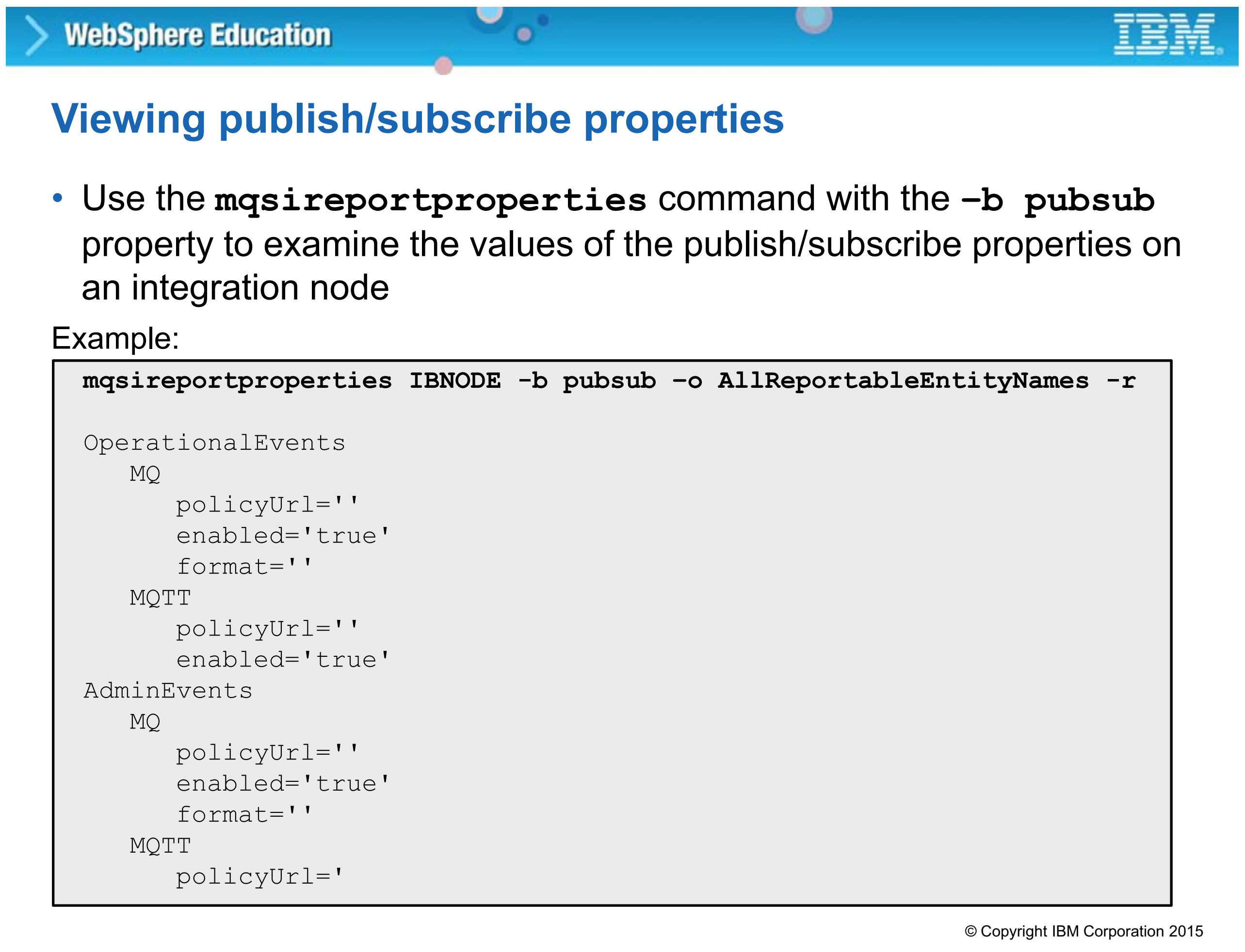
© Copyright IBM Corporation 2015

Figure 11-7. Integration Bus and publish/subscribe support (2 of 2) WM6461.0

Notes:

Integration Bus contains a built-in MQTT broker, which is enabled by default. The MQTT broker is the default transport for the publication of operational and administrative events by an integration node unless IBM MQ is installed and a queue manager is specified on the integration node.

You can configure your message flow to emit event messages that can be used to support transaction monitoring and auditing, and business process monitoring. These events messages are also know as *business events*. By default, the MQTT broker does not publish business events because they might contain confidential information and require more security than MQTT provides.



WebSphere Education IBM

Viewing publish/subscribe properties

- Use the **`mqsireportproperties`** command with the **`–b pubsub`** property to examine the values of the publish/subscribe properties on an integration node

Example:

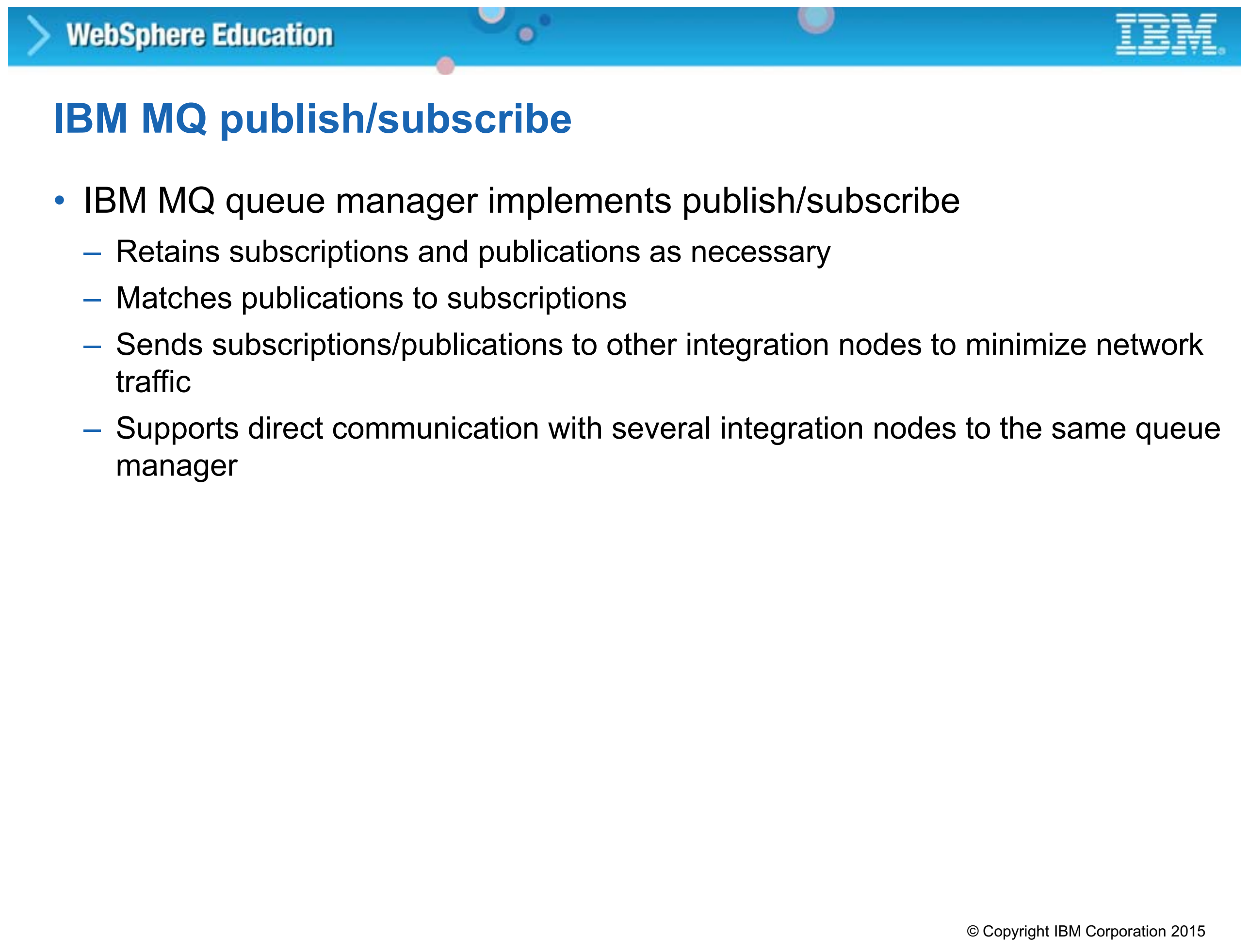
```
mqsireportproperties IBNODE -b pubsub –o AllReportableEntityNames -r

OperationalEvents
   MQ
      policyUrl=''
      enabled='true'
      format=''
   MQTT
      policyUrl=''
      enabled='true'
AdminEvents
   MQ
      policyUrl=''
      enabled='true'
      format=''
   MQTT
      policyUrl='
```

Figure 11-8. Viewing publish/subscribe properties WM6461.0

Notes:

You can use the `mqsireportproperties` command with the `-b` pubsub option to determine whether the IBM MQ or the built-in MQTT publish/subscribe broker is enabled for an integration node.

11.2. Publish/subscribe with IBM MQ



WebSphere Education

IBM MQ publish/subscribe

- IBM MQ queue manager implements publish/subscribe
  - Retains subscriptions and publications as necessary
  - Matches publications to subscriptions
  - Sends subscriptions/publications to other integration nodes to minimize network traffic
  - Supports direct communication with several integration nodes to the same queue manager

© Copyright IBM Corporation 2015

Figure 11-9. IBM MQ publish/subscribe WM6461.0

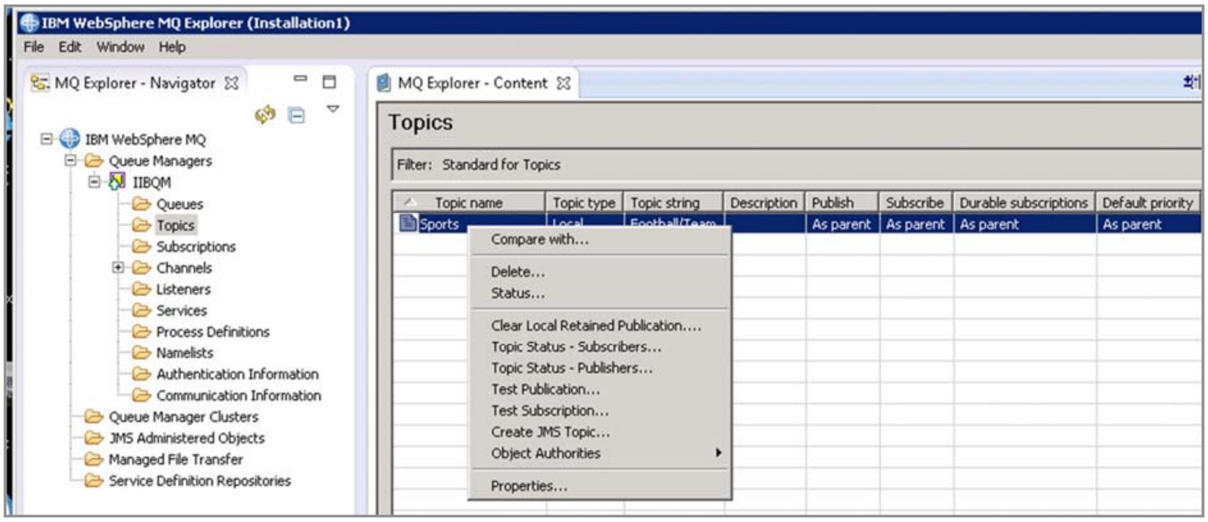
Notes:

You can use IBM MQ to support publish/subscribe applications.

The figure lists some of the publish/subscribe tasks that can be done with IBM MQ.

WebSphere Education

Managing topics with IBM MQ Explorer



Topic name	Topic type	Topic string	Description	Publish	Subscribe	Durable subscriptions	Default priority
Sports	Local	Football/Team		As parent	As parent	As parent	As parent

- **Topics** view lists the properties of all the topics on the selected queue manager
  - Create topics
  - Check topic status
  - Test publications

Figure 11-10. Managing topics with IBM MQ Explorer WM6461.0

Notes:

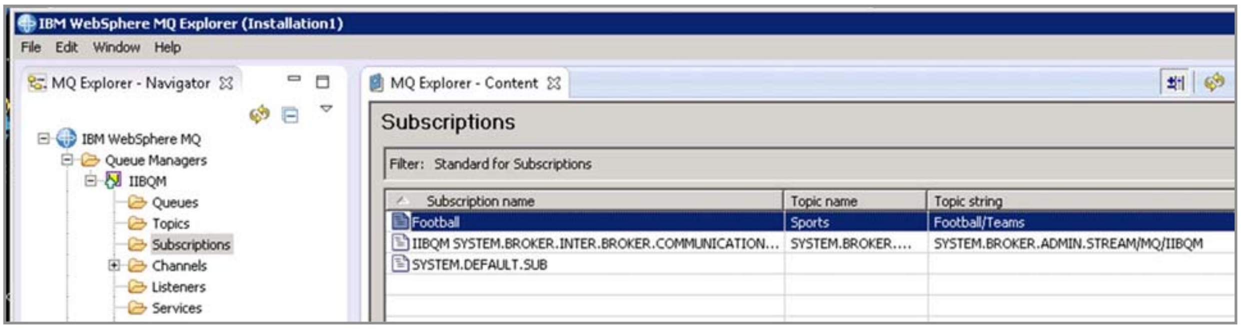
You can use IBM MQ Explorer or IBM MQ commands to manage publication topics.

To access the topics on a queue manager, click the **Topics** folder under the queue manager name in the IBM MQ Explorer Navigator.

From the **Topics** view, you can create topics and check topic status. You can also test publications.

WebSphere Education

Managing subscriptions with IBM MQ Explorer



Subscription name	Topic name	Topic string
Football	Sports	Football/Teams
IIBQM SYSTEM.BROKER.INTER.BROKER.COMMUNICATION...	SYSTEM.BROKER....	SYSTEM.BROKER.ADMIN.STREAM/MQ/IIBQM
SYSTEM.DEFAULT.SUB		

- **Subscriptions** view lists the properties of all the subscriptions on the selected queue manager
  – Create subscriptions
  – Check subscription status
  – Test subscriptions

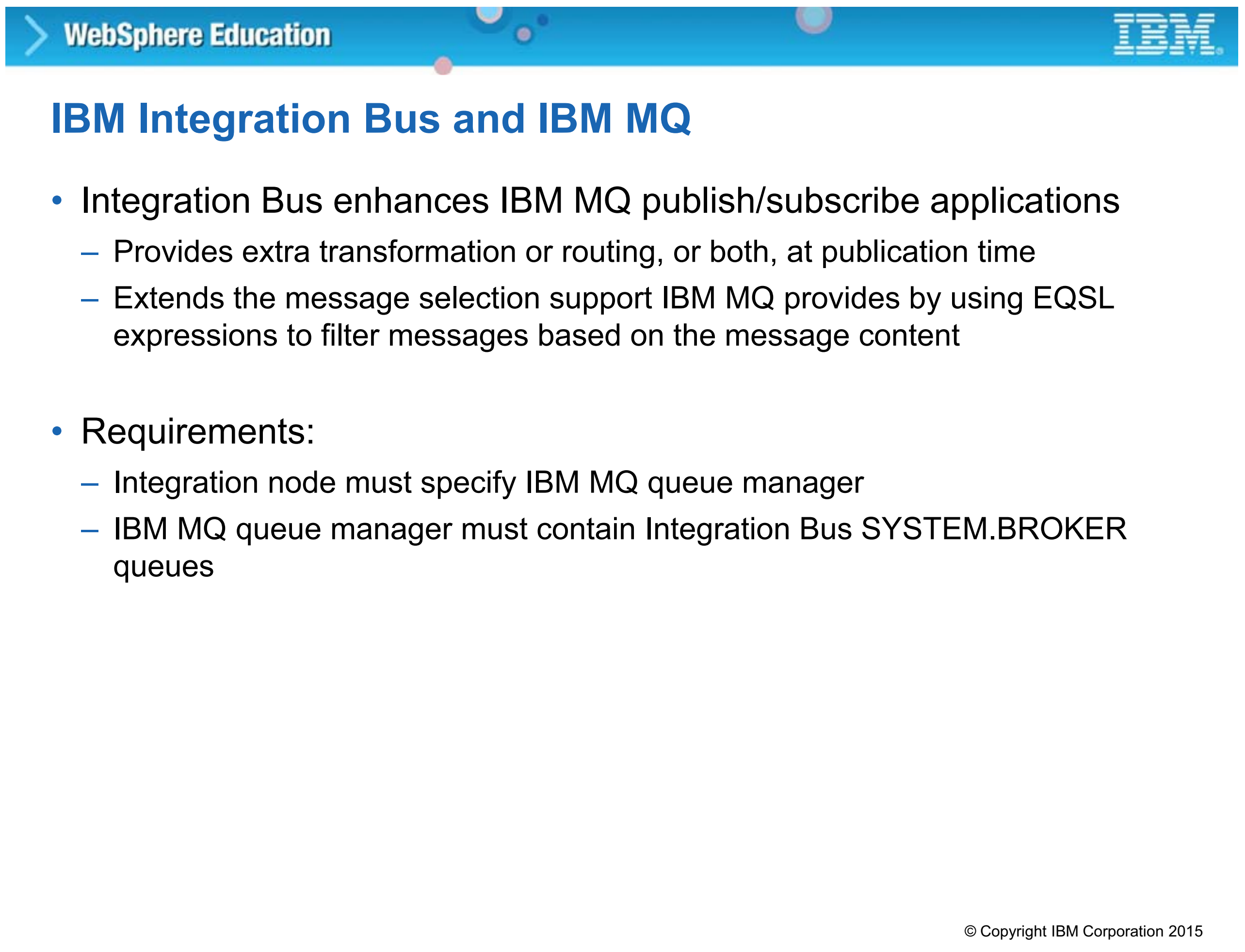
Figure 11-11. Managing subscriptions with IBM MQ Explorer WM6461.0

Notes:

You can use IBM MQ Explorer or IBM MQ commands to manage subscriptions.

To access the subscriptions on a queue manager, click the **Subscriptions** folder under the queue manager name in the IBM MQ Explorer Navigator.

From the **Subscriptions** view, you can create subscriptions and check subscription status. You can also test subscriptions.



WebSphere Education

IBM Integration Bus and IBM MQ

- Integration Bus enhances IBM MQ publish/subscribe applications
  - Provides extra transformation or routing, or both, at publication time
  - Extends the message selection support IBM MQ provides by using EQSL expressions to filter messages based on the message content
- Requirements:
  - Integration node must specify IBM MQ queue manager
  - IBM MQ queue manager must contain Integration Bus SYSTEM.BROKER queues

© Copyright IBM Corporation 2015

Figure 11-12. IBM Integration Bus and IBM MQ WM6461.0

Notes:

IBM MQ can specify a filter when a subscription is made, but the filter can refer to items in headers only. Integration Bus enhances IBM MQ publish/subscribe applications by acting as a content filter provider for IBM MQ. Integration Bus allows subscriptions to specify extended filters to that can refer to elements in the body of publications.

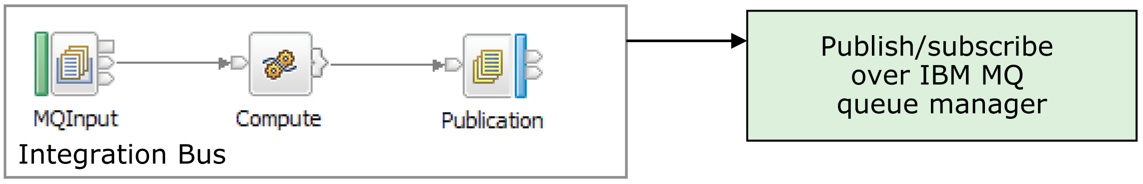
Integration Bus can also be used to provide extra transformation and routing for publications.

The requirements for IBM MQ content filtering are listed in the figure.

IBM MQ publish/subscribe is taught in course WM212, *IBM MQ V8 Advanced System Administration (Distributed)*.

WebSphere Education

Transformation and routing at publication



- Publication node
  - Interfaces with IBM MQ publish/subscribe broker
  - Filters output messages from a message flow and transmits them through an IBM MQ publish/subscribe broker to subscribers

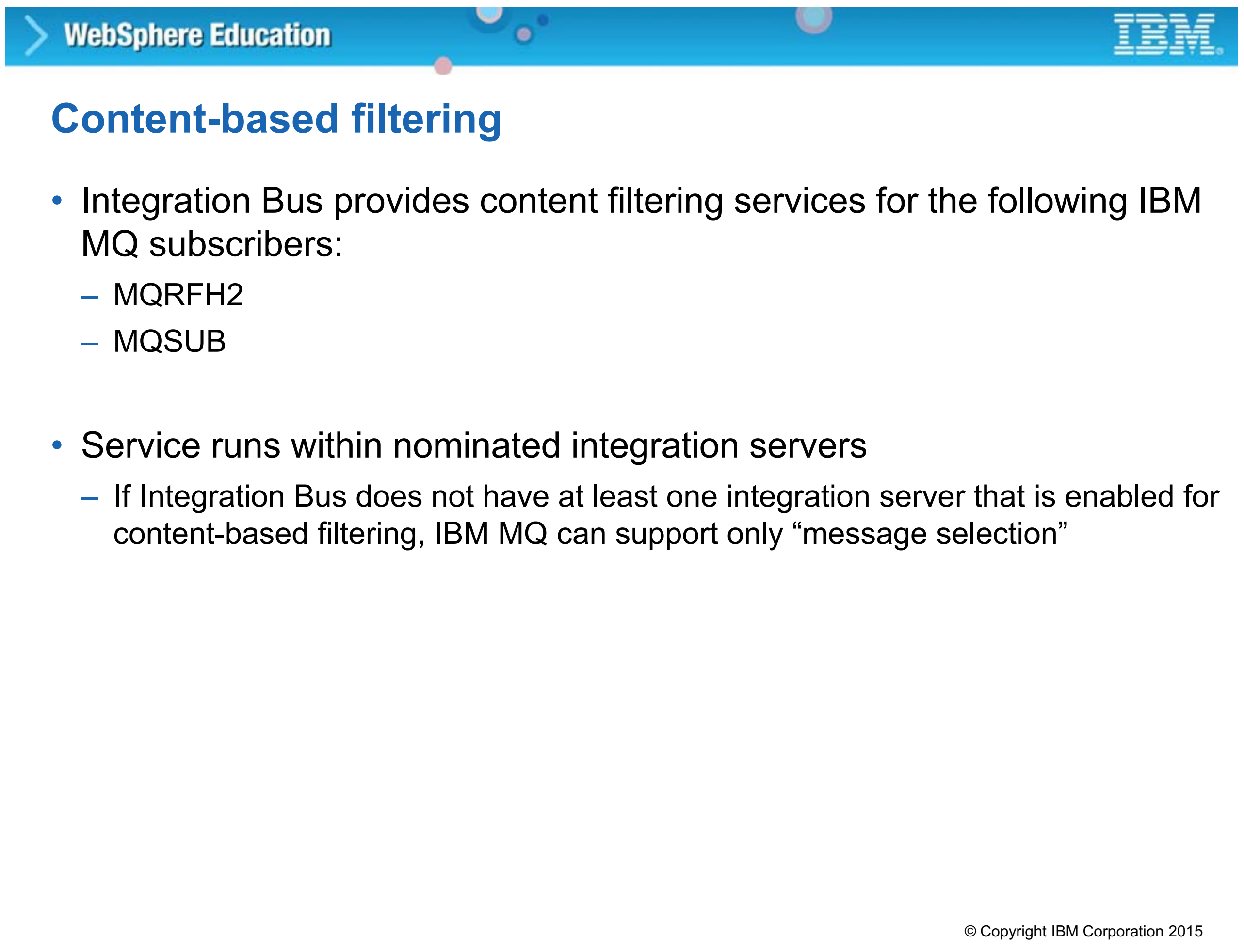
Figure 11-13. Transformation and routing at publication WM6461.0

Notes:

Integration Bus can also provide extra transformation and routing for publications by using the Publication node in a message flow.

Integration Bus parses the message that is propagated to the Publication node and calls the IBM MQ API so that IBM MQ can match and deliver to subscribers at the correct subscription level.

The Publication node uses the topic, or topics, and any options present in the command message to publish the message. The IBM MQ publish/subscribe broker delivers the publication to all subscribing applications that match the topic, and any other options that are specified on their subscriptions.



Content-based filtering

- Integration Bus provides content filtering services for the following IBM MQ subscribers:
  – MQRFH2
  – MQSUB
- Service runs within nominated integration servers
  – If Integration Bus does not have at least one integration server that is enabled for content-based filtering, IBM MQ can support only “message selection”

© Copyright IBM Corporation 2015

Figure 11-14. Content-based filtering WM6461.0

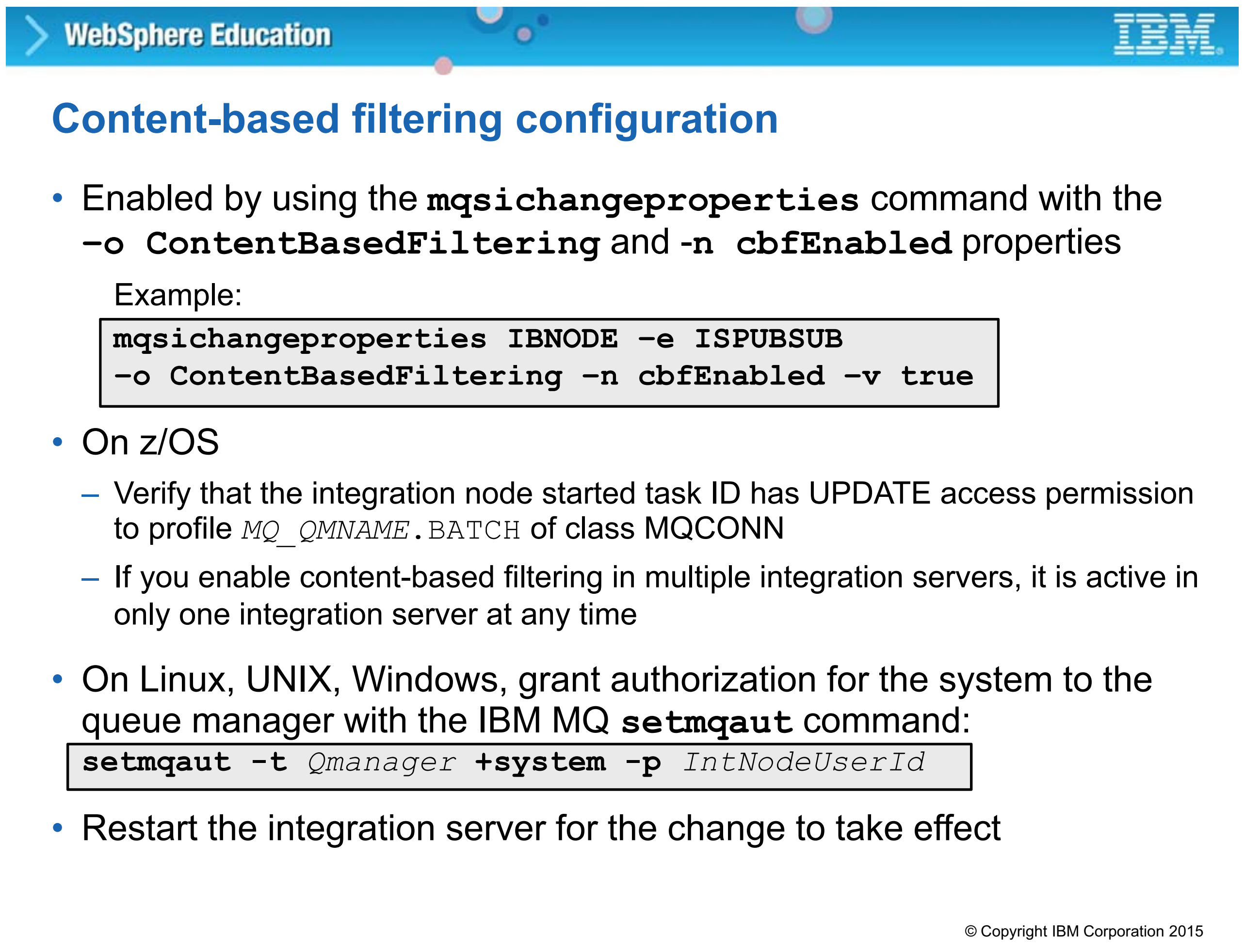
Notes:

IBM Integration Bus enables IBM MQ to support content-based filtering. Integration Bus provides content filtering services for the following IBM MQ subscribers:

- MQRFH2
- MQSUB

Content based filters are specified as ESQL expressions. An example of filter expression is `Body.Person.Salary>10000`.

You configure an integration server to provide content-based filtering by using the `mqsichangeproperties` command.



WebSphere Education IBM

Content-based filtering configuration

- Enabled by using the **`mqsichangeproperties`** command with the **`-o ContentBasedFiltering`** and **`-n cbfEnabled`** properties

  Example:

  ```
  mqsichangeproperties IBNODE -e ISPUBSUB
  -o ContentBasedFiltering -n cbfEnabled -v true
  ```

- On z/OS
  - Verify that the integration node started task ID has UPDATE access permission to profile `MQ_QMNAME.BATCH` of class MQCONN
  - If you enable content-based filtering in multiple integration servers, it is active in only one integration server at any time
- On Linux, UNIX, Windows, grant authorization for the system to the queue manager with the IBM MQ **`setmqaut`** command:

  ```
  setmqaut -t Qmanager +system -p IntNodeUserId
  ```

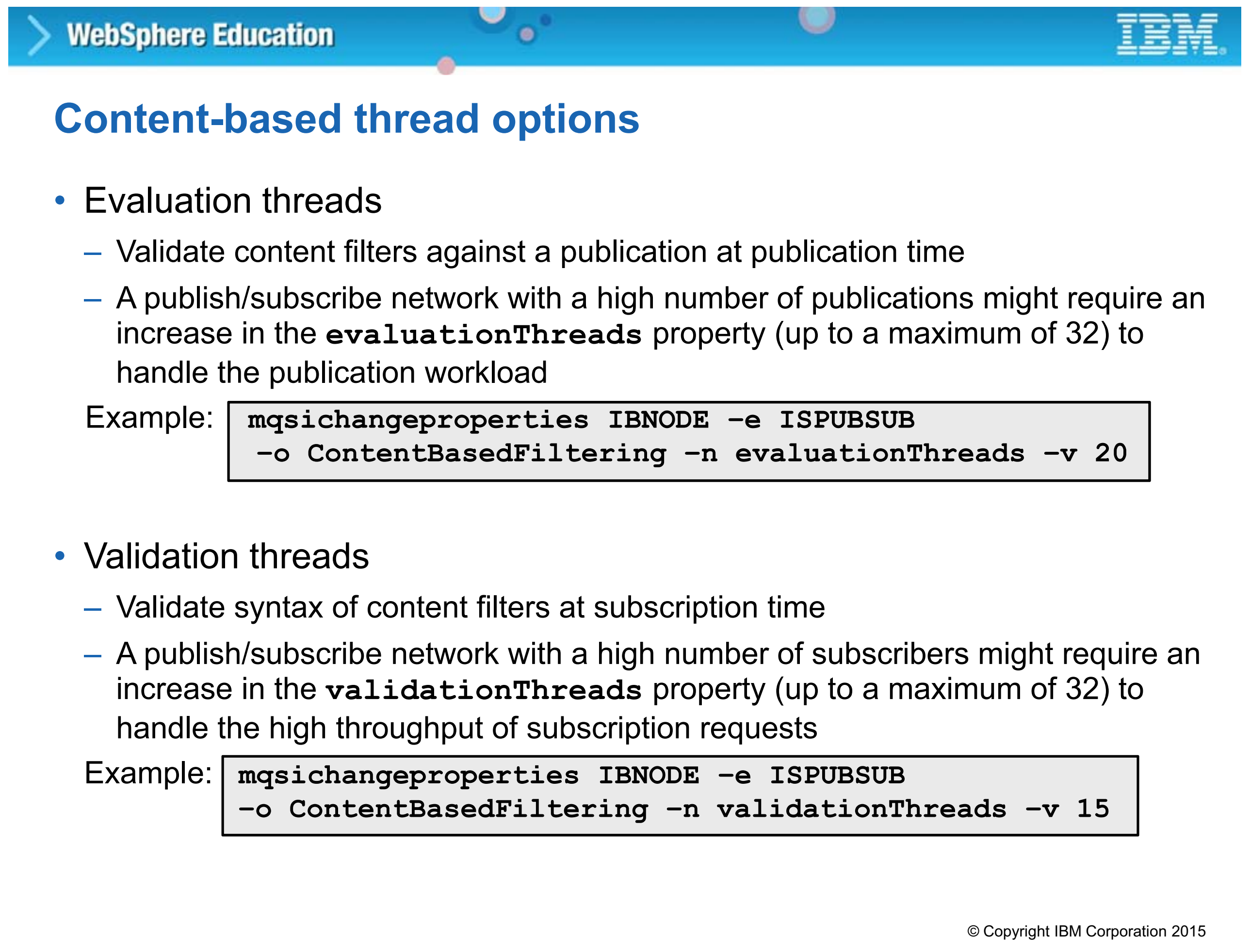
- Restart the integration server for the change to take effect

Figure 11-15. Content-based filtering configuration WM6461.0

Notes:

You enable content-based filtering on an integration server by using the `mqsichangeproperties` command with the `-o ContentBasedFiltering` option, as shown in the figure.

On distributed systems, you must grant authorization to the queue manager by using the IBM MQ `setmqaut` command.



Content-based thread options

- Evaluation threads
  - Validate content filters against a publication at publication time
  - A publish/subscribe network with a high number of publications might require an increase in the **`evaluationThreads`** property (up to a maximum of 32) to handle the publication workload

Example:

```
mqsichangeproperties IBNODE -e ISPUBSUB
 -o ContentBasedFiltering -n evaluationThreads -v 20
```

- Validation threads
  - Validate syntax of content filters at subscription time
  - A publish/subscribe network with a high number of subscribers might require an increase in the **`validationThreads`** property (up to a maximum of 32) to handle the high throughput of subscription requests

Example:

```
mqsichangeproperties IBNODE -e ISPUBSUB
-o ContentBasedFiltering -n validationThreads -v 15
```

Figure 11-16. Content-based thread options WM6461.0

Notes:

Content-based filtering supports two types of threads.

Evaluation threads validate content filters against a publication at publication time. A network with a high number of publications might require an increase in the `evaluationThreads` property (up to a maximum of 32) to handle the workload at publication time.

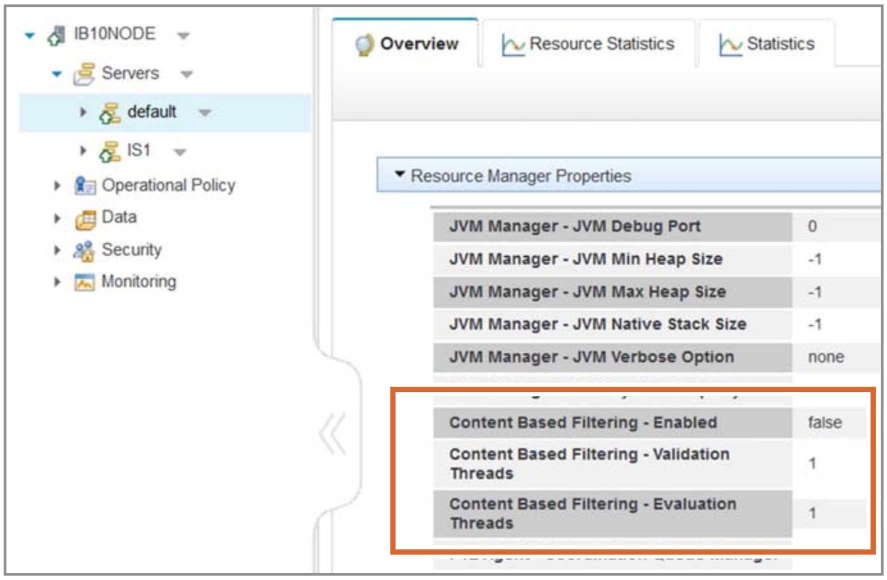
Validation threads validate the syntax of content filters at subscription time. A publish/subscribe network with a high number of subscribers, especially dynamic subscribers, might require an increase in the `validationThreads` property (up to a maximum of 32) to handle the high throughput of subscription requests.

Both the `evaluationThreads` and `validationThreads` properties default to a value of `1` if content-based filtering is enabled.

WebSphere Education IBM

Configure content-based filtering in Integration web interface

- Current settings are shown in the integration server **Resource Manager Properties**
- Click **Edit** to change the values
- Restart the integration server for the changes to take effect



JVM Manager - JVM Debug Port	0
JVM Manager - JVM Min Heap Size	-1
JVM Manager - JVM Max Heap Size	-1
JVM Manager - JVM Native Stack Size	-1
JVM Manager - JVM Verbose Option	none
Content Based Filtering - Enabled	false
Content Based Filtering - Validation Threads	1
Content Based Filtering - Evaluation Threads	1

© Copyright IBM Corporation 2015

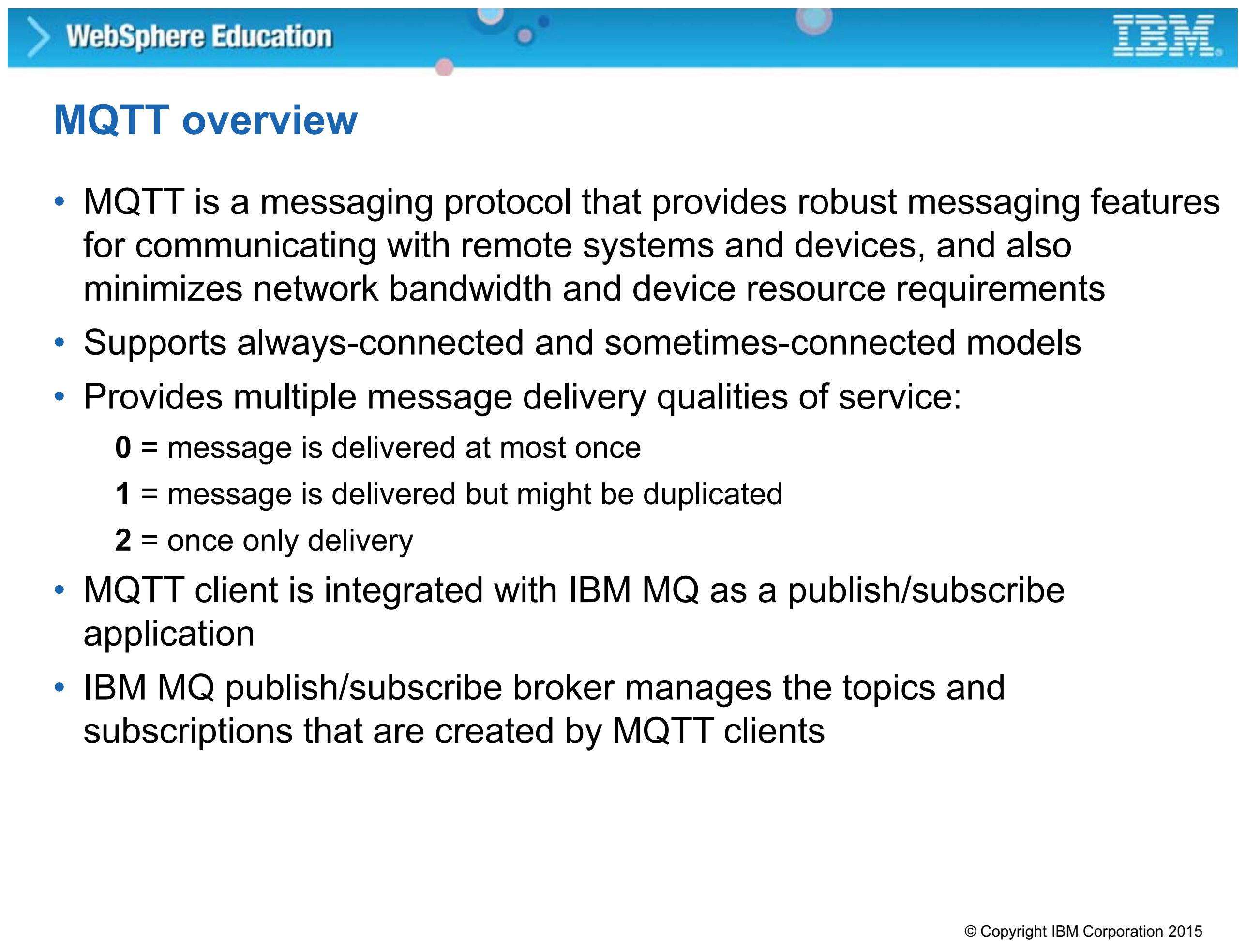
Figure 11-17. Configure content-based filtering in Integration web interface WM6461.0

Notes:

You can view and modify the content-based filtering properties in the Integration web interface by clicking an integration server and then expanding the **Resource Manager Properties** on the **Overview** tab.

You must restart the integration server for the change to take effect.

11.3. Publish/subscribe with MQTT



MQTT overview

- MQTT is a messaging protocol that provides robust messaging features for communicating with remote systems and devices, and also minimizes network bandwidth and device resource requirements
- Supports always-connected and sometimes-connected models
- Provides multiple message delivery qualities of service:
  - **0** = message is delivered at most once
  - **1** = message is delivered but might be duplicated
  - **2** = once only delivery
- MQTT client is integrated with IBM MQ as a publish/subscribe application
- IBM MQ publish/subscribe broker manages the topics and subscriptions that are created by MQTT clients

Figure 11-18. MQTT overview WM6461.0

Notes:

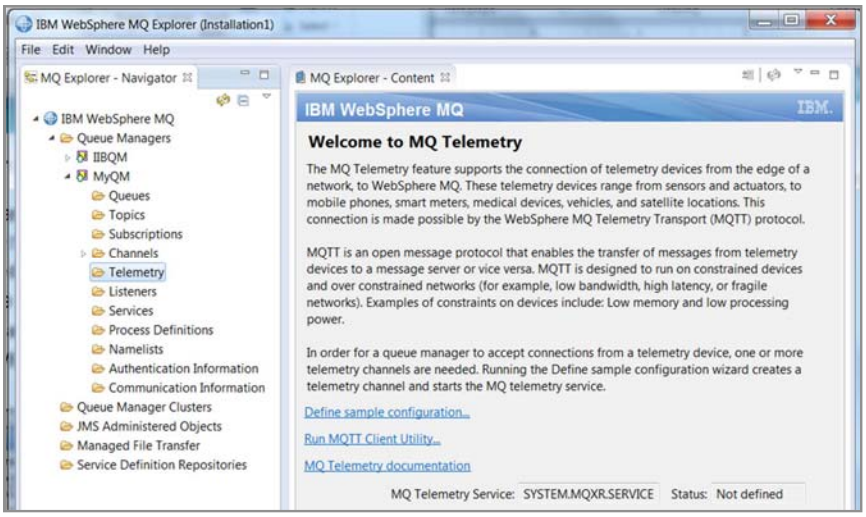
MQTT is a protocol that is designed for devices in constrained environments, such as embedded systems, cell phones, and sensors with limited processing ability and memory, and for systems that are connected to unreliable networks.

IBM MQ can be used to manage MQTT topics and subscriptions.

WebSphere Education

MQTT administration with IBM MQ Explorer

- MQTT subscriptions appear as IBM MQ subscriptions
  - Subscription name is: `mqtt_`*`MQTTClientIdentifier`*`_`*`Topic`*
  - Can be administered and deleted from Subscriptions view
- MQTT and IBM MQ share the topic space
  - Publish from MQTT to IBM MQ and vice versa
- IBM MQ provides a sample MQTT configuration through the IBM MQ Explorer



Welcome to MQ Telemetry

The MQ Telemetry feature supports the connection of telemetry devices from the edge of a network, to WebSphere MQ. These telemetry devices range from sensors and actuators, to mobile phones, smart meters, medical devices, vehicles, and satellite locations. This connection is made possible by the WebSphere MQ Telemetry Transport (MQTT) protocol.

MQTT is an open message protocol that enables the transfer of messages from telemetry devices to a message server or vice versa. MQTT is designed to run on constrained devices and over constrained networks (for example, low bandwidth, high latency, or fragile networks). Examples of constraints on devices include: Low memory and low processing power.

In order for a queue manager to accept connections from a telemetry device, one or more telemetry channels are needed. Running the Define sample configuration wizard creates a telemetry channel and starts the MQ telemetry service.

Define sample configuration...

Run MQTT Client Utility...

MQ Telemetry documentation

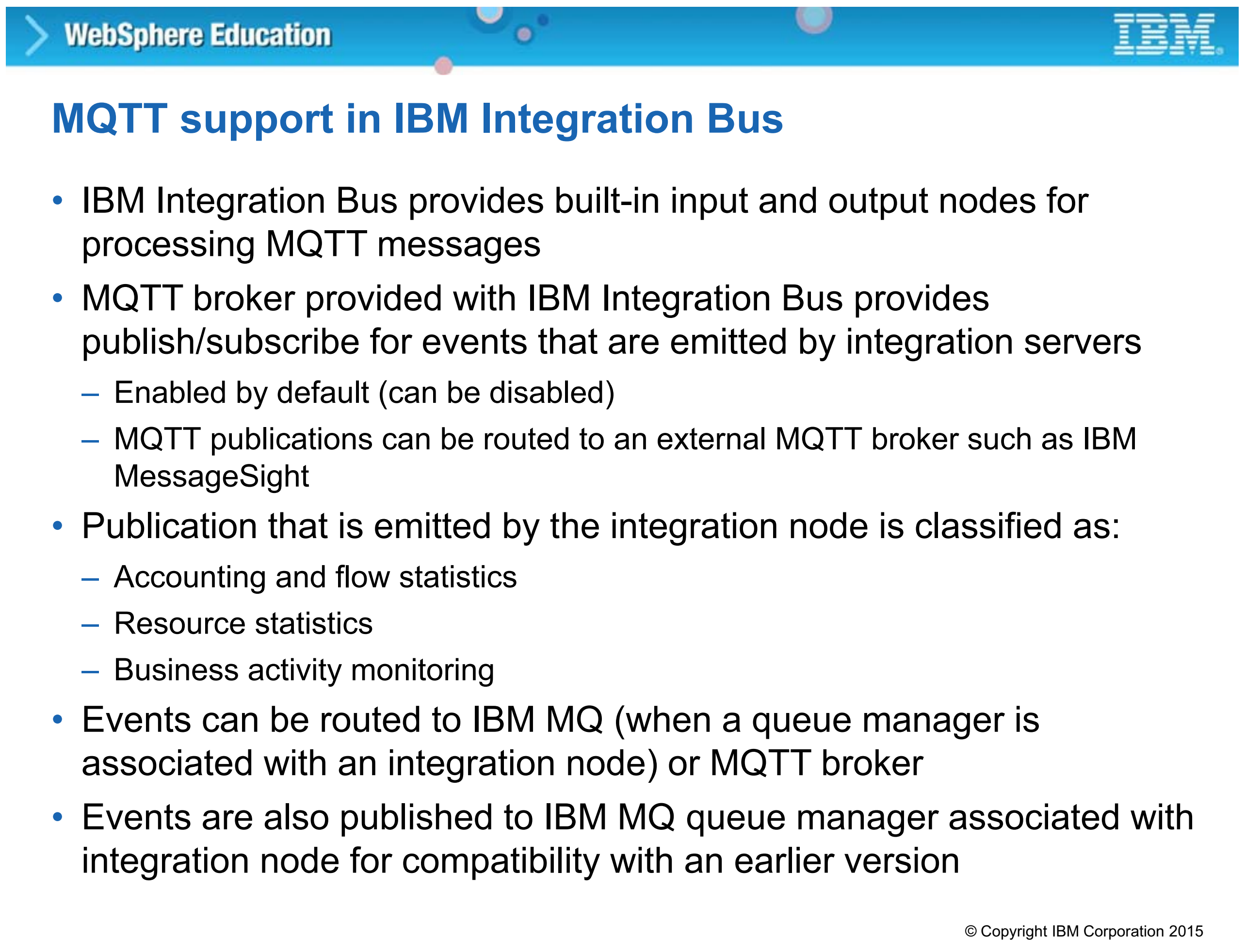
MQ Telemetry Service: SYSTEM.MQXR.SERVICE Status: Not defined

Figure 11-19. MQTT administration with IBM MQ Explorer WM6461.0

Notes:

You can use IBM MQ Explorer to administer MQTT subscriptions and topics.

IBM MQ Explorer also provides a sample MQTT configuration.



MQTT support in IBM Integration Bus

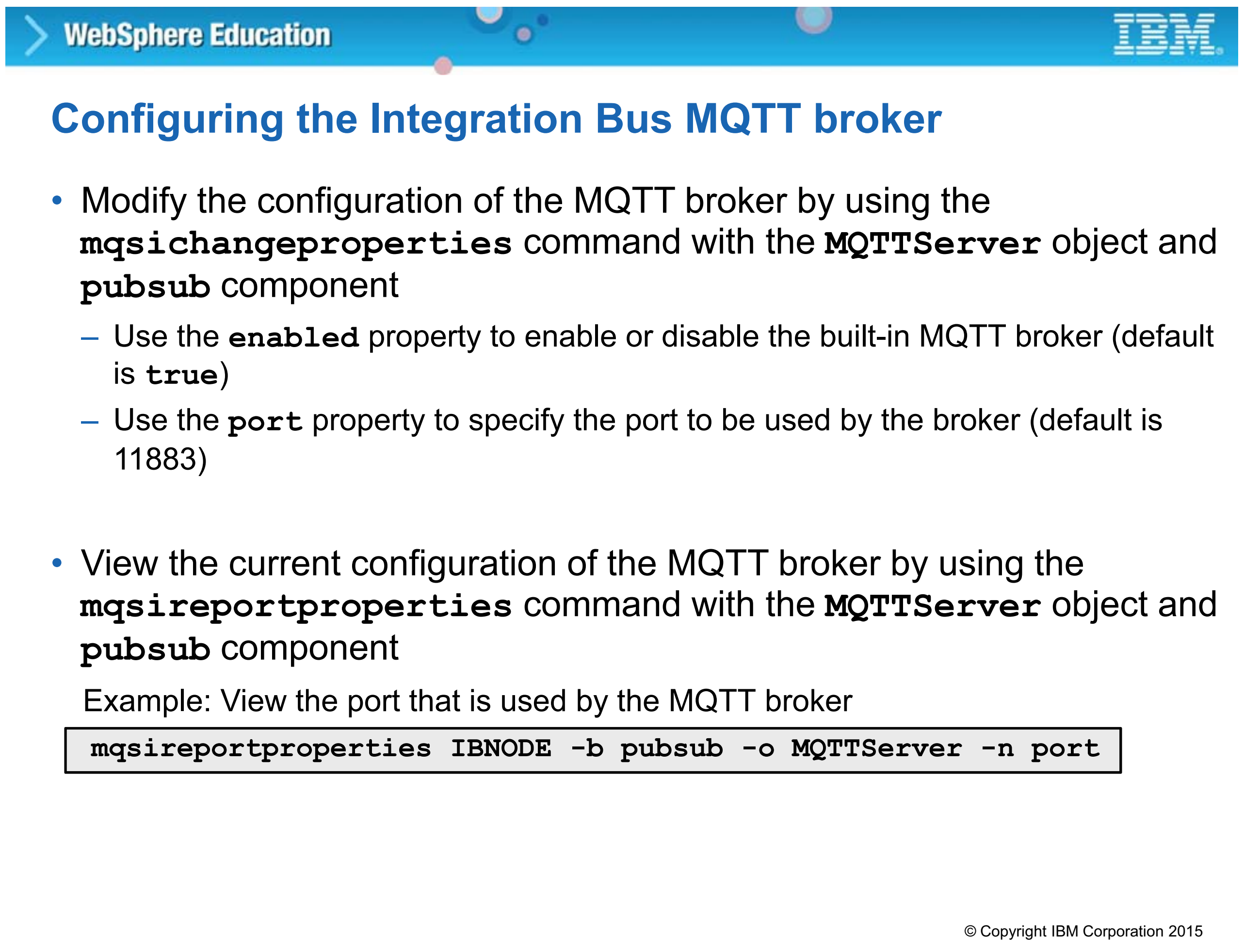
- IBM Integration Bus provides built-in input and output nodes for processing MQTT messages
- MQTT broker provided with IBM Integration Bus provides publish/subscribe for events that are emitted by integration servers
  – Enabled by default (can be disabled)
  – MQTT publications can be routed to an external MQTT broker such as IBM MessageSight
- Publication that is emitted by the integration node is classified as:
  – Accounting and flow statistics
  – Resource statistics
  – Business activity monitoring
- Events can be routed to IBM MQ (when a queue manager is associated with an integration node) or MQTT broker
- Events are also published to IBM MQ queue manager associated with integration node for compatibility with an earlier version

Figure 11-20. MQTT support in IBM Integration Bus WM6461.0

Notes:

In Integration Bus, you can create message flows to receive an MQTT message by using the MQTTSubscribe node to subscribe to one or more topics on an MQTT server. You can send an MQTT message by using the MQTTPublish node in your message flow to publish messages to a topic on an MQTT server.

Integration Bus publishes status and event information in known topics. You can use the Integration Bus built-in MQTT broker for events that integration servers emit.



Configuring the Integration Bus MQTT broker

- Modify the configuration of the MQTT broker by using the **`mqsichangeproperties`** command with the **`MQTTServer`** object and **`pubsub`** component
  - Use the **`enabled`** property to enable or disable the built-in MQTT broker (default is **`true`**)
  - Use the **`port`** property to specify the port to be used by the broker (default is 11883)
- View the current configuration of the MQTT broker by using the **`mqsireportproperties`** command with the **`MQTTServer`** object and **`pubsub`** component

  Example: View the port that is used by the MQTT broker

```
mqsireportproperties IBNODE -b pubsub -o MQTTServer -n port
```

Figure 11-21. Configuring the Integration Bus MQTT broker WM6461.0

Notes:

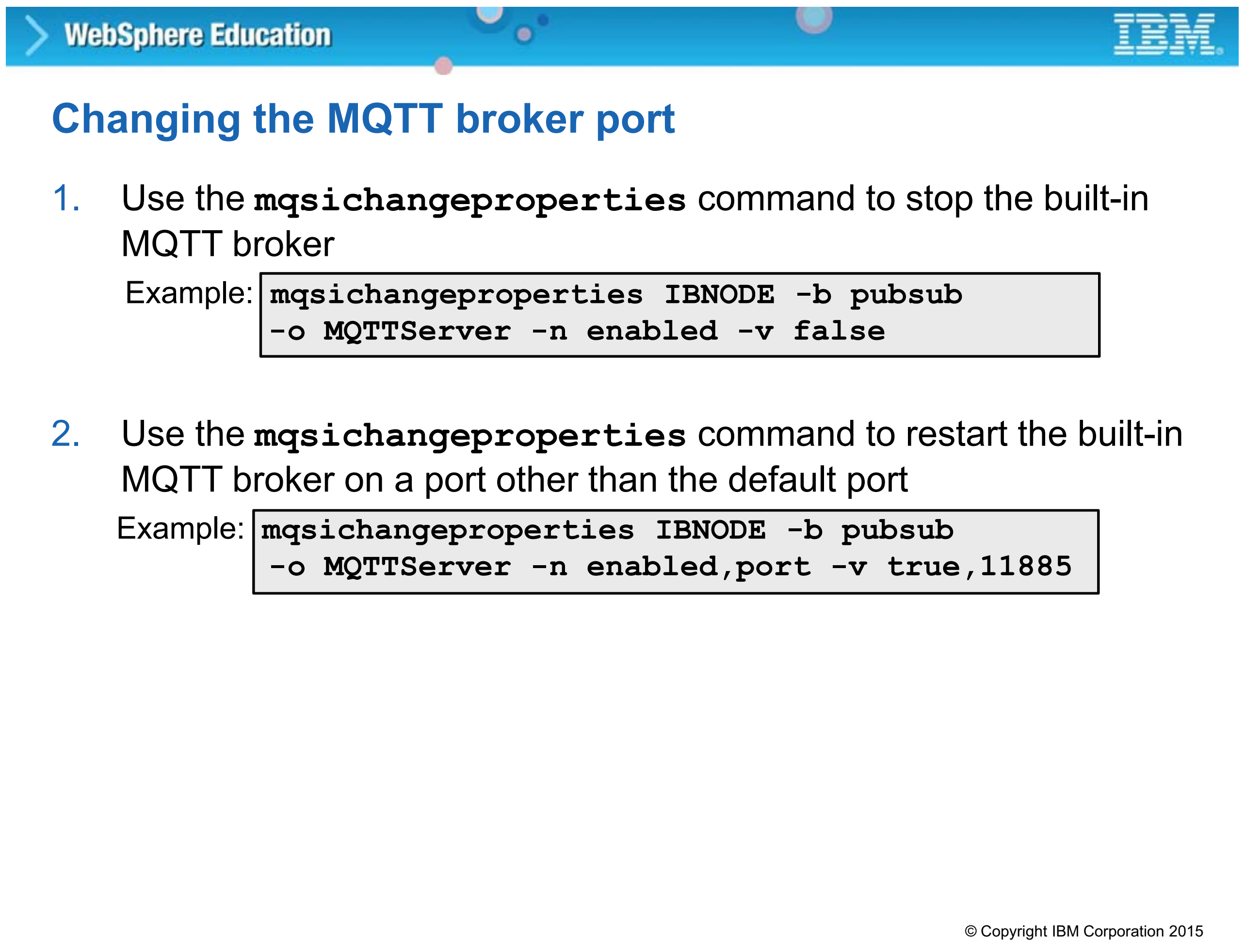
You can modify the configuration of the built-in MQTT broker by using the `mqsichangeproperties` command.

Use the `enabled` property for the `-o MQTTServer` object in the `-b pubsub` component to enable or disable the built-in MQTT broker. Use the `-n port` property to specify the port to be used by the broker. By default, the `enabled` property is set to `true` and the `port` is set to 11883.

When an integration node starts, the built-in MQTT broker starts on the port that is configured by the MQTT server `port` property. If more than one integration node is configured with the same MQTT server port, only one MQTT broker starts. All integration nodes that use the same MQTT server port, use the same MQTT broker to publish their events.

Subscribers that connect to the MQTT broker receive all the events that the broker publishes, unless the subscriber includes the name of the integration node in their subscriptions.

You can view the current configuration of the built-in MQTT broker by using the `mqsireportproperties` command.



WebSphere Education

Changing the MQTT broker port

1. Use the **`mqsichangeproperties`** command to stop the built-in MQTT broker

   Example:
   ```
   mqsichangeproperties IBNODE -b pubsub
   -o MQTTServer -n enabled -v false
   ```

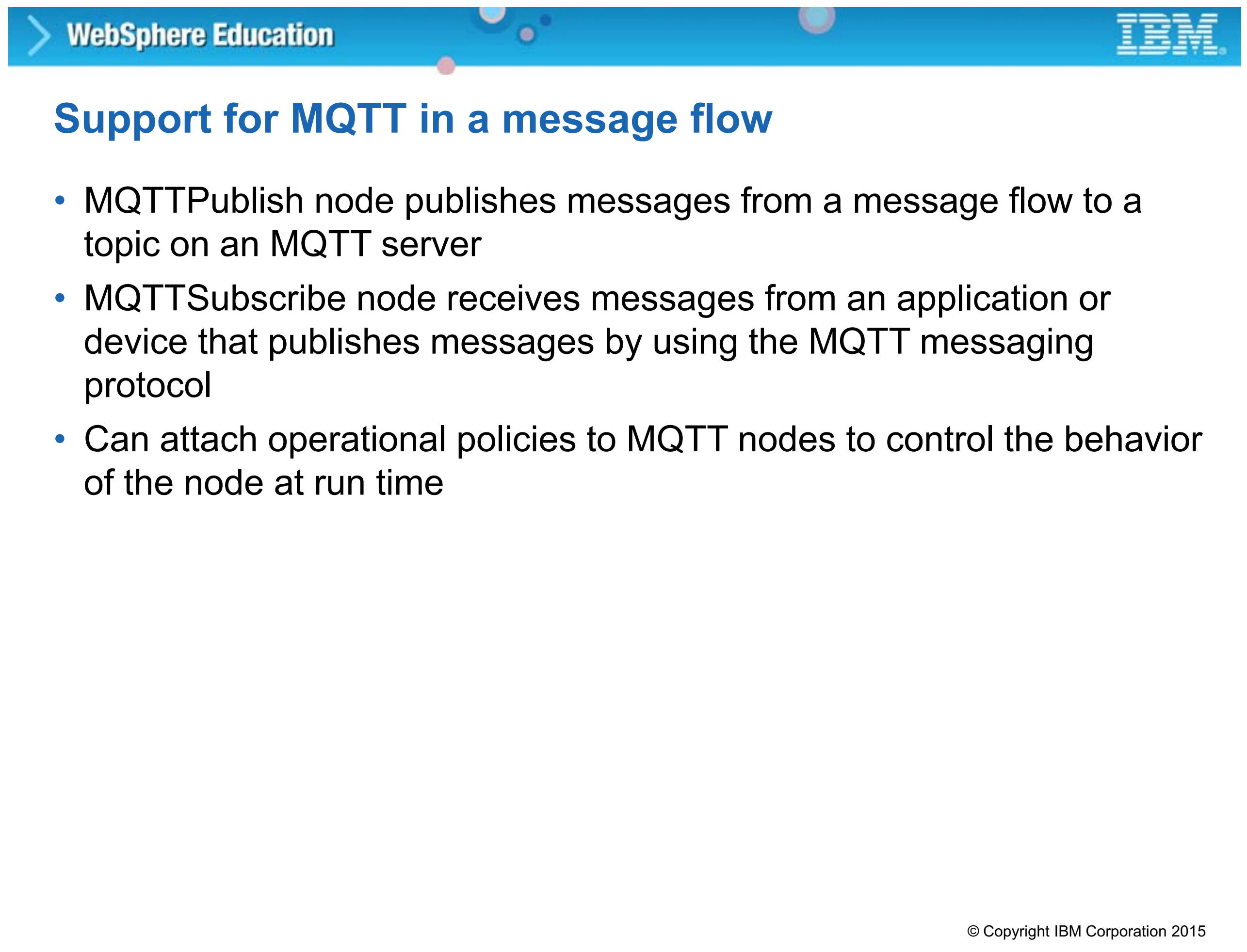
2. Use the **`mqsichangeproperties`** command to restart the built-in MQTT broker on a port other than the default port

   Example:
   ```
   mqsichangeproperties IBNODE -b pubsub
    -o MQTTServer -n enabled,port -v true,11885
   ```

Figure 11-22. Changing the MQTT broker port WM6461.0

Notes:

You can use the `mqsichangeproperties` command to stop the built-in MQTT broker and change the MQTT server port, as shown in the figure.



WebSphere Education
IBM

Support for MQTT in a message flow

- MQTTPublish node publishes messages from a message flow to a topic on an MQTT server
- MQTTSubscribe node receives messages from an application or device that publishes messages by using the MQTT messaging protocol
- Can attach operational policies to MQTT nodes to control the behavior of the node at run time

Figure 11-23. Support for MQTT in a message flow WM6461.0

Notes:

A developer can use the MQTTPublish node to publish messages from a message flow to a topic on an MQTT server. When you use the MQTTPublish node in a message flow, you can dynamically override some of its properties with elements in the local environment message tree.

A developer can use the MQTTSubscribe node to receive messages from an application or device that publishes messages by using the MQTT messaging protocol. Integration Bus can then propagate these messages in a message flow.

WebSphere Education IBM

MQTT operational policies

- Manage by using the IBM Integration web interface, Integration Toolkit, or **`mqsicreatepolicy`** command
- Stored in Integration Registry on the integration node
- Can update at run time without redeployment of an application or integration service

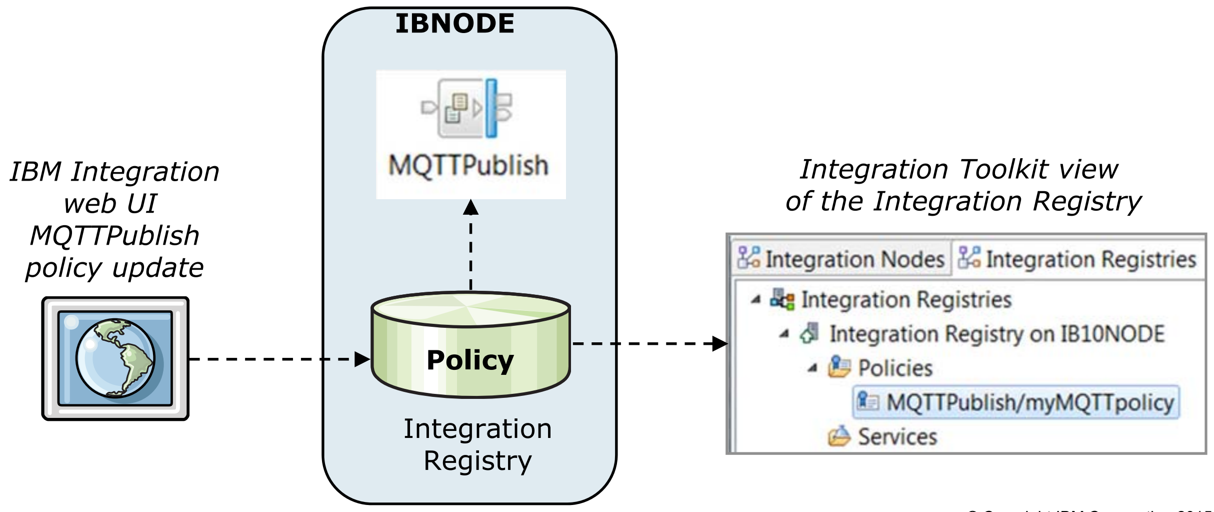


Figure 11-24. MQTT operational policies WM6461.0

Notes:

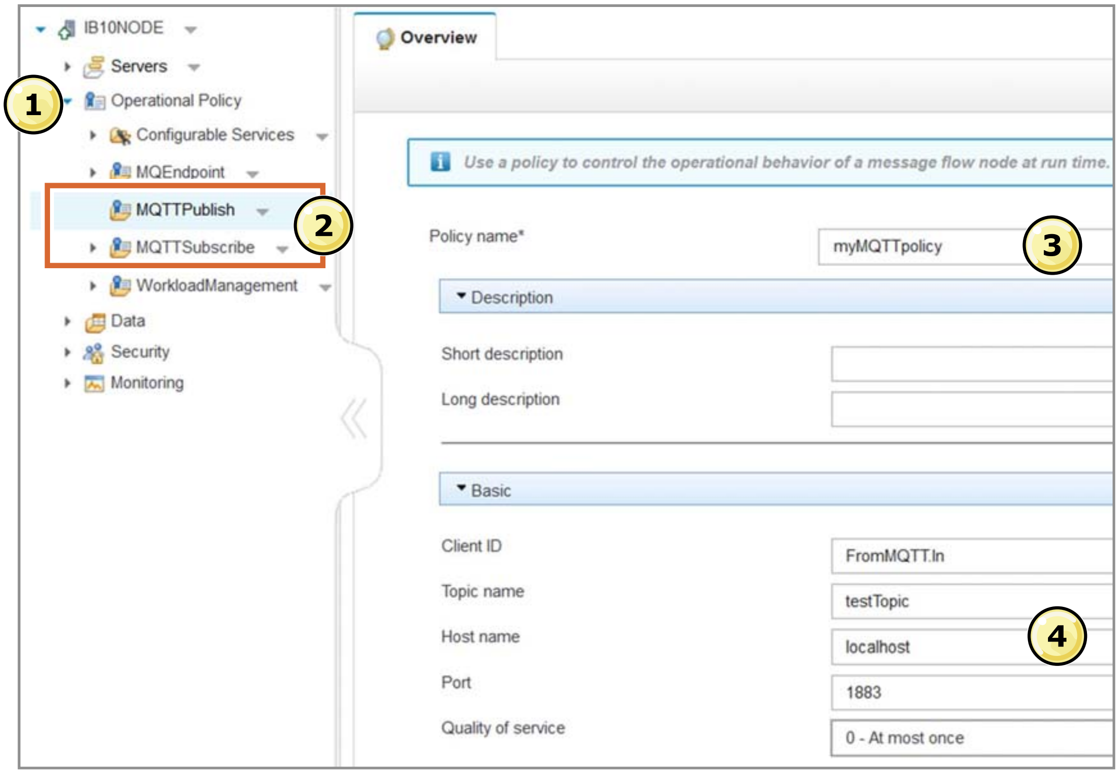
Administrators and developers can use operational policies to control the behavior of message flows, and the nodes within message flows, at run time without redeploying resources.

- An MQTTPublish policy can be attached to one or more MQTTPublish nodes in a message flow to control the value of specific MQTT publishing properties at run time.
- An MQTTSubscribe policy can be attached to one or more MQTTSubscribe nodes in a message flow to control the value of specific MQTT subscription properties at run time.

Policies can be created and updated at any time in the solution lifecycle. A policy instance is made up of a set of defined operational properties that are stored in a policy document, and a policy instance can be applied at the message flow, or node level. You can set operational property values on the node at development time, override them in the BAR file, and set them in an operational policy.

WebSphere Education

Creating MQTT policies in the Integration web interface



Use a policy to control the operational behavior of a message flow node at run time.

Policy name* myMQTTpolicy

Description

Short description

Long description

Basic

Client ID FromMQTT.In

Topic name testTopic

Host name localhost

Port 1883

Quality of service 0 - At most once

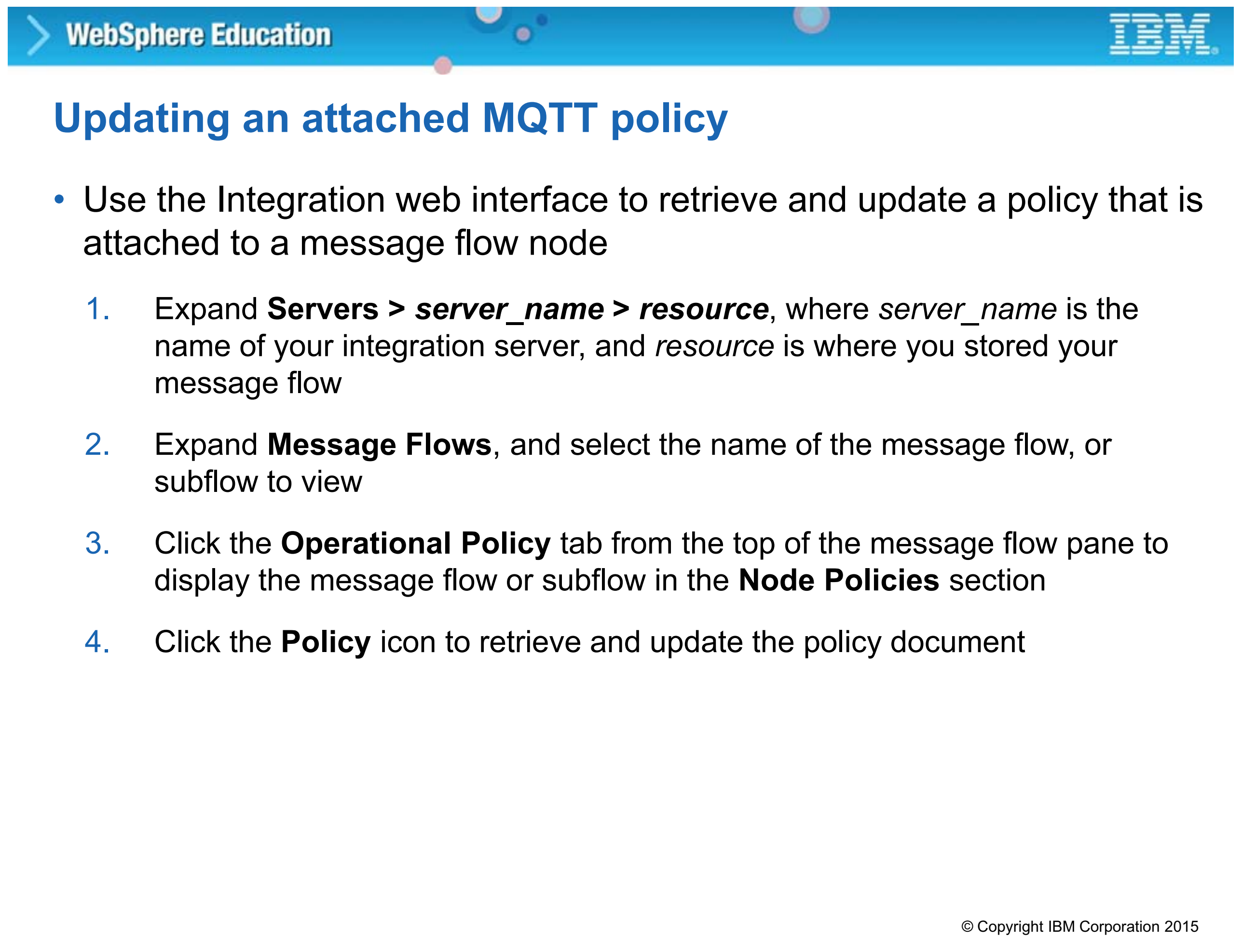
Figure 11-25. Creating MQTT policies in the Integration web interface WM6461.0

Notes:

You can define operational policies in the Integration Toolkit or by using the Integration web interface.

To define an MQTTPublish or MQTTSubscribe policy with the Integration web interface:

1. Expand **Operational Policy**.
2. Click **MQTTPublish > Create** or click **MQTTSubscribe > Create**.
3. Enter a policy name and edit the property values as required. With the **Save As** function, you can save an existing policy with a new name.



Updating an attached MQTT policy

- Use the Integration web interface to retrieve and update a policy that is attached to a message flow node
  1. Expand **Servers > *server_name* > *resource***, where *server_name* is the name of your integration server, and *resource* is where you stored your message flow
  2. Expand **Message Flows**, and select the name of the message flow, or subflow to view
  3. Click the **Operational Policy** tab from the top of the message flow pane to display the message flow or subflow in the **Node Policies** section
  4. Click the **Policy** icon to retrieve and update the policy document

Figure 11-26. Updating an attached MQTT policy WM6461.0

Notes:

You can also use the message flow view to retrieve and update a policy that is attached to a message flow node.

1. In the navigation tree, expand **Servers > server_name > resource**, where *server_name* is the name of your integration server, and *resource* is where you stored your message flow.
2. Expand **Message Flows**, and select the name of the message flow, or subflow, you want to view.
3. Click the **Operational Policy** tab from the top of the message flow pane, and the message flow, or subflow, is displayed in the **Node Policies** section.
4. If the message flow, or subflow, includes a node that has an operational policy that is attached, the policy icon is displayed on the upper-right corner of the node icon. Click the **Policy** icon to retrieve and update the policy document.

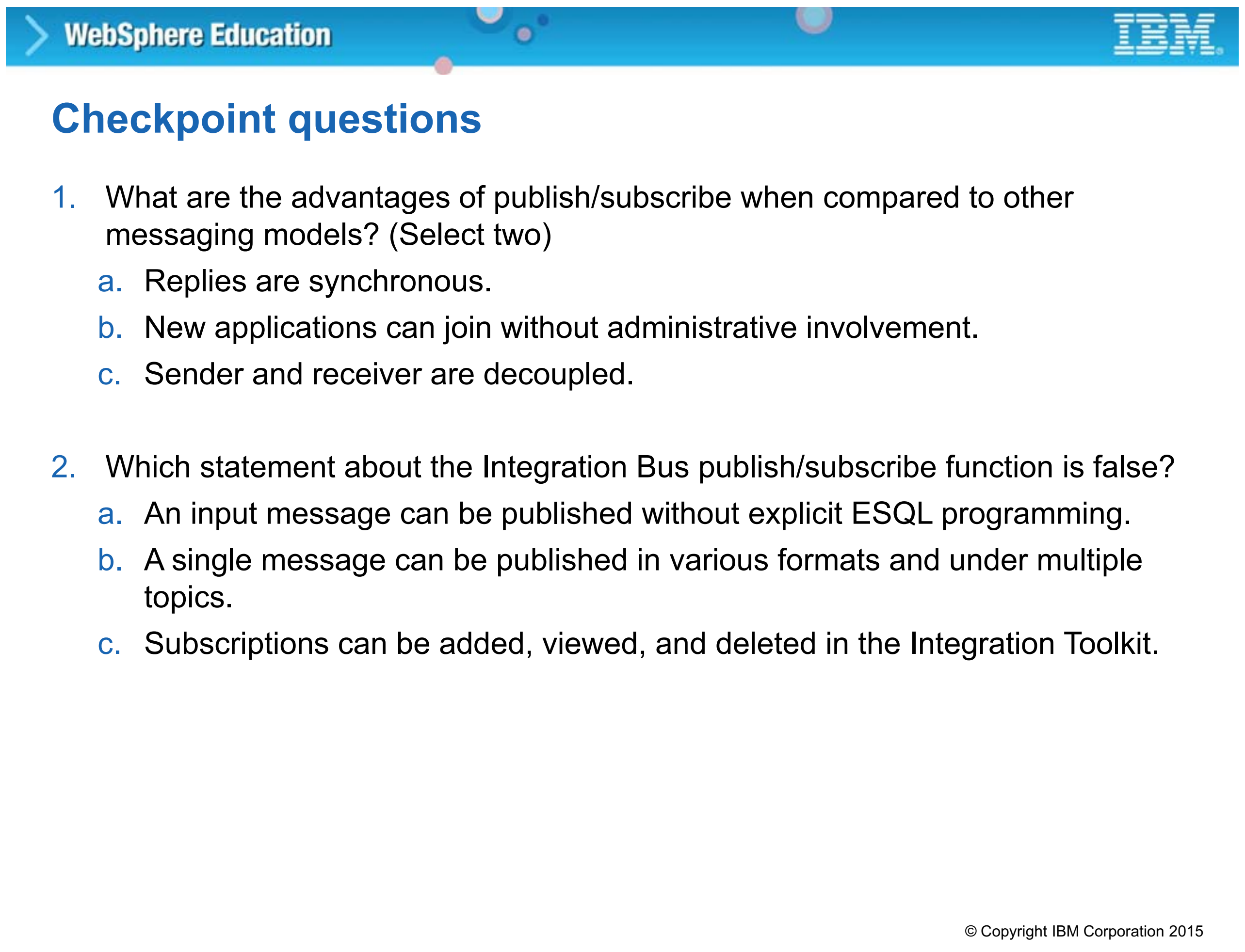
Unit summary

Having completed this unit, you should be able to:

- Manage IBM MQ and IBM Integration Bus publish/subscribe integration
- Manage MQTT and IBM Integration Bus publish/subscribe integration

Figure 11-27. Unit summary WM6461.0

Notes:



WebSphere Education

Checkpoint questions

1. What are the advantages of publish/subscribe when compared to other messaging models? (Select two)
   a. Replies are synchronous.
   b. New applications can join without administrative involvement.
   c. Sender and receiver are decoupled.

2. Which statement about the Integration Bus publish/subscribe function is false?
   a. An input message can be published without explicit ESQL programming.
   b. A single message can be published in various formats and under multiple topics.
   c. Subscriptions can be added, viewed, and deleted in the Integration Toolkit.

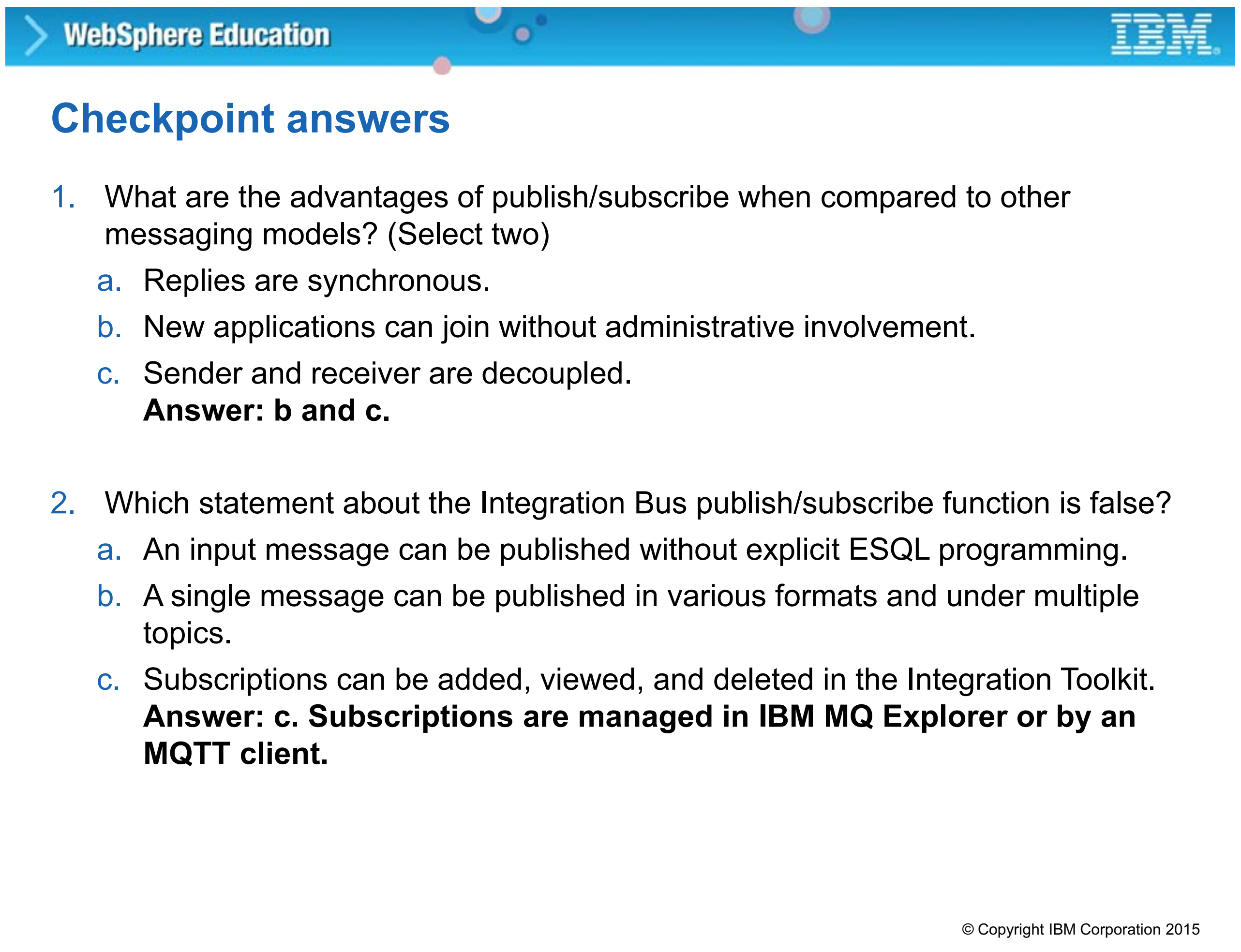
Figure 11-28. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



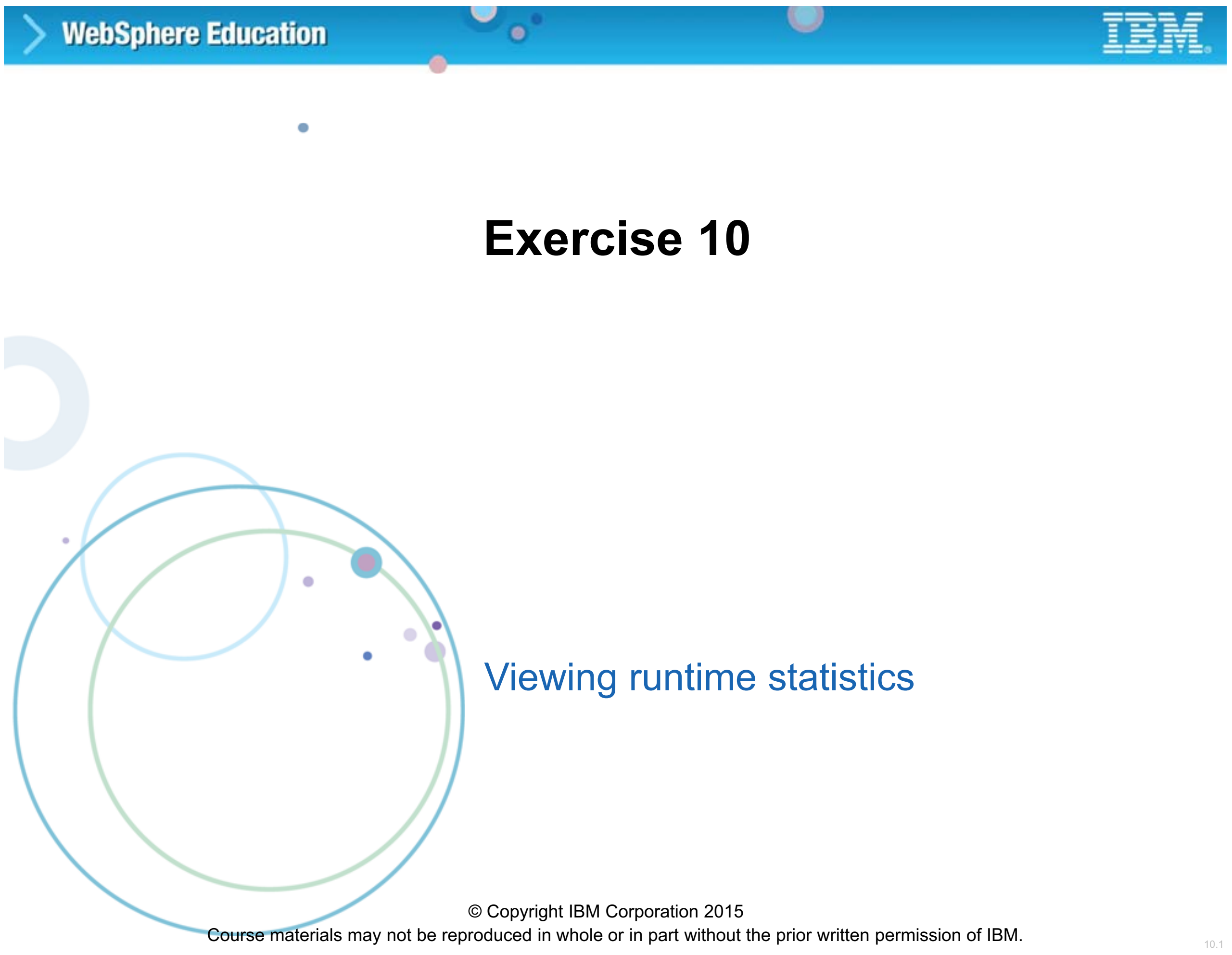
Checkpoint answers

1. What are the advantages of publish/subscribe when compared to other messaging models? (Select two)
   a. Replies are synchronous.
   b. New applications can join without administrative involvement.
   c. Sender and receiver are decoupled.
   Answer: b and c.

2. Which statement about the Integration Bus publish/subscribe function is false?
   a. An input message can be published without explicit ESQL programming.
   b. A single message can be published in various formats and under multiple topics.
   c. Subscriptions can be added, viewed, and deleted in the Integration Toolkit.
   Answer: c. Subscriptions are managed in IBM MQ Explorer or by an MQTT client.

Figure 11-29. Checkpoint answers WM6461.0

Notes:



WebSphere Education

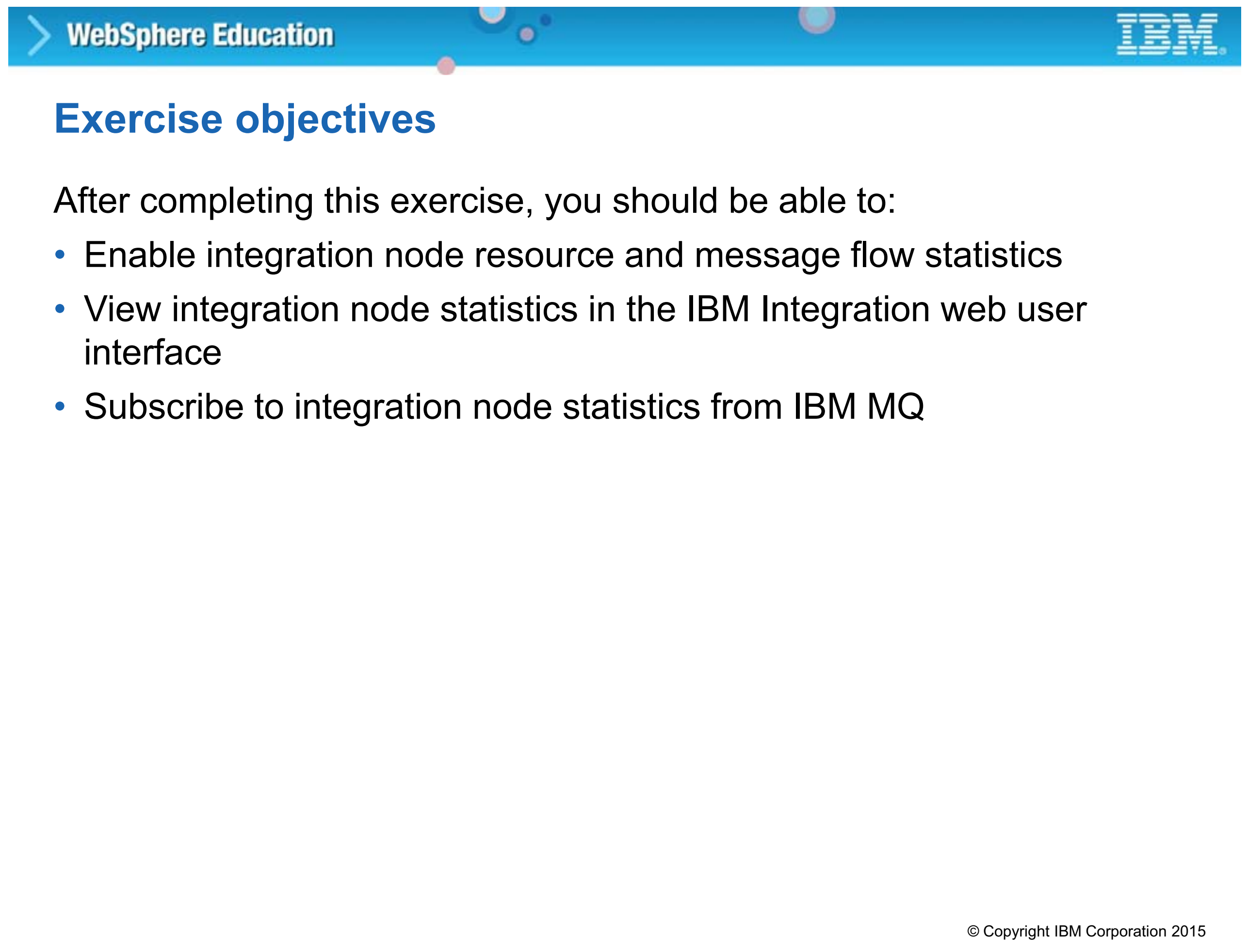
Exercise 10

Viewing runtime statistics

10.1

Figure 11-30. Exercise 10 WM6461.0

Notes:



Exercise objectives

After completing this exercise, you should be able to:
- Enable integration node resource and message flow statistics
- View integration node statistics in the IBM Integration web user interface
- Subscribe to integration node statistics from IBM MQ

Figure 11-31. Exercise objectives WM6461.0

Notes:

Proceed to the *Student Exercise Guide* and complete Exercise 10.

Unit 12. Configuring IBM Integration Bus for high availability

What this unit is about

In this unit, you learn the basic configurations that take advantage of high availability processing in the IBM Integration Bus environment. The unit includes how to use workload management policies to specify goals for the processing rate of a message flow.

What you should be able to do

After completing this unit, you should be able to:

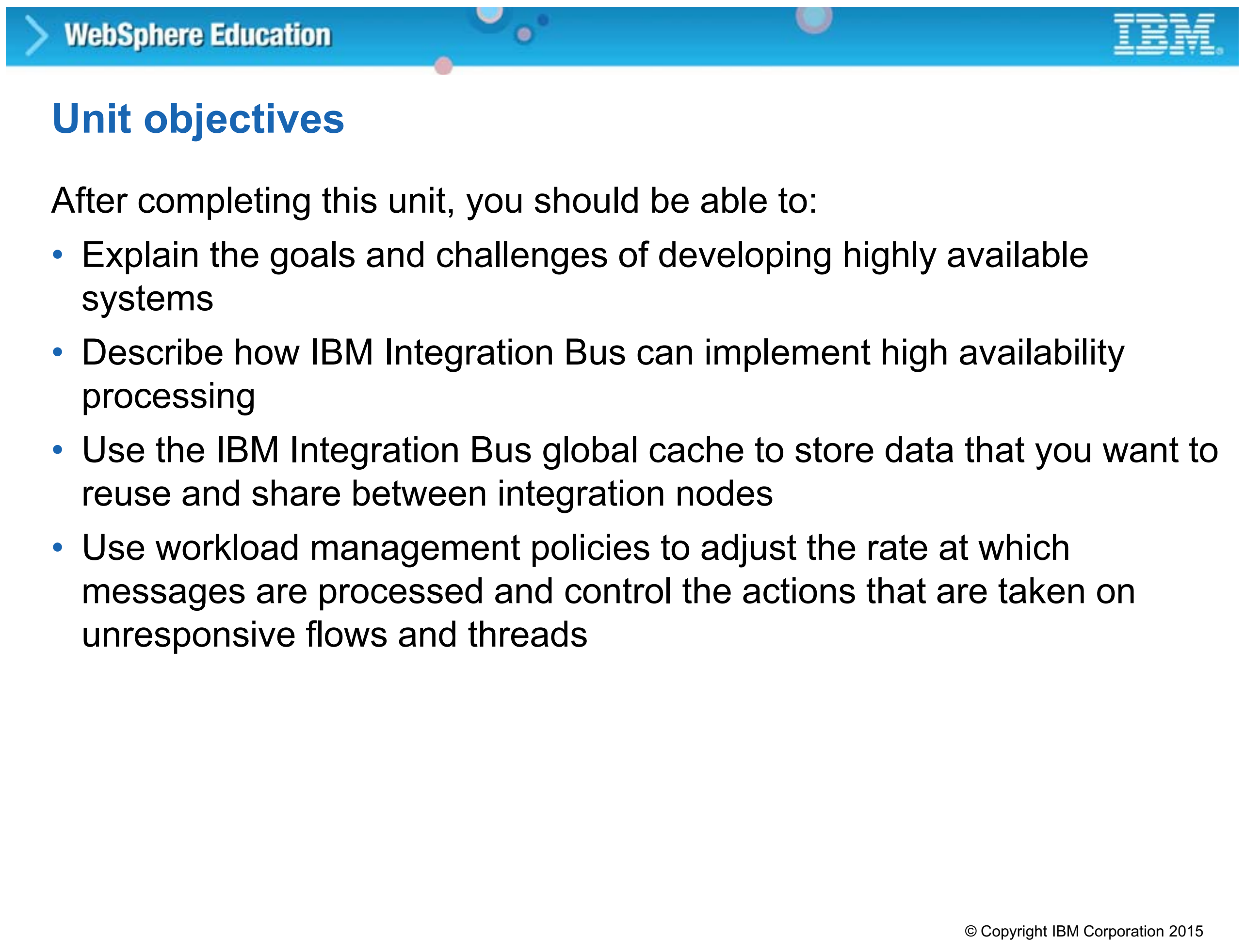
- Explain the goals and challenges of developing highly available systems
- Describe how IBM Integration Bus can implement high availability processing
- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Use workload management policies to adjust the rate at which messages are processed and control the actions that are taken on unresponsive flows and threads

How you will check your progress

- Checkpoints
- Exercise: Managing the workload

References

IBM Knowledge Center for IBM Integration Bus V10



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

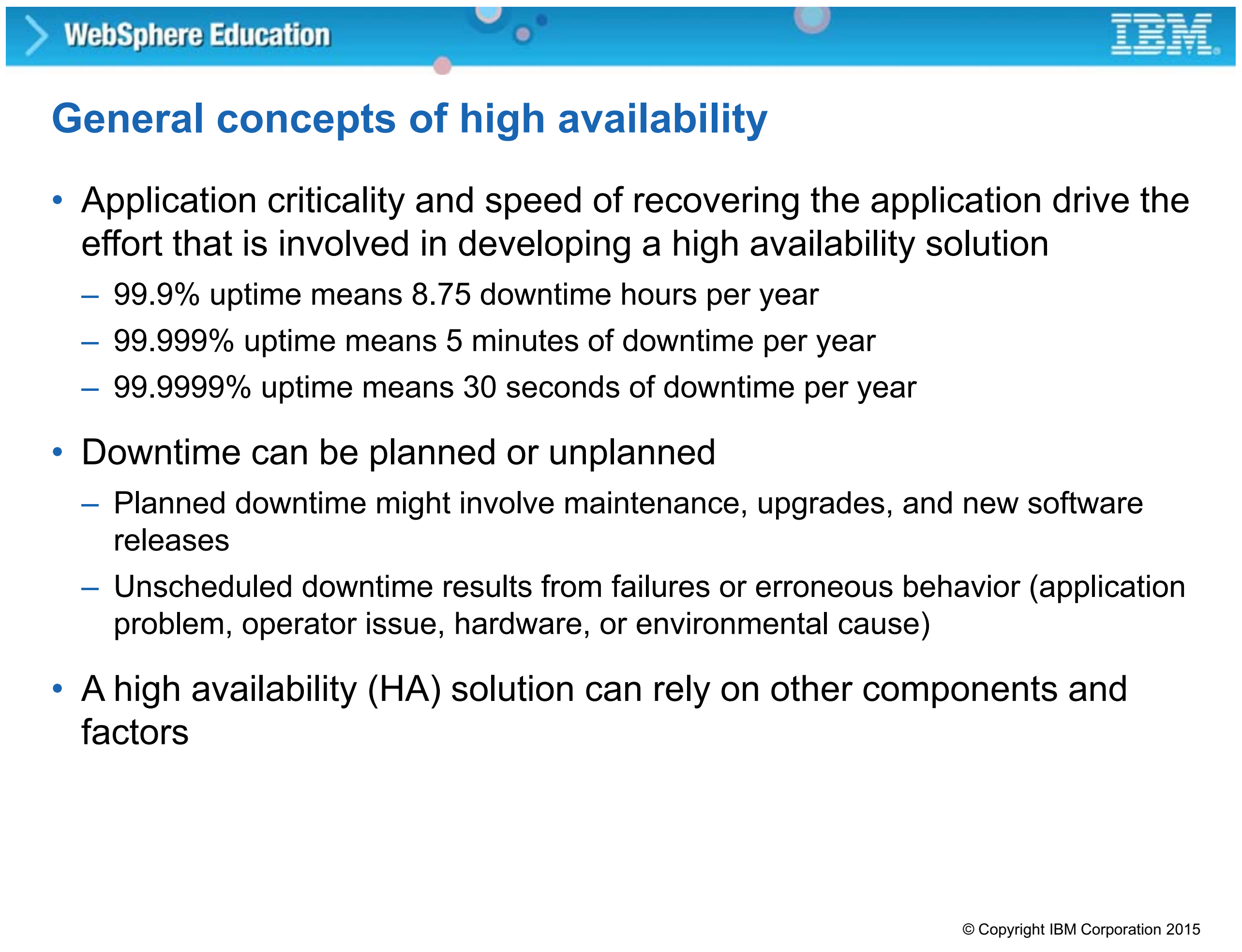
- Explain the goals and challenges of developing highly available systems
- Describe how IBM Integration Bus can implement high availability processing
- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Use workload management policies to adjust the rate at which messages are processed and control the actions that are taken on unresponsive flows and threads

© Copyright IBM Corporation 2015

Figure 12-1. Unit objectives WM6461.0

Notes:

12.1.High availability concepts



WebSphere Education

General concepts of high availability

- Application criticality and speed of recovering the application drive the effort that is involved in developing a high availability solution
  - 99.9% uptime means 8.75 downtime hours per year
  - 99.999% uptime means 5 minutes of downtime per year
  - 99.9999% uptime means 30 seconds of downtime per year
- Downtime can be planned or unplanned
  - Planned downtime might involve maintenance, upgrades, and new software releases
  - Unscheduled downtime results from failures or erroneous behavior (application problem, operator issue, hardware, or environmental cause)
- A high availability (HA) solution can rely on other components and factors

Figure 12-2. General concepts of high availability WM6461.0

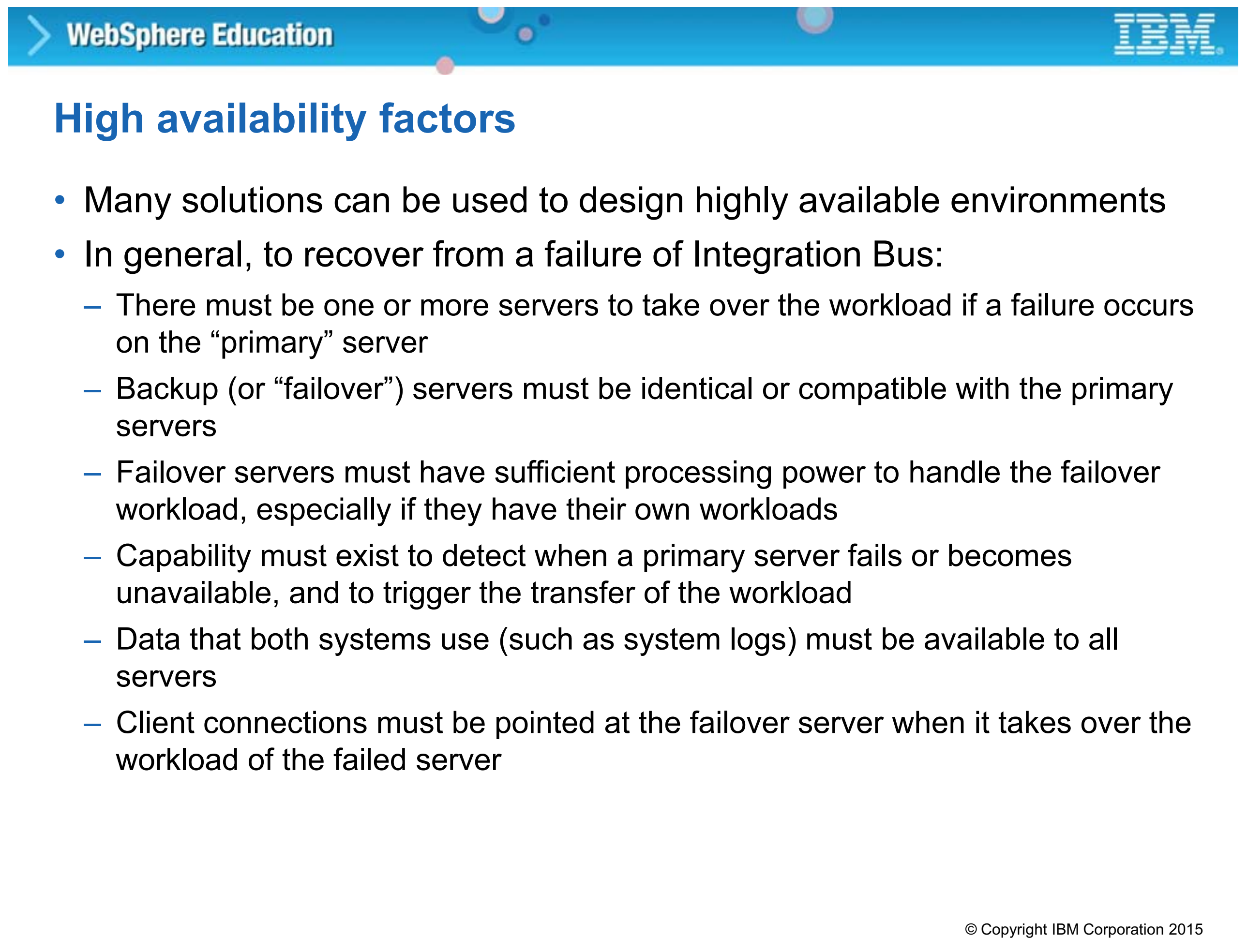
Notes:

High availability (HA) solutions are typically designed to minimize the impact of a failure. The following are examples of failures.

- A hardware failure, such as a disk failure or corruption
- A software failure, such as an unexpected or unintended application problem
- An environmental problem, such as a power failure or a water leak in a server room
- A human error that leads to an application outage

The cost of a failure or an outage can be high. Conversely, the cost of developing a highly available solution can also be high. The specific needs for the availability of an application system depend on many factors. One application might be able to tolerate some amount of outage, while another application might not.

An Integration Bus application or system can be made highly available. However, in most cases there are many more factors to consider beyond just the Integration Bus environment.



High availability factors

- Many solutions can be used to design highly available environments
- In general, to recover from a failure of Integration Bus:
  - There must be one or more servers to take over the workload if a failure occurs on the “primary” server
  - Backup (or “failover”) servers must be identical or compatible with the primary servers
  - Failover servers must have sufficient processing power to handle the failover workload, especially if they have their own workloads
  - Capability must exist to detect when a primary server fails or becomes unavailable, and to trigger the transfer of the workload
  - Data that both systems use (such as system logs) must be available to all servers
  - Client connections must be pointed at the failover server when it takes over the workload of the failed server

Figure 12-3. High availability factors WM6461.0

Notes:

There is more than one way to implement a highly available Integration Bus application. This figure describes some of the requirements. All high availability solutions must address these requirements.

WebSphere Education

Failover processing: Normal mode

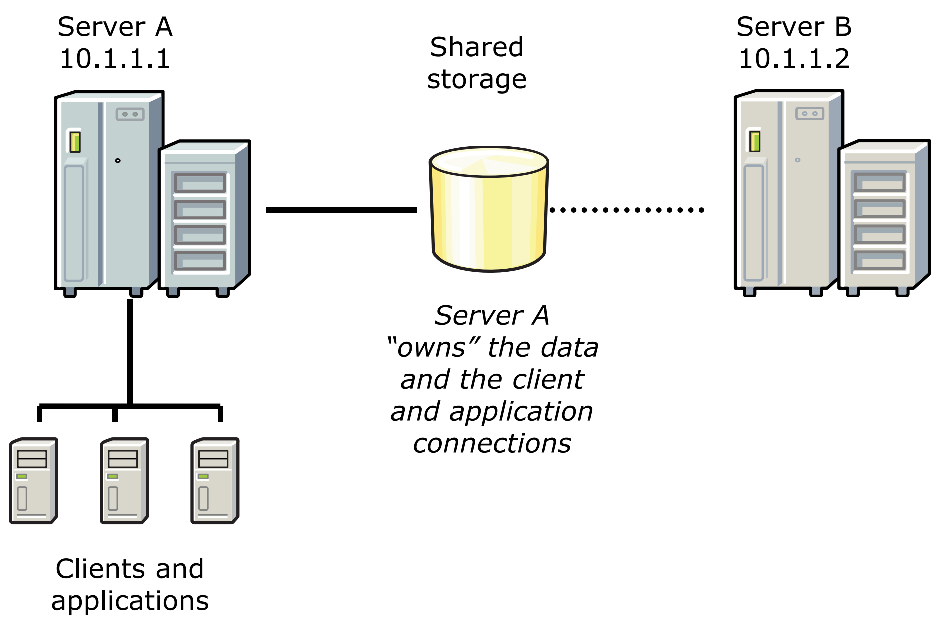
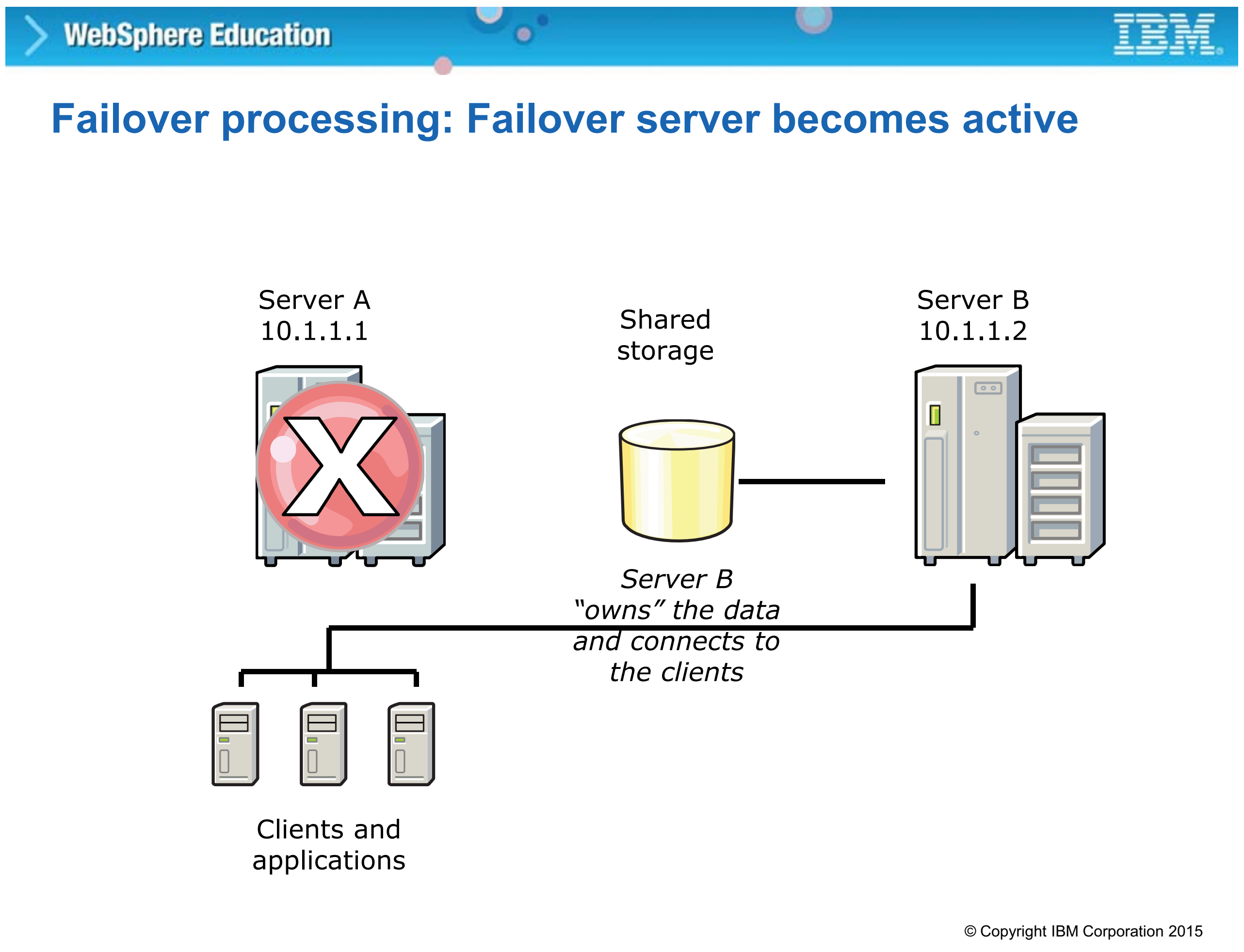

Server A
10.1.1.1
Shared
storage
Server B
10.1.1.2
Server A "owns" the data and the client and application connections
Clients and
applications

Figure 12-4. Failover processing: Normal mode WM6461.0

Notes:

The figure shows two servers that implement failover processing. Applications run on Server A and Server B. Server A exclusively controls the data on the shared storage because it is the primary server. Client applications connect to Server A.



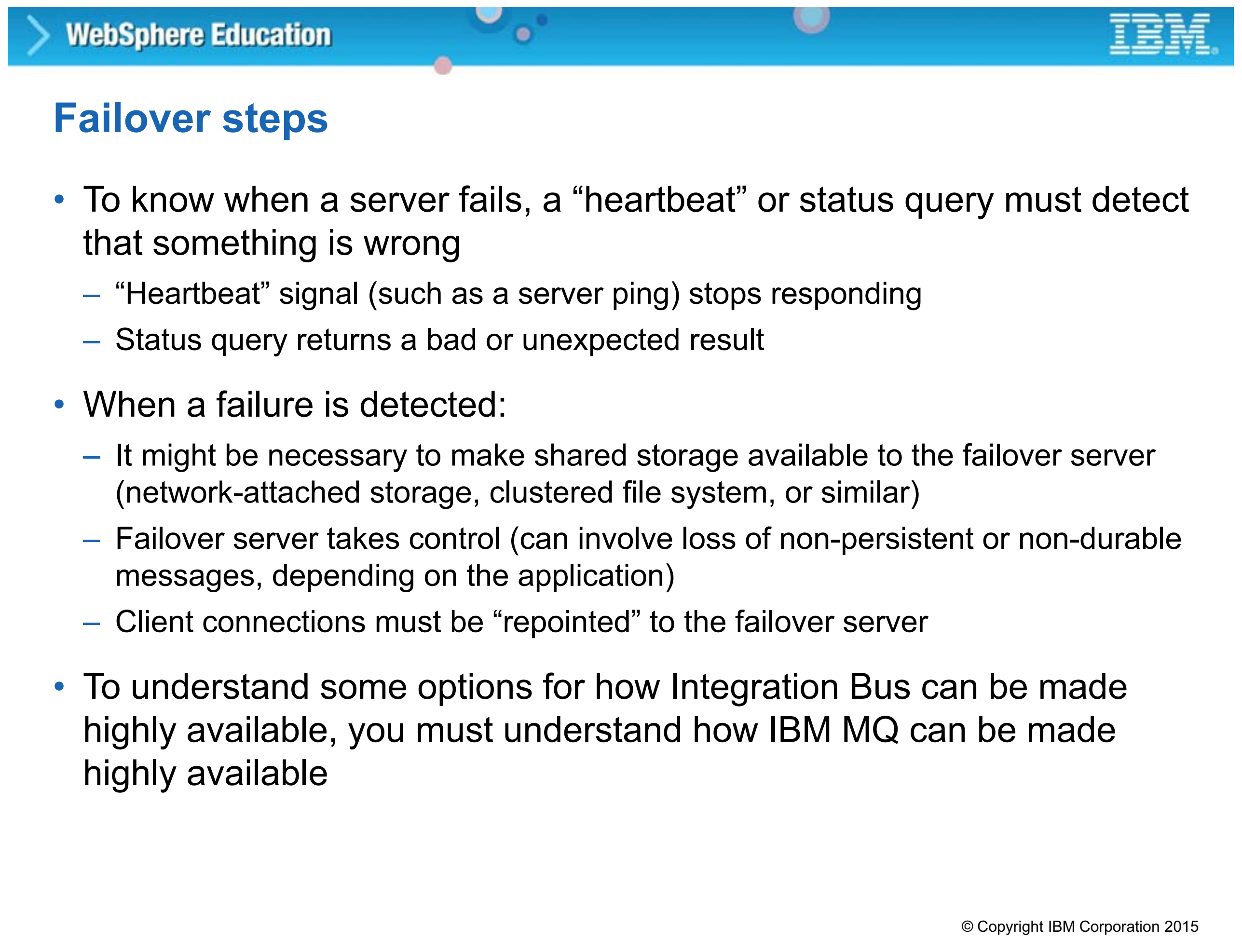
Failover processing: Failover server becomes active

Figure 12-5. Failover processing: Failover server becomes active WM6461.0

Notes:

This figure shows the state of the network when Server A fails.

When Server A fails, Server B is now online. It takes over as the owner of the data on shared storage. It begins processing workload when the client application connections are established to Server B.



WebSphere Education

Failover steps

- To know when a server fails, a “heartbeat” or status query must detect that something is wrong
  - “Heartbeat” signal (such as a server ping) stops responding
  - Status query returns a bad or unexpected result
- When a failure is detected:
  - It might be necessary to make shared storage available to the failover server (network-attached storage, clustered file system, or similar)
  - Failover server takes control (can involve loss of non-persistent or non-durable messages, depending on the application)
  - Client connections must be “repointed” to the failover server
- To understand some options for how Integration Bus can be made highly available, you must understand how IBM MQ can be made highly available

© Copyright IBM Corporation 2015

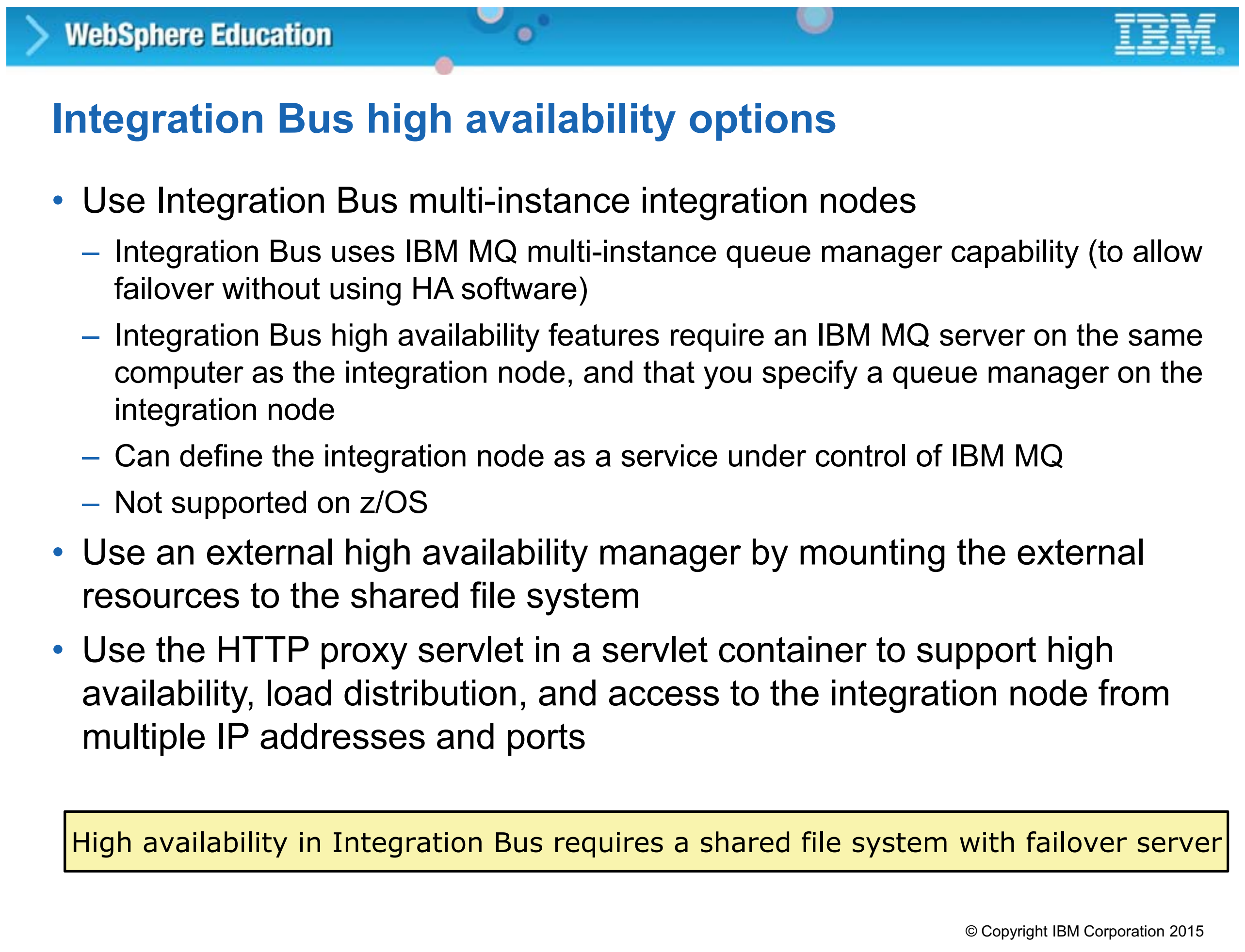
Figure 12-6. Failover steps WM6461.0

Notes:

To initiate a failover, there must be a way to know that a server failed or is no longer communicating. You can determine that a failover occurs by establishing a “heartbeat” connection that periodically polls or communicates with the primary server. When the server stops responding to the heartbeat signal or query, a failover must be initiated.

When a failure is detected, some steps must be taken for failover to occur. One of these events is the failover server “taking possession” of the shared storage. For a true high availability solution, this taking possession must occur automatically, without human intervention.

Integration Bus depends on IBM MQ for control and management functions (consider the SYSTEM.* queues, for example), so it makes sense to take advantage of IBM MQ high availability capabilities.



WebSphere Education

Integration Bus high availability options

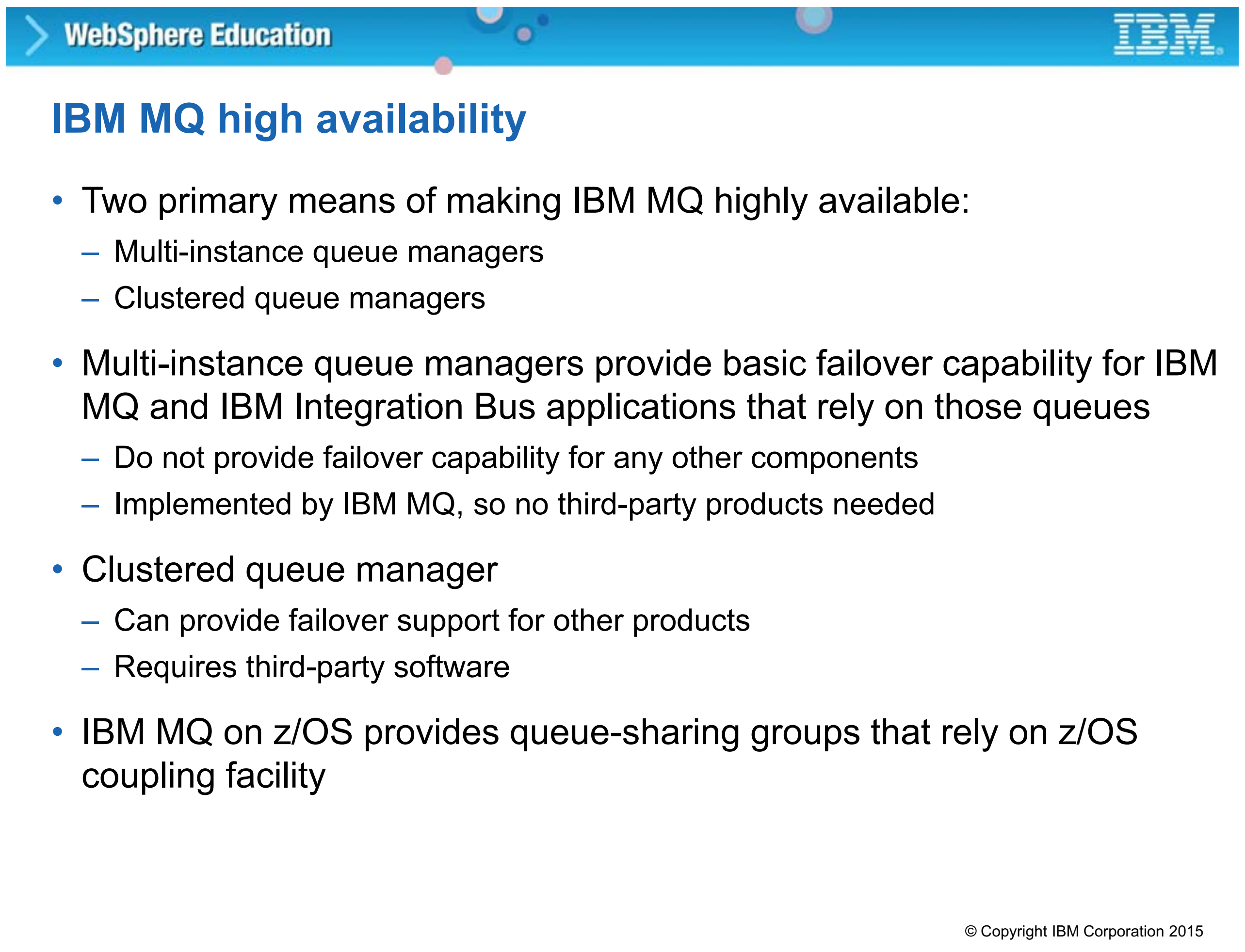
- Use Integration Bus multi-instance integration nodes
  - Integration Bus uses IBM MQ multi-instance queue manager capability (to allow failover without using HA software)
  - Integration Bus high availability features require an IBM MQ server on the same computer as the integration node, and that you specify a queue manager on the integration node
  - Can define the integration node as a service under control of IBM MQ
  - Not supported on z/OS
- Use an external high availability manager by mounting the external resources to the shared file system
- Use the HTTP proxy servlet in a servlet container to support high availability, load distribution, and access to the integration node from multiple IP addresses and ports

High availability in Integration Bus requires a shared file system with failover server

Figure 12-7. Integration Bus high availability options WM6461.0

Notes:

This figure summarizes the Integration Bus high availability options. Many of these options rely on IBM MQ and a shared file system with a failover server.



WebSphere Education

IBM MQ high availability

- Two primary means of making IBM MQ highly available:
  - Multi-instance queue managers
  - Clustered queue managers
- Multi-instance queue managers provide basic failover capability for IBM MQ and IBM Integration Bus applications that rely on those queues
  - Do not provide failover capability for any other components
  - Implemented by IBM MQ, so no third-party products needed
- Clustered queue manager
  - Can provide failover support for other products
  - Requires third-party software
- IBM MQ on z/OS provides queue-sharing groups that rely on z/OS coupling facility

© Copyright IBM Corporation 2015

Figure 12-8. IBM MQ high availability WM6461.0

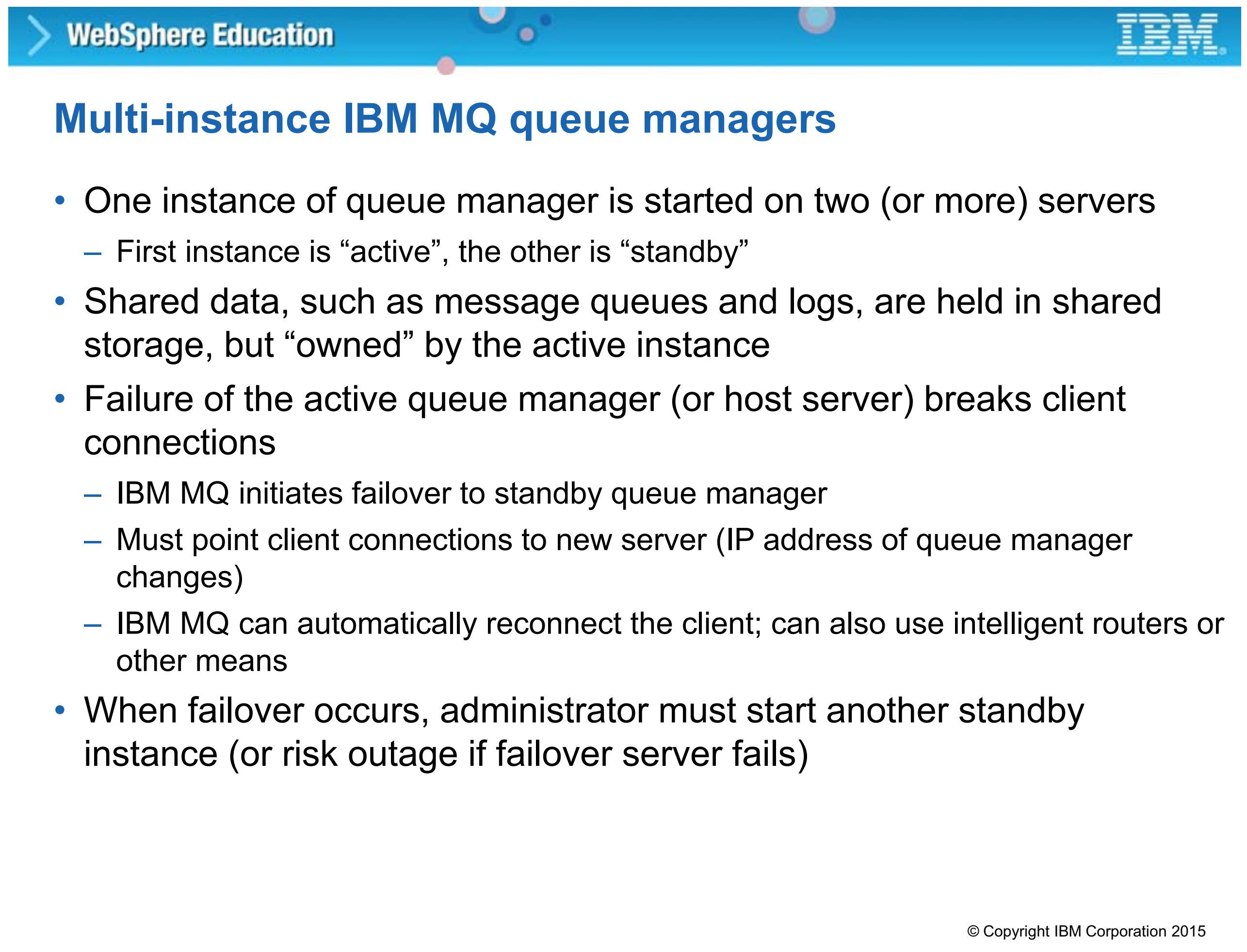
Notes:

IBM MQ supports two primary means of implementing high availability: multi-instance queue managers and queue manager clusters.

Multi-instance queue managers are sufficient for a simple high availability solution. Multi-instance queue managers are instances of the same queue manager that are configured on different servers. One instance of the queue manager is defined as the active instance and another instance is defined as the standby instance. If the active instance fails, the multi-instance queue manager restarts automatically on the standby server.

For more sophisticated recovery scenarios, consider the use of queue manager clusters. High availability clustering uses an external high availability manager by mounting the external resources to the shared file system. Script files provide for the common tasks that are required to create a multi-instance integration node.

If you are working with clusters in the z/OS environment, you can take advantage of the z/OS coupling facility for high availability.



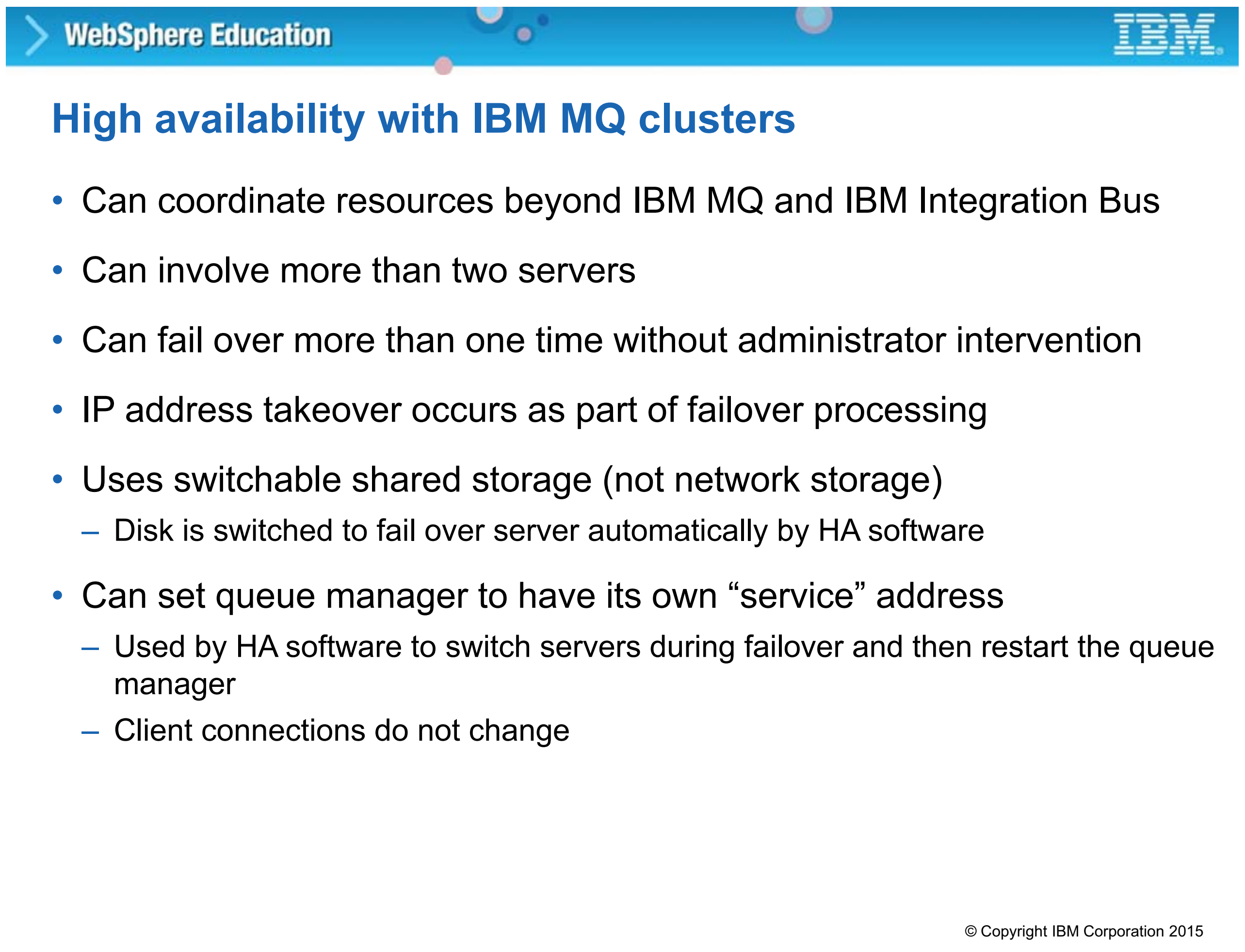
Multi-instance IBM MQ queue managers

- One instance of queue manager is started on two (or more) servers
  - First instance is “active”, the other is “standby”
- Shared data, such as message queues and logs, are held in shared storage, but “owned” by the active instance
- Failure of the active queue manager (or host server) breaks client connections
  - IBM MQ initiates failover to standby queue manager
  - Must point client connections to new server (IP address of queue manager changes)
  - IBM MQ can automatically reconnect the client; can also use intelligent routers or other means
- When failover occurs, administrator must start another standby instance (or risk outage if failover server fails)

Figure 12-9. Multi-instance IBM MQ queue managers WM6461.0

Notes:

The multi-instance queue manager capability provides a simple means of failover. Only IBM MQ and Integration Bus can take advantage of this capability.



WebSphere Education IBM.

High availability with IBM MQ clusters

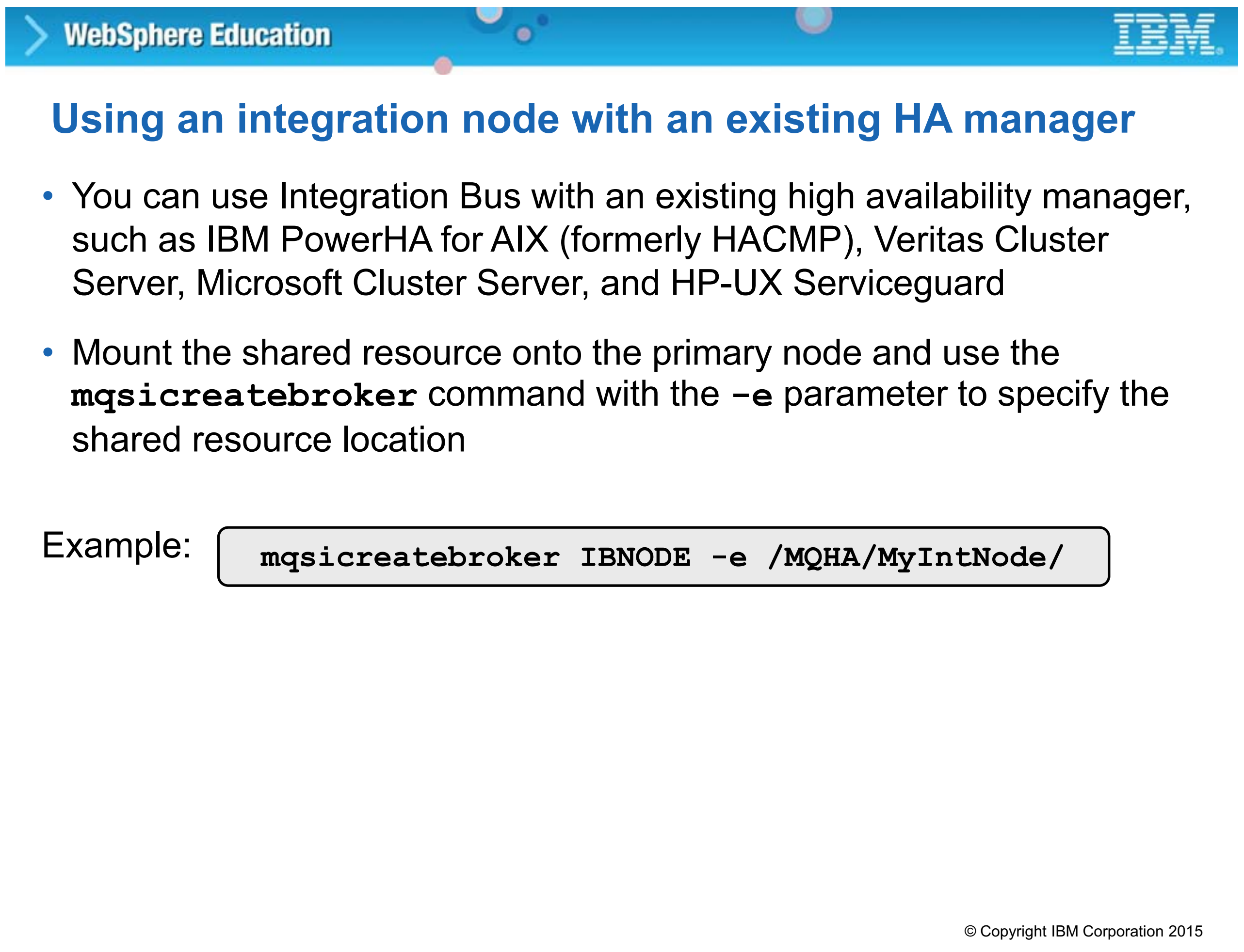
- Can coordinate resources beyond IBM MQ and IBM Integration Bus
- Can involve more than two servers
- Can fail over more than one time without administrator intervention
- IP address takeover occurs as part of failover processing
- Uses switchable shared storage (not network storage)
  - Disk is switched to fail over server automatically by HA software
- Can set queue manager to have its own "service" address
  - Used by HA software to switch servers during failover and then restart the queue manager
  - Client connections do not change

© Copyright IBM Corporation 2015

Figure 12-10. High availability with IBM MQ clusters WM6461.0

Notes:

High availability queue manager clusters rely on third-party software to monitor for failure events and then initiate failover. It is a more sophisticated approach than multi-instance queue managers.



WebSphere Education

Using an integration node with an existing HA manager

- You can use Integration Bus with an existing high availability manager, such as IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, and HP-UX Serviceguard
- Mount the shared resource onto the primary node and use the **`mqsicreatebroker`** command with the **`-e`** parameter to specify the shared resource location

Example:

```
mqsicreatebroker IBNODE -e /MQHA/MyIntNode/
```

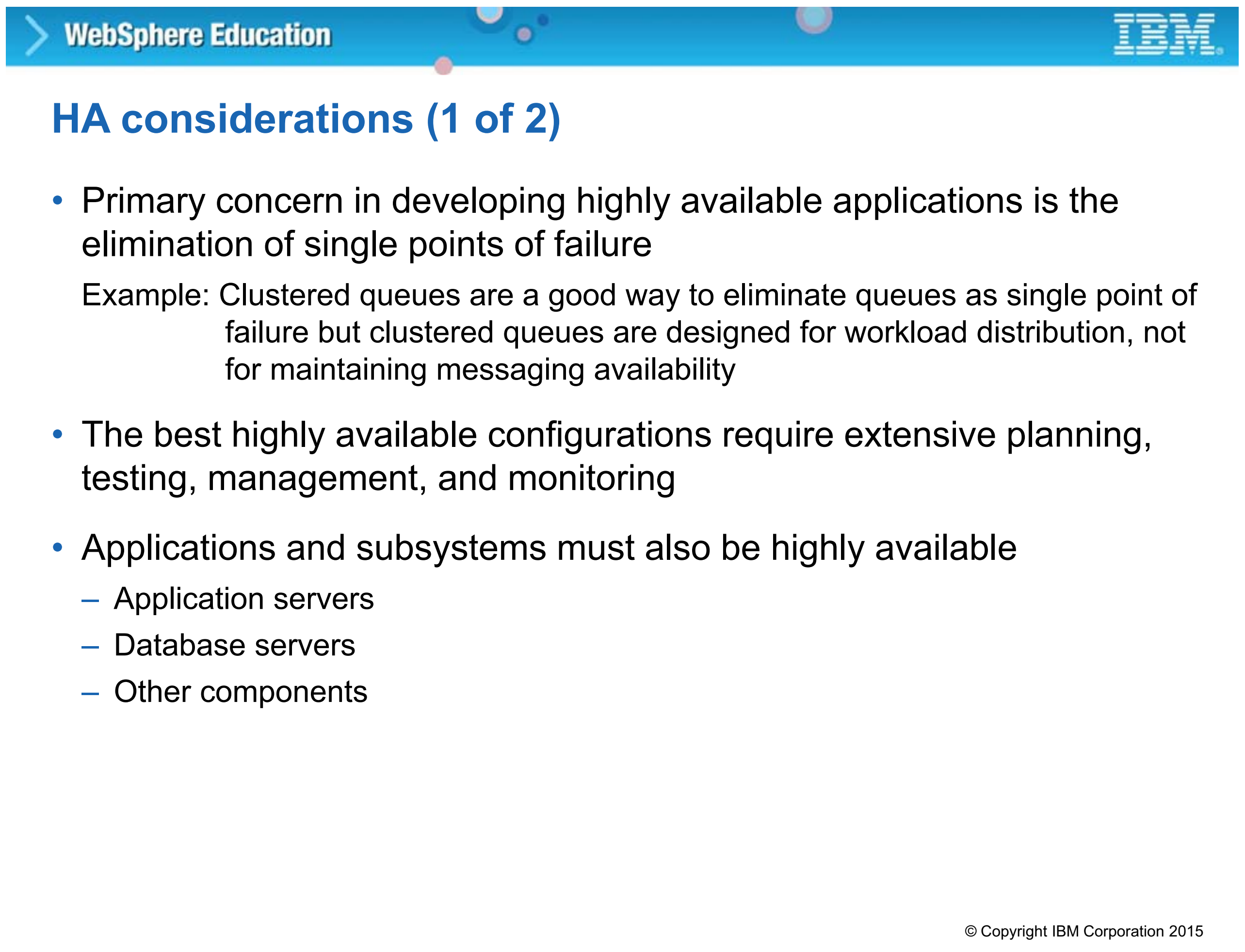
© Copyright IBM Corporation 2015

Figure 12-11. Using an integration node with an existing HA manager WM6461.0

Notes:

You can use Integration Bus with an existing high availability manager such as IBM PowerHA for AIX.

After you mount the shared resource onto the primary node, use the `mqsicreatebroker` command with the `-e` parameter to specify the shared resource location.



WebSphere Education

HA considerations (1 of 2)

- Primary concern in developing highly available applications is the elimination of single points of failure

  Example: Clustered queues are a good way to eliminate queues as single point of failure but clustered queues are designed for workload distribution, not for maintaining messaging availability

- The best highly available configurations require extensive planning, testing, management, and monitoring

- Applications and subsystems must also be highly available
  - Application servers
  - Database servers
  - Other components

© Copyright IBM Corporation 2015

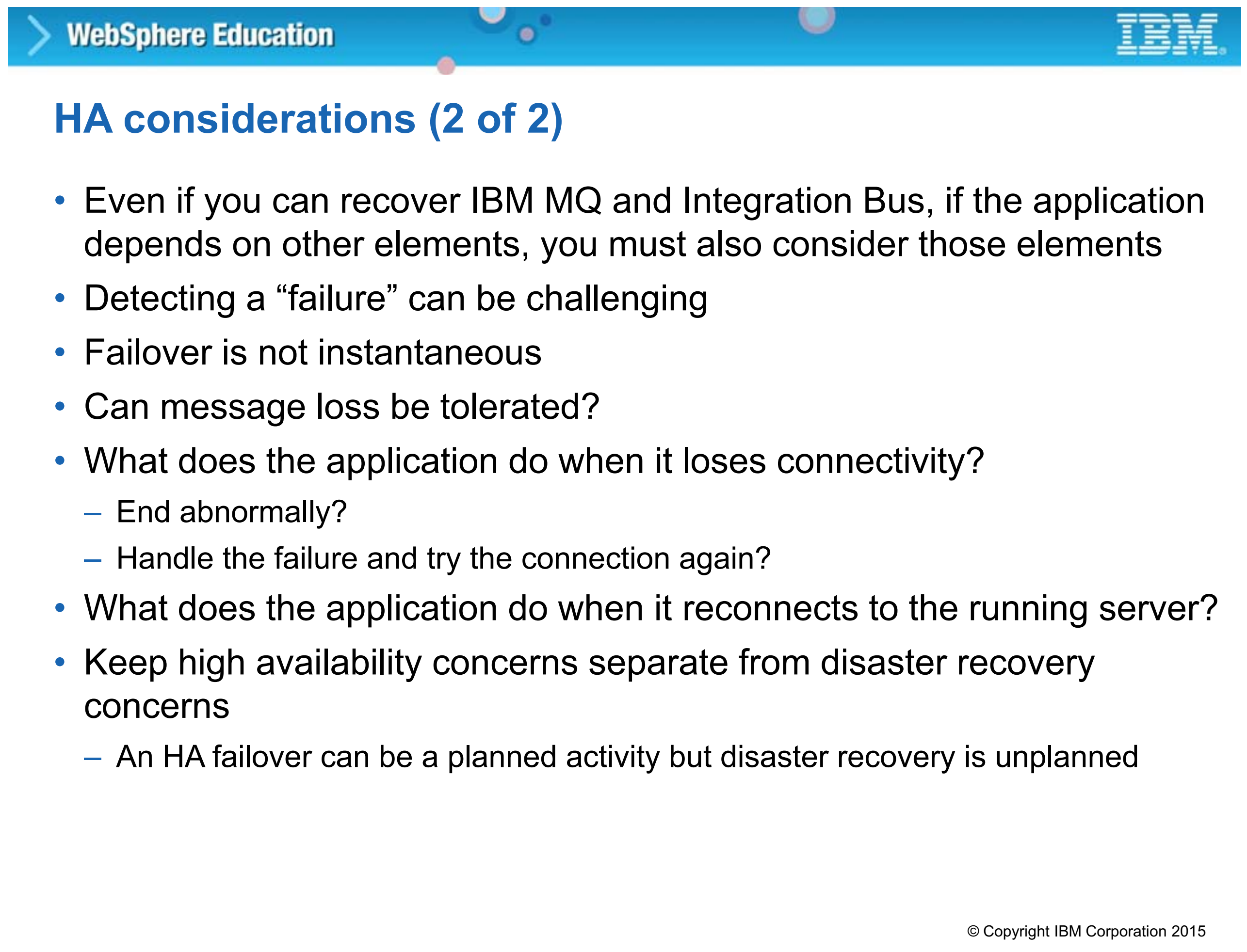
Figure 12-12. HA considerations (1 of 2) WM6461.0

Notes:

This figure and the next list considerations of high availability processing.

The most important aspect of high availability is eliminating all single points of failure. Any component whose failure can trigger an outage should be redesigned or replaced.

Also, as mentioned previously, even though IBM MQ and Integration Bus have high availability capabilities, you must consider other elements of the application design. If IBM MQ and Integration Bus can fail over but an application database is not available, for example, an outage still occurs.



WebSphere Education

HA considerations (2 of 2)

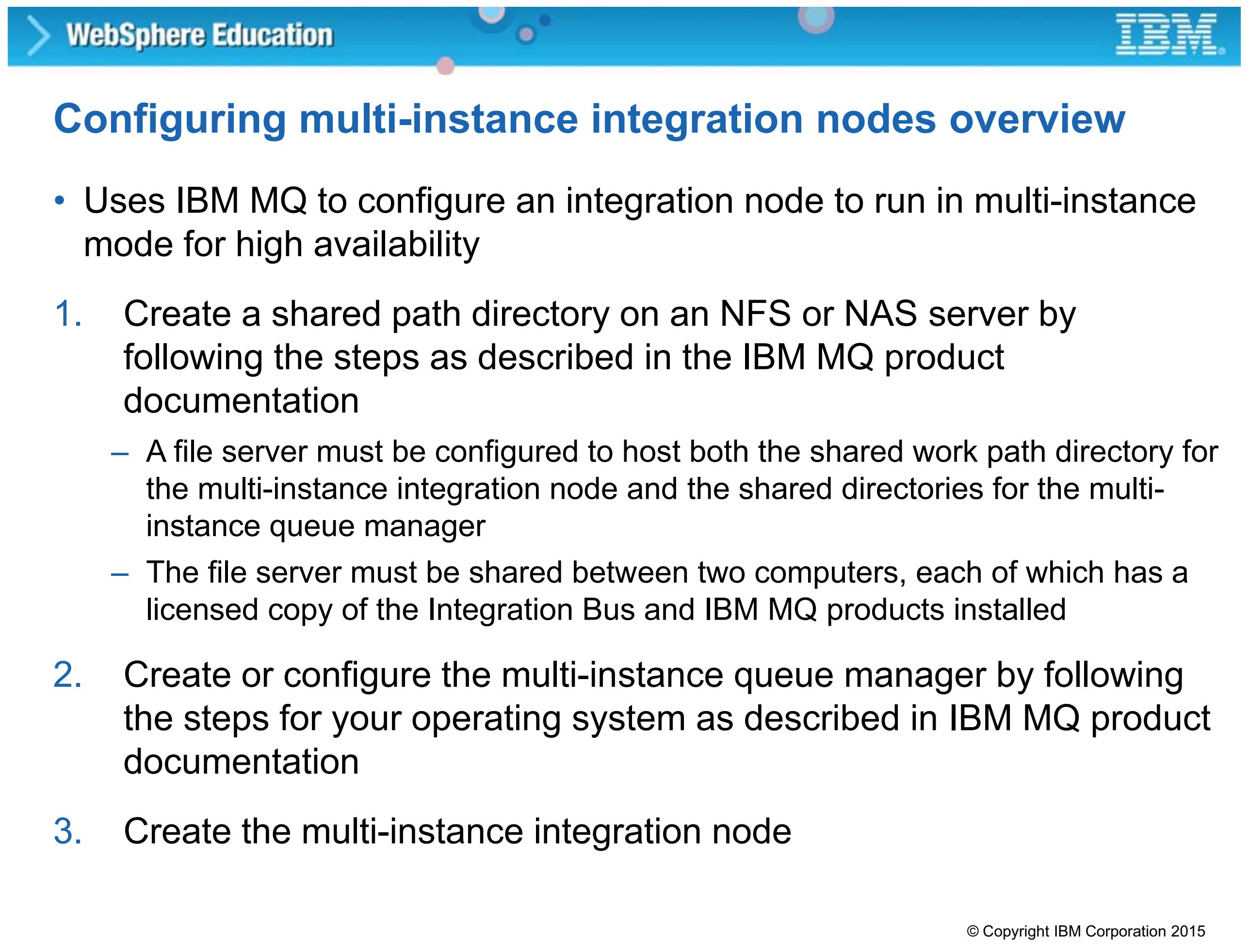
- Even if you can recover IBM MQ and Integration Bus, if the application depends on other elements, you must also consider those elements
- Detecting a “failure” can be challenging
- Failover is not instantaneous
- Can message loss be tolerated?
- What does the application do when it loses connectivity?
  - End abnormally?
  - Handle the failure and try the connection again?
- What does the application do when it reconnects to the running server?
- Keep high availability concerns separate from disaster recovery concerns
  - An HA failover can be a planned activity but disaster recovery is unplanned

Figure 12-13. HA considerations (2 of 2) WM6461.0

Notes:

Here are some other elements of high availability processing to consider. As you can see, a fully robust failover solution involves every aspect of an application.

12.2.Multi-instance integration nodes



WebSphere Education

Configuring multi-instance integration nodes overview

- Uses IBM MQ to configure an integration node to run in multi-instance mode for high availability

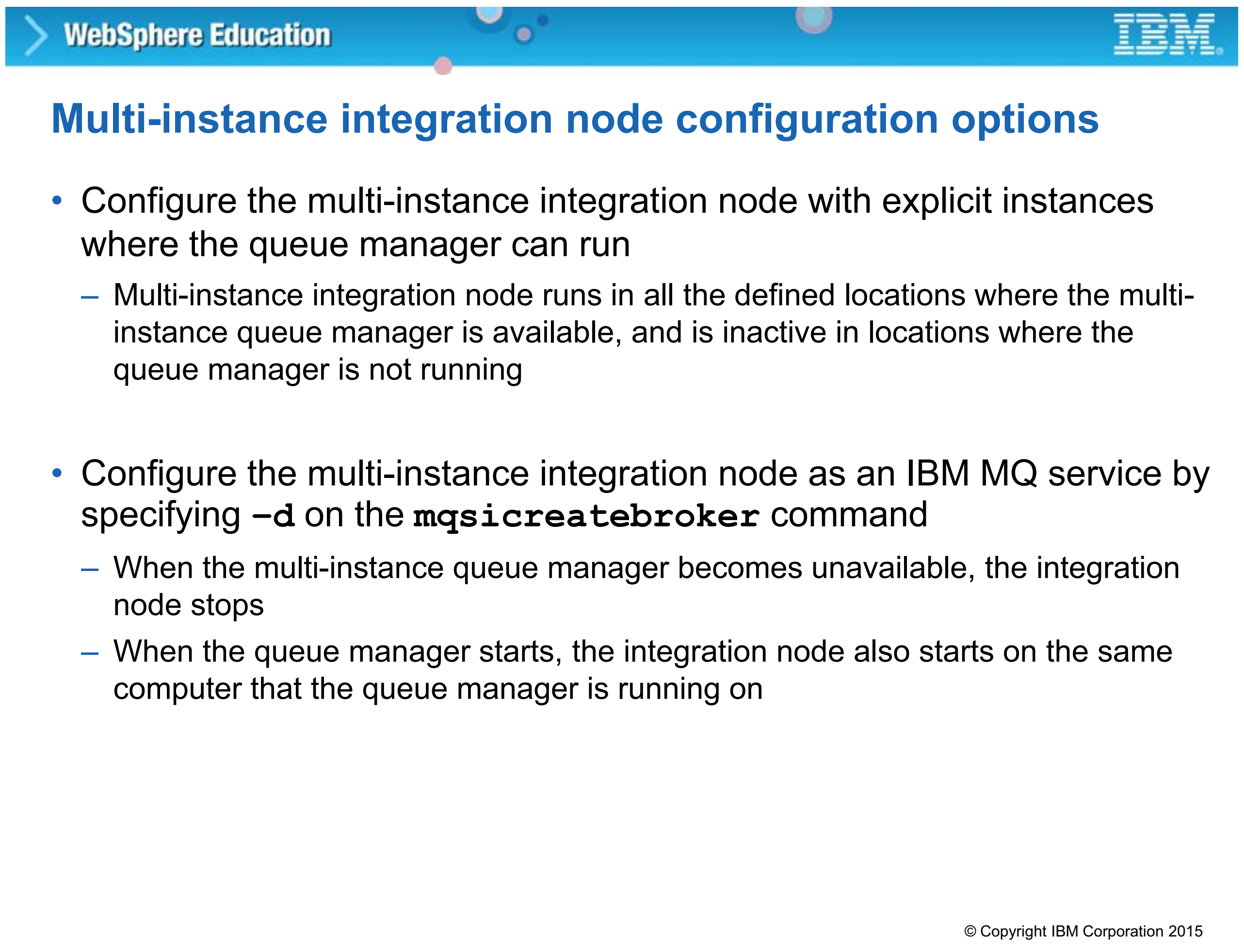
1. Create a shared path directory on an NFS or NAS server by following the steps as described in the IBM MQ product documentation
   – A file server must be configured to host both the shared work path directory for the multi-instance integration node and the shared directories for the multi-instance queue manager
   – The file server must be shared between two computers, each of which has a licensed copy of the Integration Bus and IBM MQ products installed
2. Create or configure the multi-instance queue manager by following the steps for your operating system as described in IBM MQ product documentation
3. Create the multi-instance integration node

Figure 12-14. Configuring multi-instance integration nodes overview WM6461.0

Notes:

Integration Bus uses IBM MQ to configure an integration node to run in multi-instance mode for high availability. Before you begin, review the topic “Multi-instance queue managers” in the IBM MQ product documentation to understand the use of multi-instance queue managers.

The figure summarizes the steps for configuring multi-instance integration nodes.



WebSphere Education

Multi-instance integration node configuration options

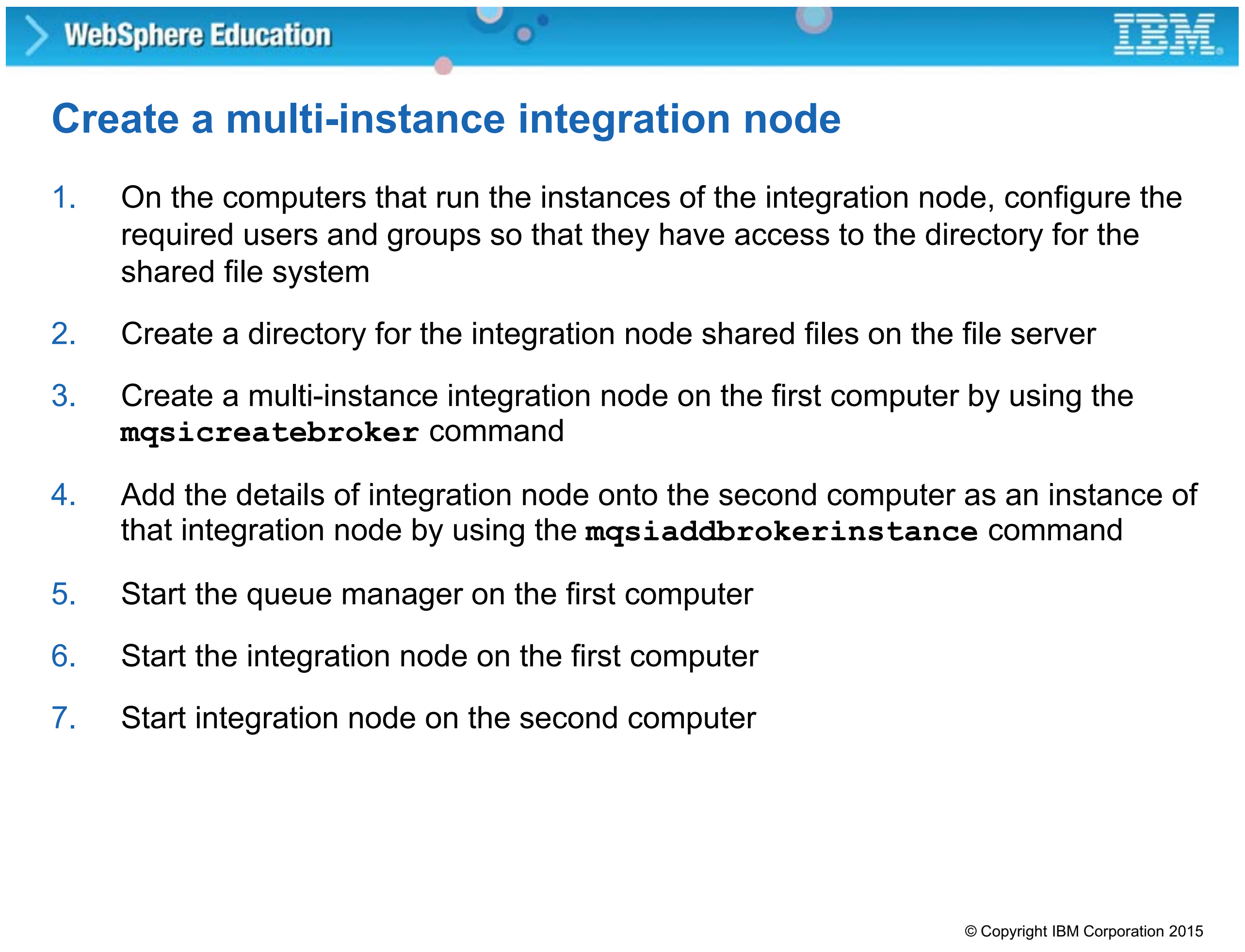
- Configure the multi-instance integration node with explicit instances where the queue manager can run
  - Multi-instance integration node runs in all the defined locations where the multi-instance queue manager is available, and is inactive in locations where the queue manager is not running
- Configure the multi-instance integration node as an IBM MQ service by specifying **`-d`** on the **`mqsicreatebroker`** command
  - When the multi-instance queue manager becomes unavailable, the integration node stops
  - When the queue manager starts, the integration node also starts on the same computer that the queue manager is running on

Figure 12-15. Multi-instance integration node configuration options WM6461.0

Notes:

There are two options for configuring multi-instance integration nodes.

- Option 1: Configure the multi-instance integration node with explicit instances where the queue manager can run.
- Option 2: Configure the multi-instance integration node as an IBM MQ service.



WebSphere Education

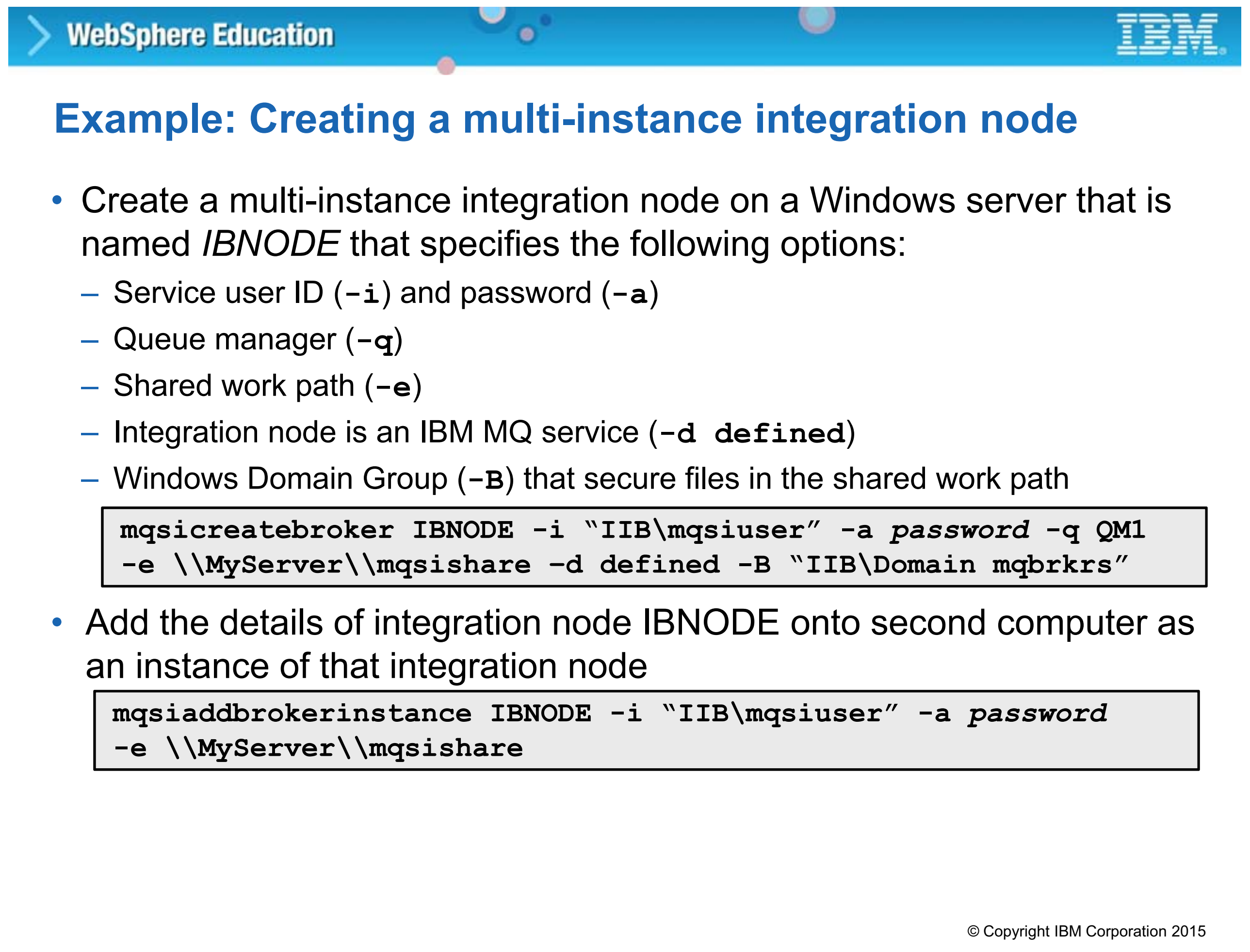
Create a multi-instance integration node

1. On the computers that run the instances of the integration node, configure the required users and groups so that they have access to the directory for the shared file system
2. Create a directory for the integration node shared files on the file server
3. Create a multi-instance integration node on the first computer by using the **`mqsicreatebroker`** command
4. Add the details of integration node onto the second computer as an instance of that integration node by using the **`mqsiaddbrokerinstance`** command
5. Start the queue manager on the first computer
6. Start the integration node on the first computer
7. Start integration node on the second computer

Figure 12-16. Create a multi-instance integration node WM6461.0

Notes:

This figure lists the steps for creating a multi-instance integration node.



Example: Creating a multi-instance integration node

- Create a multi-instance integration node on a Windows server that is named *IBNODE* that specifies the following options:
  - Service user ID (**`-i`**) and password (**`-a`**)
  - Queue manager (**`-q`**)
  - Shared work path (**`-e`**)
  - Integration node is an IBM MQ service (**`-d defined`**)
  - Windows Domain Group (**`-B`**) that secure files in the shared work path

```
mqsicreatebroker IBNODE -i “IIB\mqsiuser” -a password -q QM1
-e \\MyServer\\mqsishare –d defined -B “IIB\Domain mqbrkrs”
```

- Add the details of integration node IBNODE onto second computer as an instance of that integration node

```
mqsiaddbrokerinstance IBNODE -i “IIB\mqsiuser” -a password
-e \\MyServer\\mqsishare
```

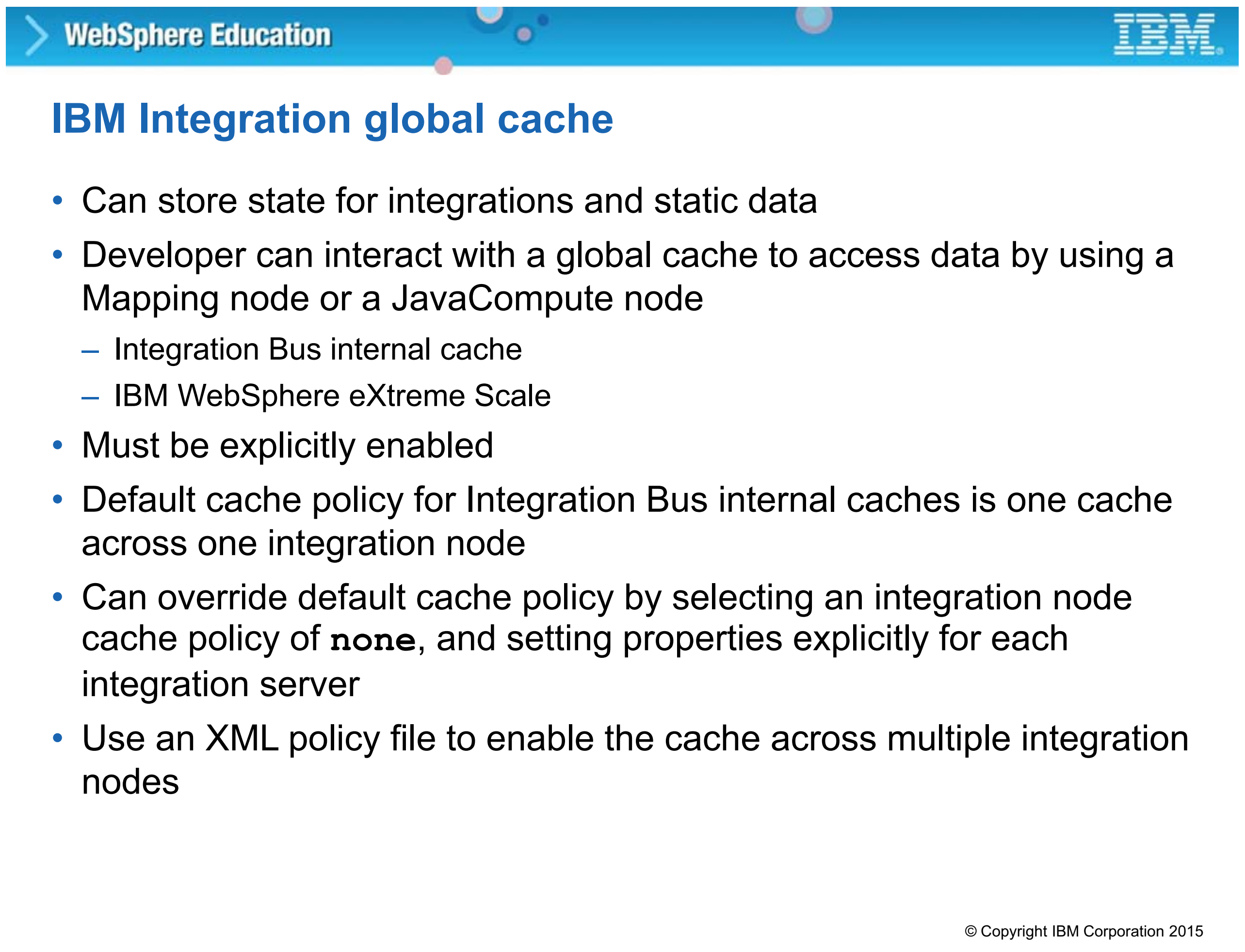
Figure 12-17. Example: Creating a multi-instance integration node WM6461.0

Notes:

This figure provides an example of how to create a multi-instance integration node on a Windows server by using the `mqsicreatebroker` command. The configuration requires a service user ID, IBM MQ queue manager, and a shared data location.

In the example, the first command creates the multi-instance integration node on the primary server. The second command creates an instance of the integration node on the secondary server.

12.3.Global caching



WebSphere Education

IBM Integration global cache

- Can store state for integrations and static data
- Developer can interact with a global cache to access data by using a Mapping node or a JavaCompute node
  - Integration Bus internal cache
  - IBM WebSphere eXtreme Scale
- Must be explicitly enabled
- Default cache policy for Integration Bus internal caches is one cache across one integration node
- Can override default cache policy by selecting an integration node cache policy of `none`, and setting properties explicitly for each integration server
- Use an XML policy file to enable the cache across multiple integration nodes

Figure 12-18. IBM Integration global cache WM6461.0

Notes:

You can use the global cache that is supplied with Integration Bus to store data that you want to reuse.

The global cache is embedded in the integration node. By default, the cache is not enabled; to use the cache, select an appropriate cache policy.

The default cache policy creates a default topology of cache components in a single integration node. The alternatives to the default topology are to have no policy. If you do not have a policy, you can control your own topology by setting cache properties on the integration servers. Or use an XML policy file to enable the cache across multiple integration nodes.

Scenario 1: Storing state for integrations

- When Integration Bus integrates asynchronous systems, the integration node records information about the requester to correlate the replies correctly
- With a global cache, each integration node can handle replies, even when another integration node processes the request

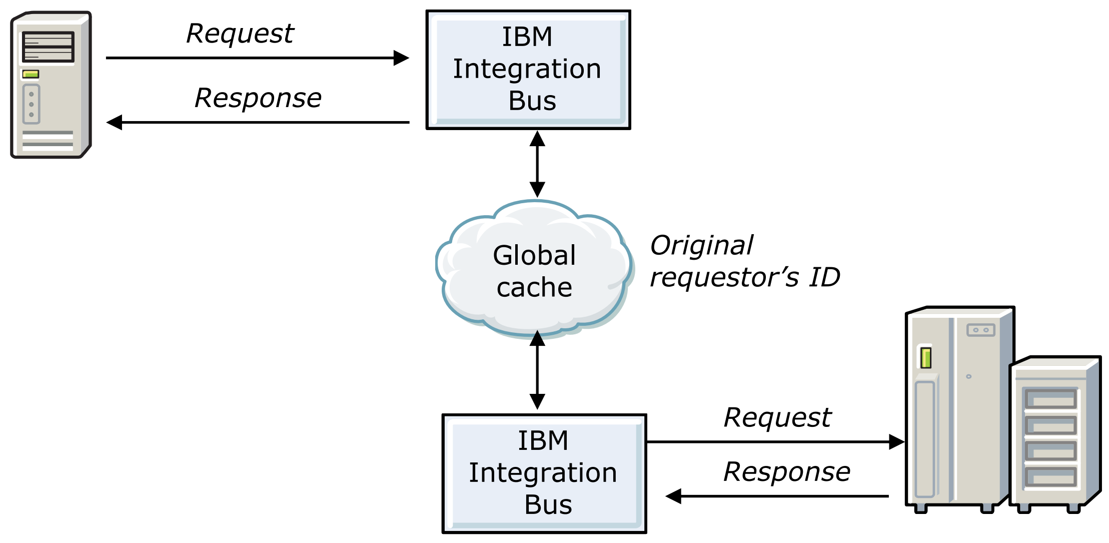


Figure 12-19. Scenario 1: Storing state for integrations WM6461.0

Notes:

In a request and reply scenario, the developer typically needs to keep the reply-to information so that it can be accessed when the message is returned to the original requester.

A global cache allows integration nodes to share this message information in memory. Before the implementation of a global cache, the problem that is described in this scenario was solved by storing the requester information in an external database.

WebSphere Education IBM

Scenario 2: Caching static data

- Use of a global cache facilitates horizontal scaling in situations where a cache is used to minimize network interactions to a back end system
  - Increase the number of clients while maintaining a predictable response time for each client

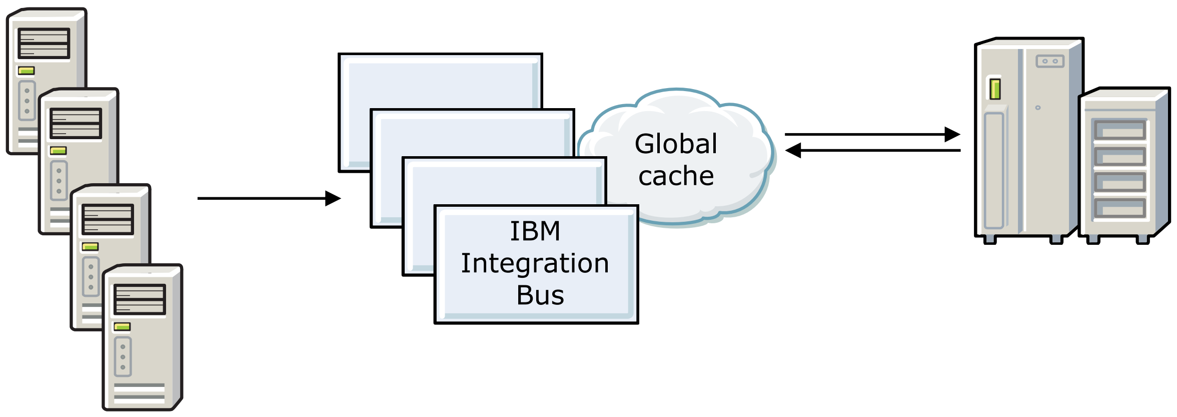
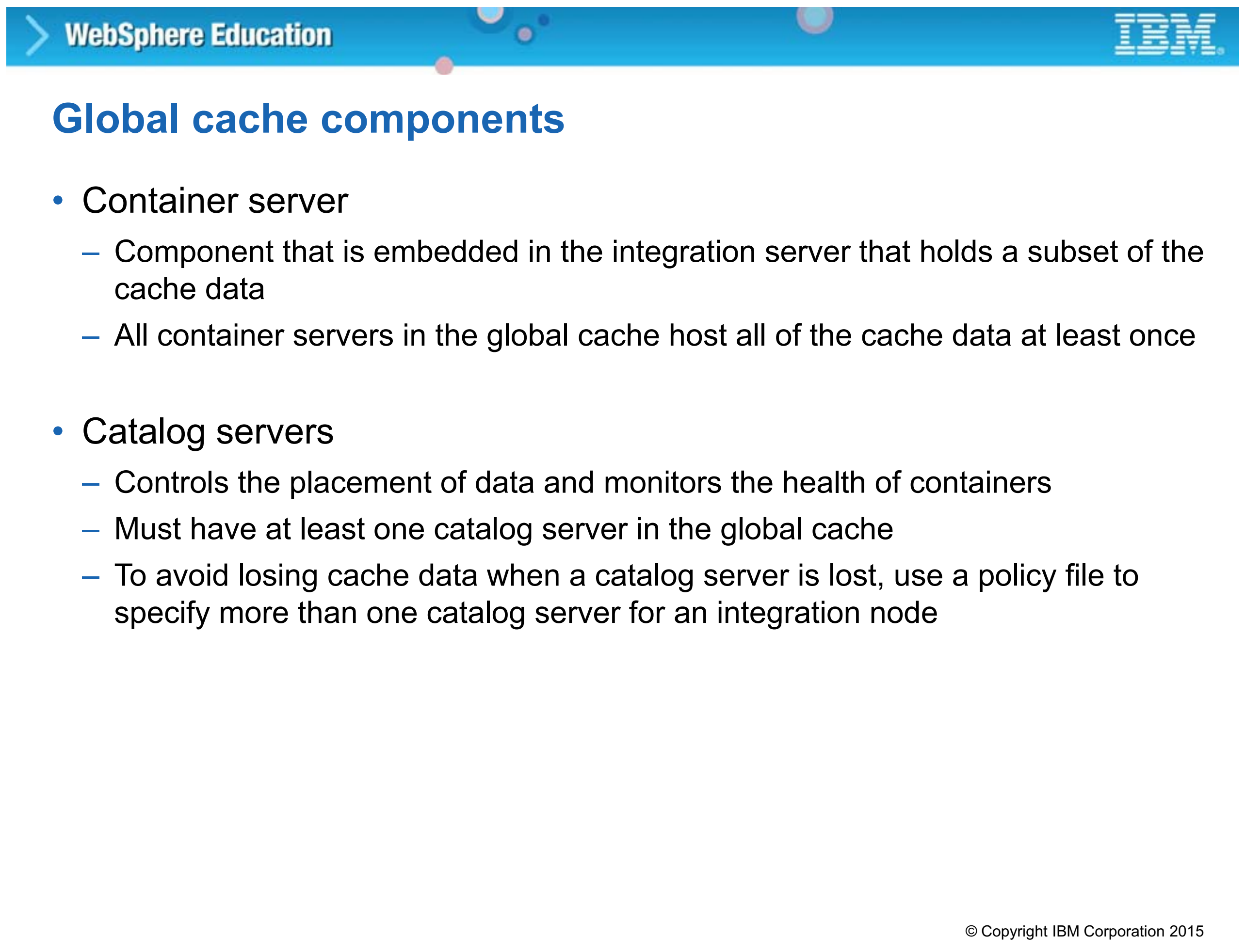


Figure 12-20. Scenario 2: Caching static data WM6461.0

Notes:

In this scenario, a global cache helps to maintain a predictable response time for each client by caching infrequently changing data.



WebSphere Education

Global cache components

- Container server
  - Component that is embedded in the integration server that holds a subset of the cache data
  - All container servers in the global cache host all of the cache data at least once
- Catalog servers
  - Controls the placement of data and monitors the health of containers
  - Must have at least one catalog server in the global cache
  - To avoid losing cache data when a catalog server is lost, use a policy file to specify more than one catalog server for an integration node

Figure 12-21. Global cache components WM6461.0

Notes:

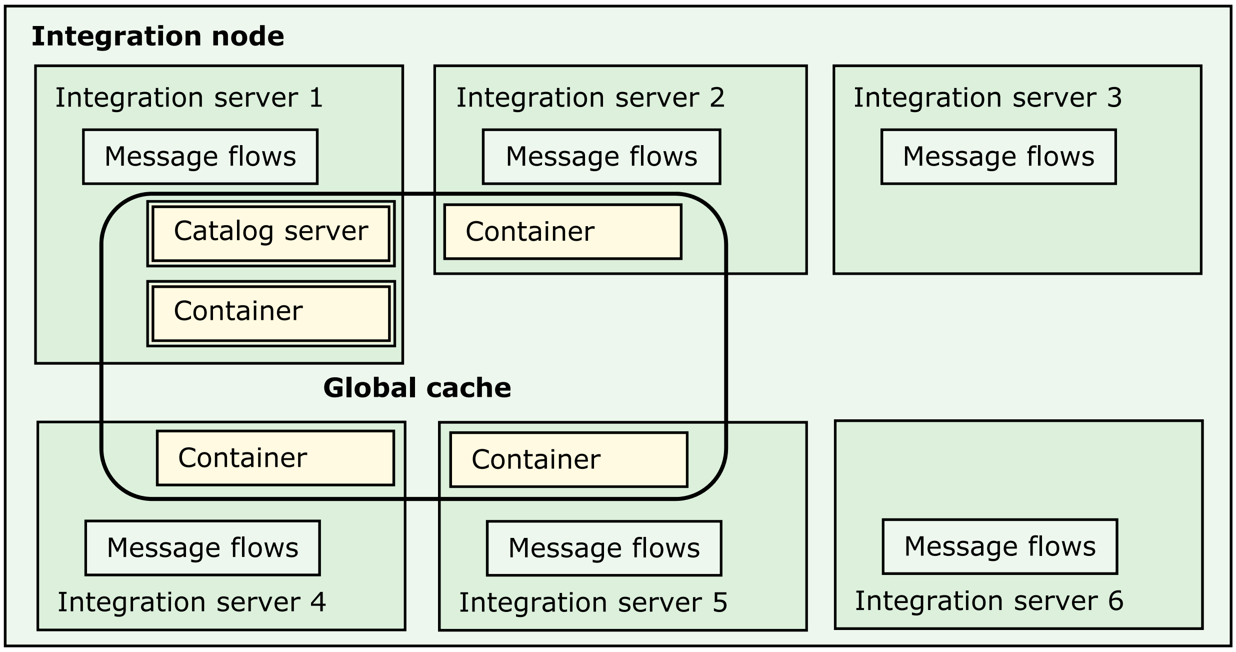
Two components that use the global cache are the container server and the catalog server.

A *container server* is a component that is embedded in the integration server that holds a subset of the cache data. Between them, all container servers in the global cache host all of the cache data at least one time. If more than one container exists, the default cache policy ensures that all data is replicated at least one time. In this way, the global cache can cope with the loss of container servers without losing data.

The *catalog server* controls placement of data and monitors the health of containers. You must have at least one catalog server in your global cache. To avoid losing cache data when a catalog server is lost, use a policy file to specify more than one catalog server for an integration node. For example, if you specify two catalog servers for a single integration node, if one catalog server fails, the integration node switches to the other catalog server.

WebSphere Education IBM

Integration Bus default cache topology



Cache components that are using the default policy are hosted in an integration node with six integration servers

Figure 12-22. Integration Bus default cache topology WM6461.0

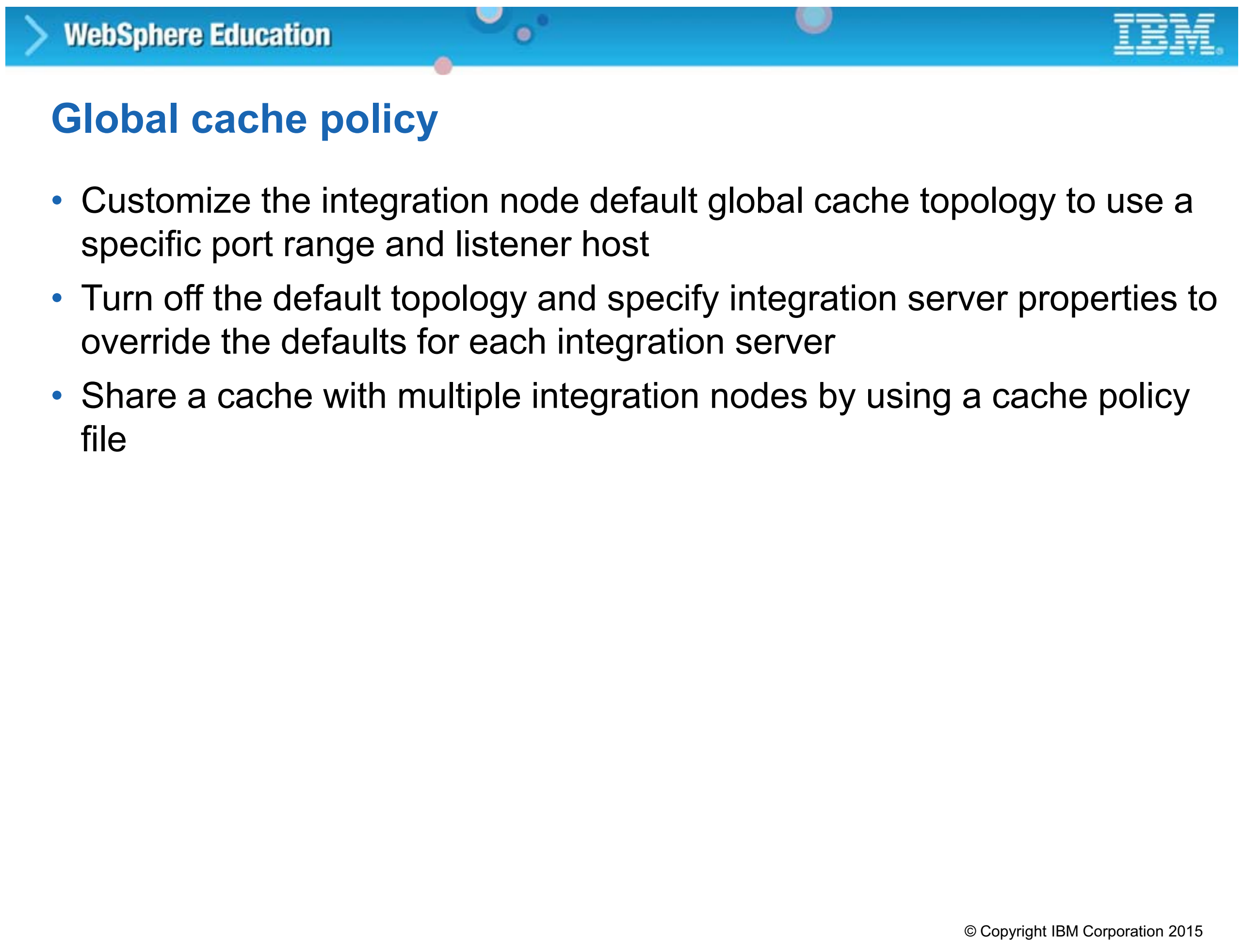
Notes:

The figure shows the Integration Bus default cache topology. By default, one integration server in the integration node hosts a *catalog server*. The catalog server controls placement of data and monitors the health of container servers.

Up to three other integration servers in that integration node can host *container servers*. A container server is a component that is embedded in the integration server that holds a subset of the cache data.

The catalog server and container servers are placed in integration servers dynamically when the integration node starts. All integration servers can communicate with the global cache, regardless of whether they are hosting catalog servers, container servers, or neither.

Each integration server contains a *cache manager*, which manages the cache components that are embedded in that integration server.



Global cache policy

- Customize the integration node default global cache topology to use a specific port range and listener host
- Turn off the default topology and specify integration server properties to override the defaults for each integration server
- Share a cache with multiple integration nodes by using a cache policy file

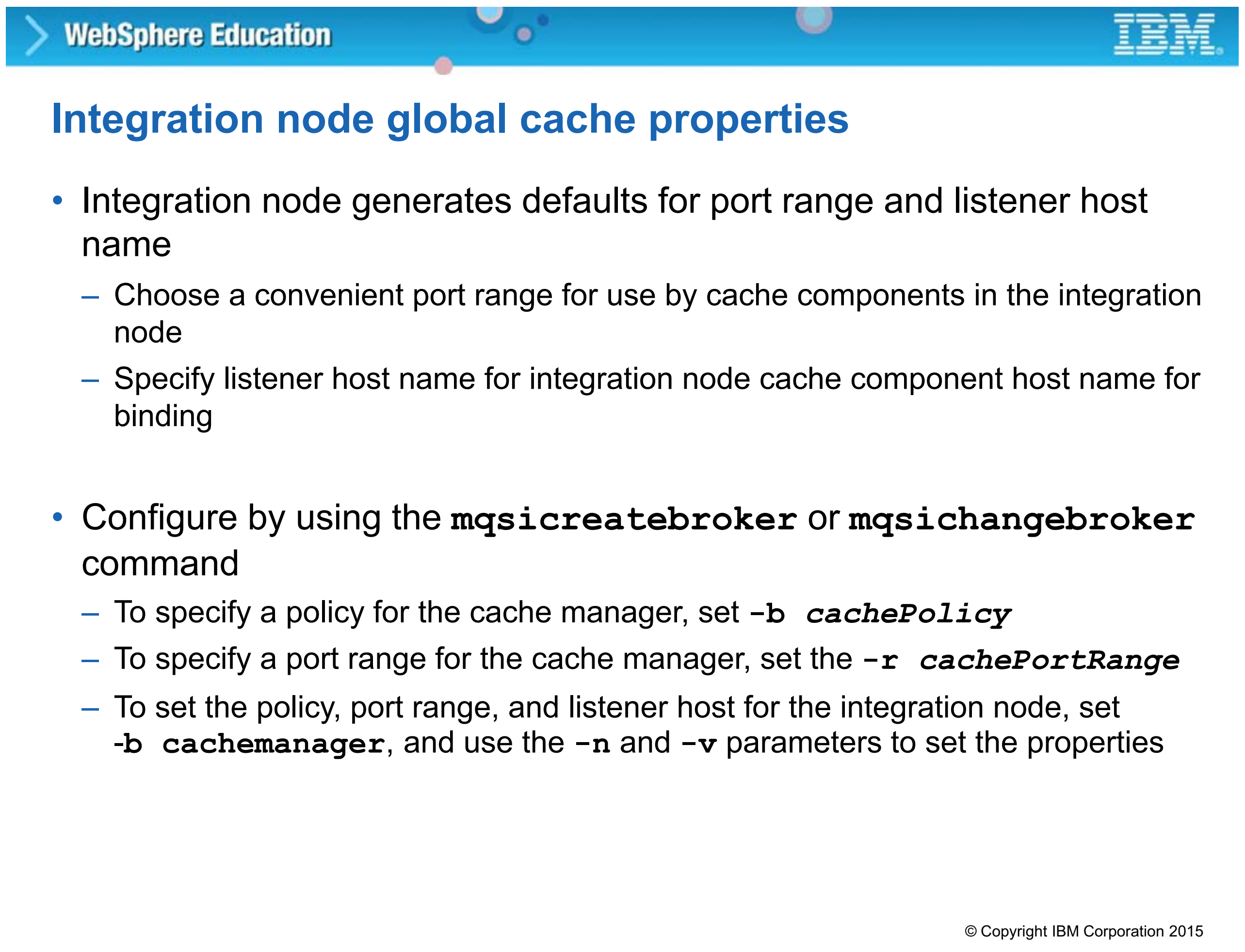
Figure 12-23. Global cache policy WM6461.0

Notes:

You can configure properties of the embedded global cache by using commands, an XML cache policy file, or the IBM Integration API.

You can customize the default global cache topology to use a specific port range and listener host. You can turn off the default topology and specify your own integration server properties.

Multiple integration nodes can share a cache by using a cache policy file.



WebSphere Education
IBM

Integration node global cache properties

- Integration node generates defaults for port range and listener host name
  - Choose a convenient port range for use by cache components in the integration node
  - Specify listener host name for integration node cache component host name for binding

- Configure by using the **`mqsicreatebroker`** or **`mqsichangebroker`** command
  - To specify a policy for the cache manager, set **`-b`** ***`cachePolicy`***
  - To specify a port range for the cache manager, set the **`-r`** ***`cachePortRange`***
  - To set the policy, port range, and listener host for the integration node, set **`-b cachemanager`**, and use the **`-n`** and **`-v`** parameters to set the properties

Figure 12-24. Integration node global cache properties WM6461.0

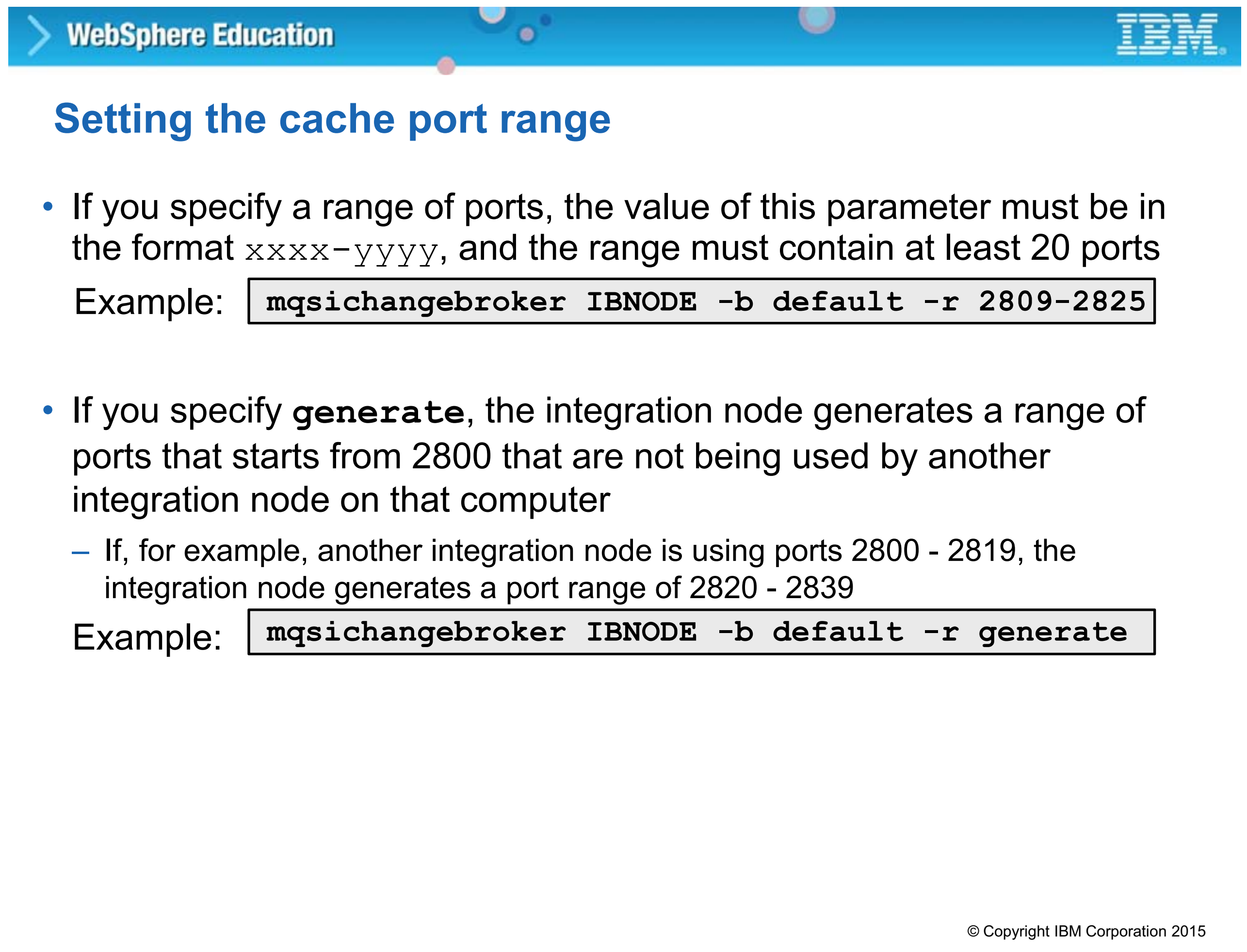
Notes:

You can set cache manager properties at integration node level and integration server level by using the `mqsicreatebroker` or `mqsichangebroker` command.

You can set the cache policy (`-b`) to none, `default`, `disabled`, or the fully qualified name of an XML policy file.

- If you set this property to `default`, the default global cache topology is used.
- If you set this property to none, you must set the integration server properties explicitly.
- If you set this property to `disabled`, all cache components in the integration node are not enabled. The cache is not enabled by default.
- If you specify the fully qualified name of a policy file, the integration nodes that are listed in the policy file are configured to share the data in the global cache. The path must be absolute, not relative.

Set the listener host name or IP address that the cache components use to listen. Explicitly set the listener host name if you have more than one network card available.



Setting the cache port range

- If you specify a range of ports, the value of this parameter must be in the format `xxxx-yyyy`, and the range must contain at least 20 ports

  Example:
  ```
  mqsichangebroker IBNODE -b default -r 2809-2825
  ```

- If you specify **`generate`**, the integration node generates a range of ports that starts from 2800 that are not being used by another integration node on that computer
  - If, for example, another integration node is using ports 2800 - 2819, the integration node generates a port range of 2820 - 2839

  Example:
  ```
  mqsichangebroker IBNODE -b default -r generate
  ```

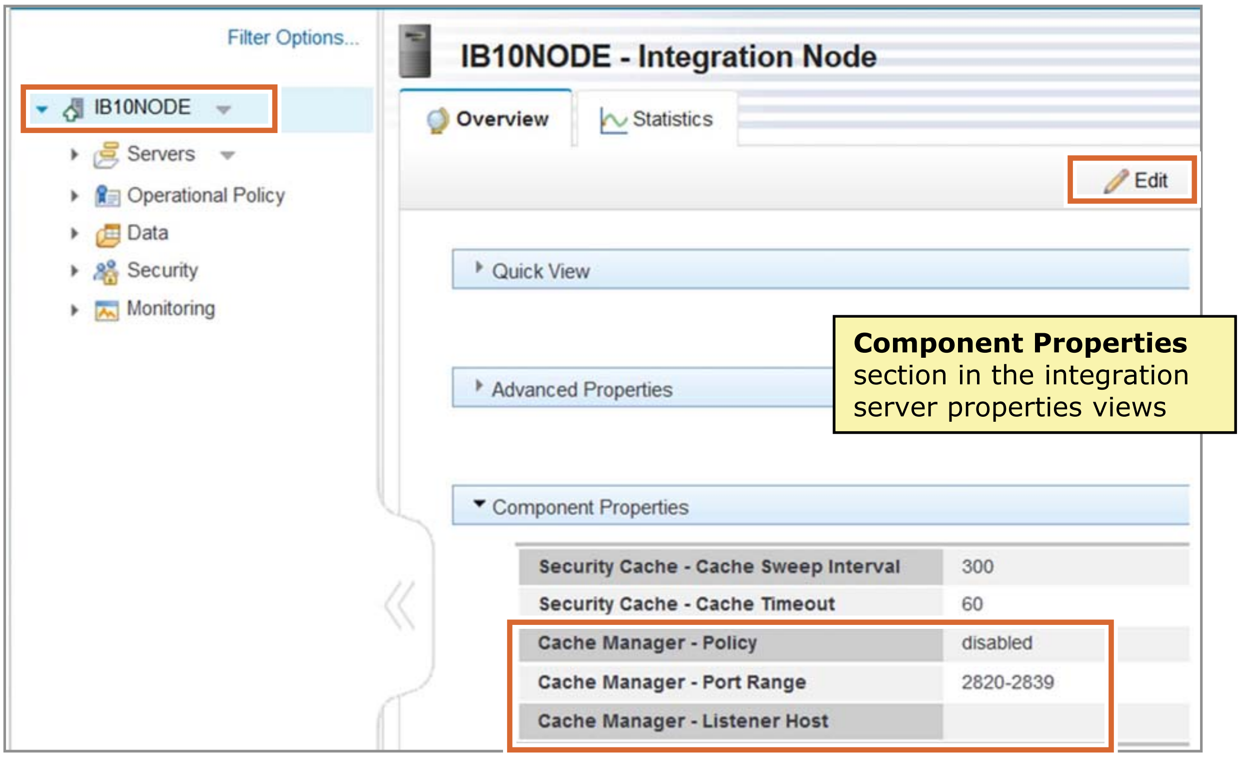
Figure 12-25. Setting the cache port range WM6461.0

Notes:

Set the port range (`-r`) to generate or to a specific range of ports. If you specify a range of ports, the value of this parameter must be in the format xxxx-yyyy, and the range must contain at least 20 ports.

WebSphere Education IBM

Integration node cache properties in the Integration web interface



Component Properties section in the integration server properties views

Security Cache - Cache Sweep Interval	300
Security Cache - Cache Timeout	60
Cache Manager - Policy	disabled
Cache Manager - Port Range	2820-2839
Cache Manager - Listener Host	

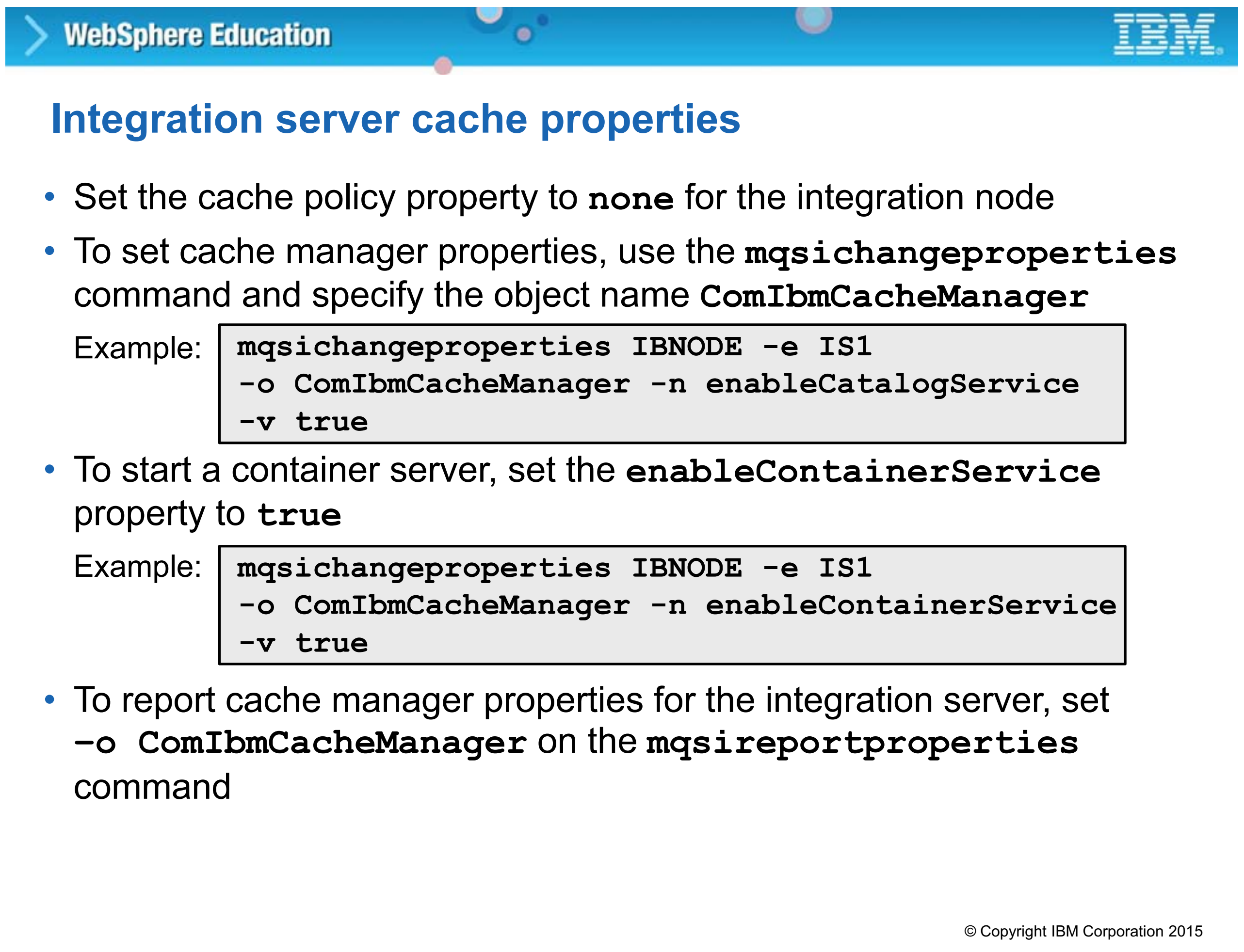
© Copyright IBM Corporation 2015

Figure 12-26. Integration node cache properties in the Integration web interface WM6461.0

Notes:

You can set the global cache **Cache Manager** properties by using the Integration web interface.

1. Expand the **Component Properties** section on the integration node **Overview** tab.
2. Click **Edit**.
3. Modify the **Cache Manager** properties and then click **Save**.



Integration server cache properties

- Set the cache policy property to **`none`** for the integration node
- To set cache manager properties, use the **`mqsichangeproperties`** command and specify the object name **`ComIbmCacheManager`**

  Example:
  ```
  mqsichangeproperties IBNODE -e IS1
  -o ComIbmCacheManager -n enableCatalogService
  -v true
  ```
- To start a container server, set the **`enableContainerService`** property to **`true`**

  Example:
  ```
  mqsichangeproperties IBNODE -e IS1
  -o ComIbmCacheManager -n enableContainerService
  -v true
  ```
- To report cache manager properties for the integration server, set **`–o ComIbmCacheManager`** on the **`mqsireportproperties`** command

Figure 12-27. Integration server cache properties WM6461.0

Notes:

You can set the integration server global cache properties by using the `mqsichangeproperties` command or the Integration web interface.

You must restart the integration server for all changes to be implemented. If you stop the integration server that contains the catalog server, the global cache is not available.

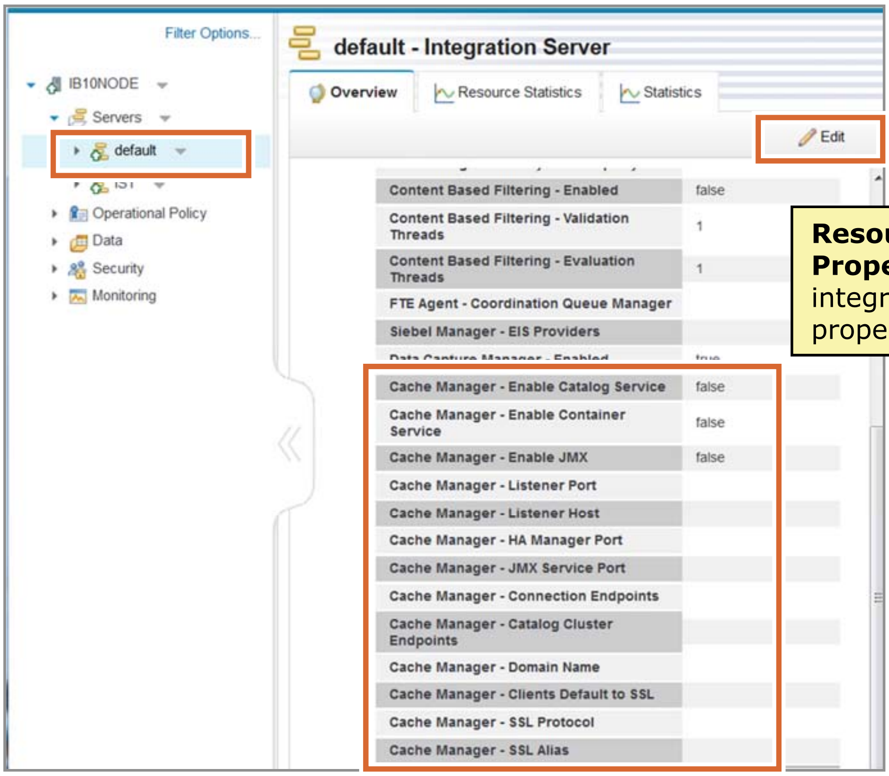
The global cache properties for the integration server include the following properties:

- `enableCatalogService` determines whether the named integration server hosts a catalog server.
- `enableContainerService` determines whether the named integration server hosts a container server.

For a description of all the properties, see the “Parameter values for the cache manager component” topic in the IBM Knowledge Center.

WebSphere Education

Integration server cache properties in the Integration web interface



Content Based Filtering - Enabled	false
Content Based Filtering - Validation Threads	1
Content Based Filtering - Evaluation Threads	1
FTE Agent - Coordination Queue Manager	
Siebel Manager - EIS Providers	
Data Capture Manager - Enabled	true
Cache Manager - Enable Catalog Service	false
Cache Manager - Enable Container Service	false
Cache Manager - Enable JMX	false
Cache Manager - Listener Port	
Cache Manager - Listener Host	
Cache Manager - HA Manager Port	
Cache Manager - JMX Service Port	
Cache Manager - Connection Endpoints	
Cache Manager - Catalog Cluster Endpoints	
Cache Manager - Domain Name	
Cache Manager - Clients Default to SSL	
Cache Manager - SSL Protocol	
Cache Manager - SSL Alias	

Resource Manager Properties sections in the integration server properties view

Figure 12-28. Integration server cache properties in the Integration web interface WM6461.0

Notes:

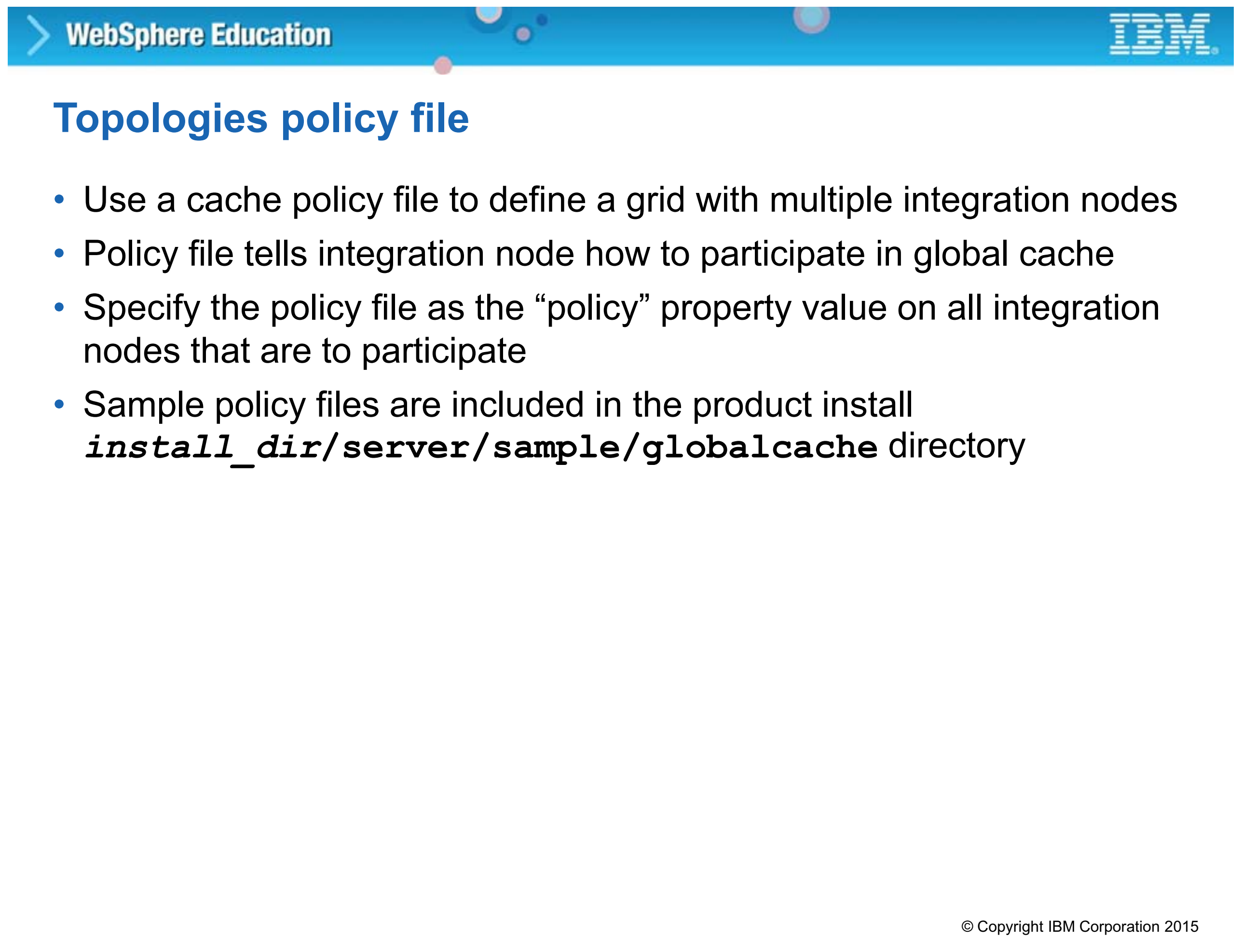
You can set the integration server global cache properties in the IBM Integration web interface. You must restart the integration server for all changes to be implemented. If you stop the integration server that contains the catalog server, the global cache is not available.

The **Cache Manager** properties for the integration server include the following properties:

- **Enable Catalog Service** determines whether the named integration server hosts a catalog server.
- **Enable Container Service** determines whether the named integration server hosts a container server.
- **Enable JMX** determines whether the named integration server can provide administration information about the grid.
- **Listener Port** is the value of the listener port if either the catalog server or container server is enabled. The value must be unique.
- **Listener Host** is the host name of the local computer. If this parameter is not set, the integration node uses as the default the first host name that it identifies for this computer. Set this parameter if you have more than one network card.

- **HA Manager Port** is required if either the catalog server or container server is enabled. This value must be unique.
- **JMX Service Port** is required if the **Enable JMX** property is set to true.
- **Connection Endpoints** is a string that comprises connection endpoints for container servers to connect to catalog servers, and for client connections to the embedded grid. The format of this value is a comma-separated list of endpoints. Each endpoint has the format `listenerHost:listenerPort`.

For a description of all the properties, see the IBM Knowledge Center for IBM Integration Bus V10.



WebSphere Education

Topologies policy file

- Use a cache policy file to define a grid with multiple integration nodes
- Policy file tells integration node how to participate in global cache
- Specify the policy file as the “policy” property value on all integration nodes that are to participate
- Sample policy files are included in the product install ***install_dir*/server/sample/globalcache** directory

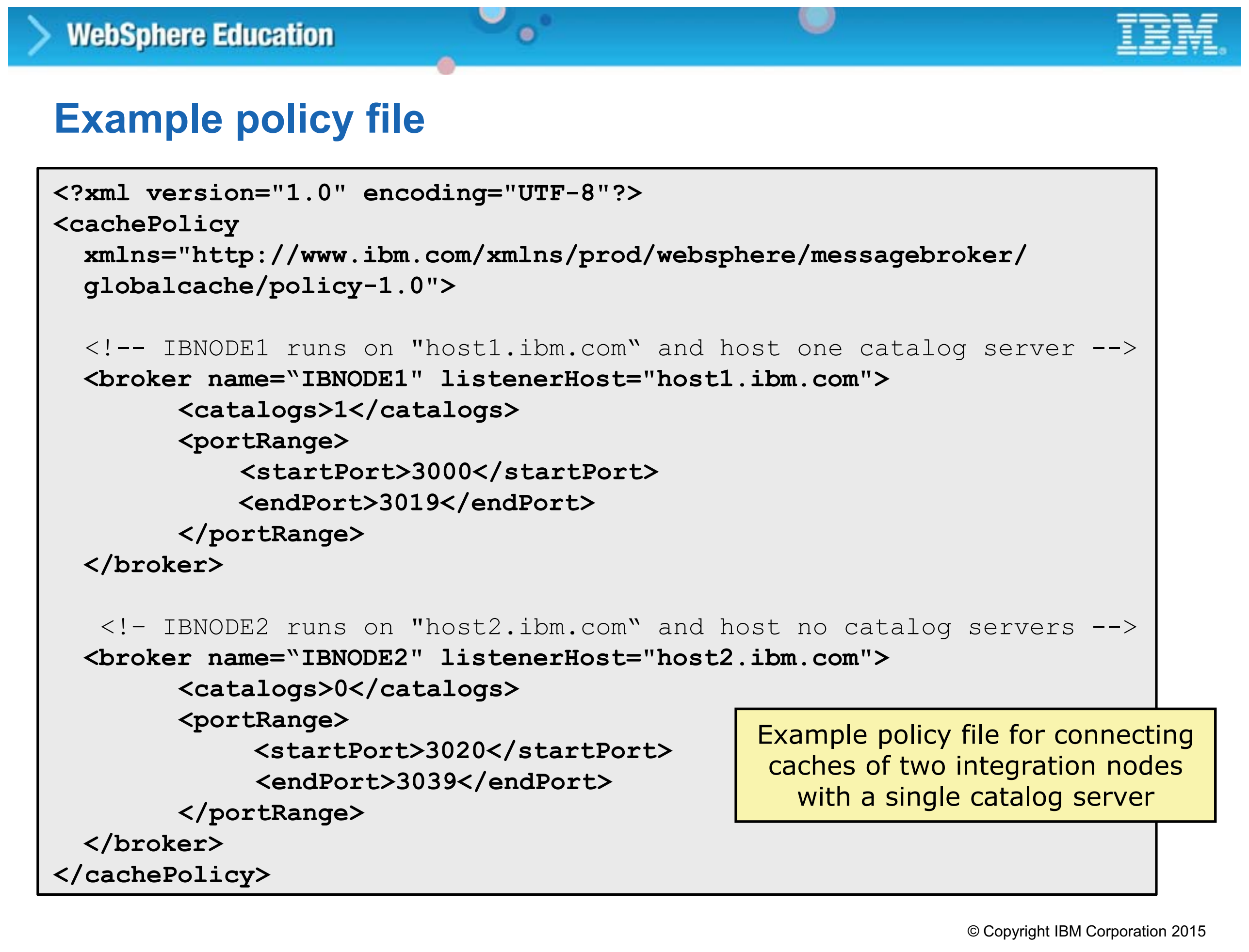
© Copyright IBM Corporation 2015

Figure 12-29. Topologies policy file WM6461.0

Notes:

You can configure the global cache to span multiple integration nodes by providing an XML policy file. This policy file lists the integration nodes that share the cache, and for each integration node specifies the listener host, port range, and the number of hosted catalog servers.

Specify a policy file for your integration nodes to use by setting the cache policy to the name and file path of the policy file. Sample policy files are provided in the IBM Integration Bus `sample\globalcache` directory on Windows.



Example policy file

```
<?xml version="1.0" encoding="UTF-8"?>
<cachePolicy
  xmlns="http://www.ibm.com/xmlns/prod/websphere/messagebroker/
  globalcache/policy-1.0">

  <!-- IBNODE1 runs on "host1.ibm.com‶ and host one catalog server -->
  <broker name=‶IBNODE1" listenerHost="host1.ibm.com">
        <catalogs>1</catalogs>
        <portRange>
            <startPort>3000</startPort>
            <endPort>3019</endPort>
        </portRange>
  </broker>

   <!- IBNODE2 runs on "host2.ibm.com‶ and host no catalog servers -->
  <broker name=‶IBNODE2" listenerHost="host2.ibm.com">
        <catalogs>0</catalogs>
        <portRange>
             <startPort>3020</startPort>
             <endPort>3039</endPort>
        </portRange>
  </broker>
</cachePolicy>
```

WebSphere Education
IBM
Example policy file for connecting caches of two integration nodes with a single catalog server

Figure 12-30. Example policy file WM6461.0

Notes:

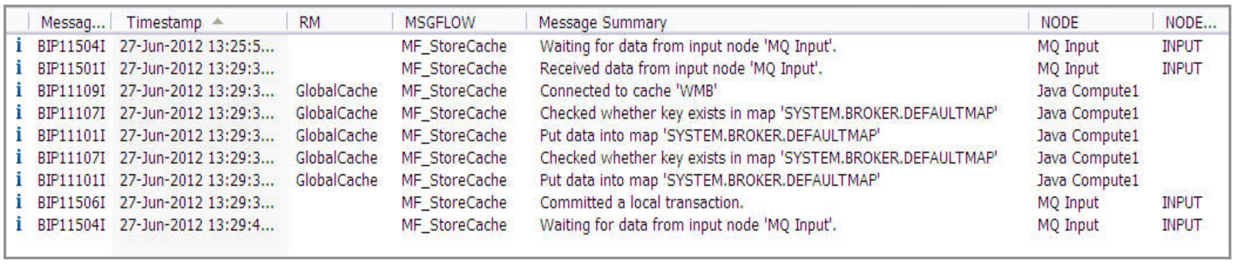
As shown in the figure, the global cache policy file specifies the integration node names and listener hosts, the port range that the integration node uses, and how many catalog servers that the integration node hosts. Optionally, you can also specify a domain name for all catalog servers in the embedded cache.

Ensure that the policy meets the following criteria:

- You can define 0, 1, or 2 catalog servers for an individual integration node, but at least one catalog server must be defined in the policy.
- If two integration nodes share a host name, you must set a distinct port range for each integration node.
- Ensure that the port range for each integration node includes at least 20 ports.
- The integration node names and listener hosts that are specified in the policy must match the values that are defined for the integration nodes.
- The policy file must be encoded in UTF-8 and contain valid XML.

WebSphere Education IBM

Global cache administrative tools



Messag...	Timestamp	RM	MSGFLOW	Message Summary	NODE	NODE...
BIP11504I	27-Jun-2012 13:25:5...		MF_StoreCache	Waiting for data from input node 'MQ Input'.	MQ Input	INPUT
BIP11501I	27-Jun-2012 13:29:3...		MF_StoreCache	Received data from input node 'MQ Input'.	MQ Input	INPUT
BIP11109I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Connected to cache 'WMB'	Java Compute1	
BIP11107I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Checked whether key exists in map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
BIP11101I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Put data into map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
BIP11107I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Checked whether key exists in map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
BIP11101I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Put data into map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
BIP11506I	27-Jun-2012 13:29:3...		MF_StoreCache	Committed a local transaction.	MQ Input	INPUT
BIP11504I	27-Jun-2012 13:29:4...		MF_StoreCache	Waiting for data from input node 'MQ Input'.	MQ Input	INPUT

- Full resource statistics and activity log in the Integration web interface **Administration Log** provide information about the state of the cache and cache interactions
- **`mqsicacheadmin`** command
  – Provides advanced information about the underlying grid
  – Validates that all specified integration nodes are participating in a grid with multiple integration nodes
  – Checks that the data is distributed evenly in underlying cache elements
  – Use with the **`-c showMapSizes`** option to show the size of embedded cache
  – Use with the **`-c clearGrid -m`** ***`mapname`*** option to clear data from cache

Figure 12-31. Global cache administrative tools WM6461.0

Notes:

You can use the **`mqsicacheadmin`** command, resource statistics, and the activity log to monitor the global cache. You can use resource statistics and the activity log to monitor external grids.

Activity logs provide a high-level overview of how Integration Bus interacts with external resources. The logs help you to understand what your message flows are doing.

An integration node collects resource statistics to record performance and operating details of resources that integration servers use.

The **`mqsicacheadmin`** command provides information about the global cache that is embedded in an integration node. For example, you can find out the size of a map, list the hosts that are participating in the cache, and clear data from a map.

The **`-c`** option runs the command that follows it against the embedded cache.

- **`clearGrid`** clears all data from the specified map in the embedded cache.
- **`showMapSizes`** displays the sizes of all maps in the embedded cache.
- **`placementServiceStatus`** displays the status of the ObjectGrid placement operation.

- **`routetable`** displays the current routing table for all WebSphere eXtreme Scale shards and partitions in the embedded cache.
- **`showPlacement`** lists all container servers and their shards in the embedded cache.
- **`listHosts`** lists all hosts that are participating in the embedded cache.

WebSphere Education

Connectivity to WebSphere eXtreme Scale grids

- Can connect to external WebSphere eXtreme Scale grid
- Connections are configured by using the **WXSServer** configurable service
- Connect to multiple external grids, and the embedded global cache at the same time
- Interactions with external grids are logged in Integration Bus activity log and resource statistics

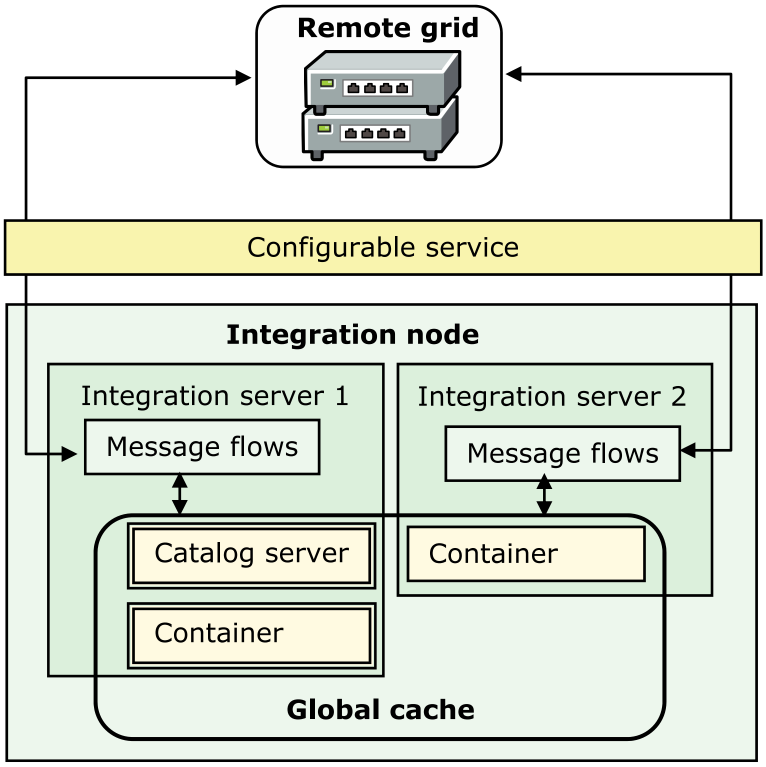


Figure 12-32. Connectivity to WebSphere eXtreme Scale grids WM6461.0

Notes:

WebSphere eXtreme Scale provides a scalable, in-memory data grid. The data grid dynamically caches, partitions, replicates, and manages data across multiple servers.

As an option, you can use one or more external WebSphere eXtreme Scale grids to store data that you want to reuse. In addition to the grid that is available (as the embedded global cache) within Integration Bus, you can integrate with WebSphere eXtreme Scale grids that are running elsewhere. You can work with multiple external grids, and the embedded grid, at the same time. The figure shows an example configuration.

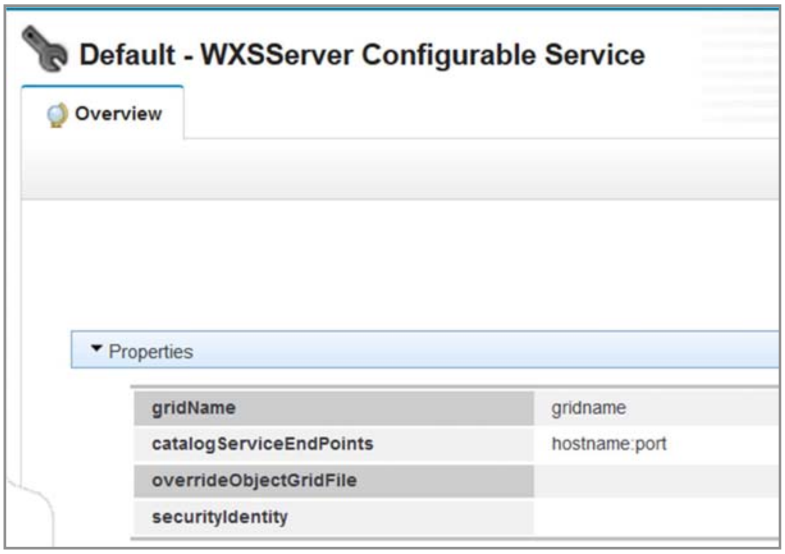
To connect to an external grid, you need the name of the grid, the host name, and port of each catalog server for the grid. By using a configurable service to specify the parameters, you can connect to an external WebSphere eXtreme Scale grid. Create a **WXSServer** configurable service by using the **`mqsicreateconfigurableservice`** command or in the Integration web interface.

Statistics are available for all activity in the embedded global cache and external WebSphere eXtreme Scale grids. One line of statistics is shown for each configurable service that is used to connect to an external grid.

WebSphere Education IBM

WXSServer configurable service

- Specify the catalog endpoints for the external grid, and the grid name
- Optionally, if the grid requires a user ID and password authorization, create a security identity and refer to it in the configurable service
- Generates command equivalent that is based on security identity
- Optionally, point to a client override file



gridName	gridname
catalogServiceEndPoints	hostname:port
overrideObjectGridFile	
securityIdentity	

Figure 12-33. WXSServer configurable service WM6461.0

Notes:

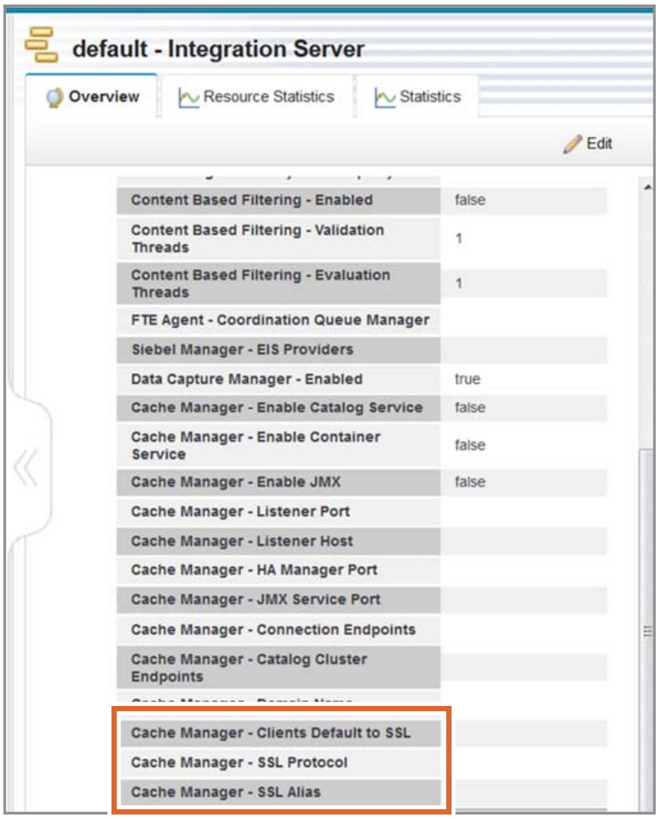
The WXSServer configurable service specifies the catalog endpoints for the external grid.

- **`catalogServiceEndPoints`** key is a comma-separated list of one or more catalog servers for the grid to which you are connecting. The format of each endpoint is `hostname:port`. This property is mandatory.
- **`gridName`** is the name of the grid to which you are connecting.
- **`overrideObjectGridFile`** is the absolute path to an XML file on the local system. This XML file defines overrides for the WebSphere eXtreme Scale client connections that are associated with this configurable service. This property is optional.
- **`securityIdentity`** is the name of the security identity to use when connecting to a secure grid. This property is optional.

WebSphere Education

SSL for external grids

- At the integration server level, specify whether the integration server should use SSL for client connections to WebSphere eXtreme Scale grids
- Optionally specify the SSL protocol, and SSL key alias, if more than one trusted key exists in the keystore
- If **Clients default to SSL** is enabled, all connections from this integration server attempt to use TLS
  - Connections can be made only from integration servers that do not host catalog or container servers



Content Based Filtering - Enabled	false
Content Based Filtering - Validation Threads	1
Content Based Filtering - Evaluation Threads	1
FTE Agent - Coordination Queue Manager	
Siebel Manager - EIS Providers	
Data Capture Manager - Enabled	true
Cache Manager - Enable Catalog Service	false
Cache Manager - Enable Container Service	false
Cache Manager - Enable JMX	false
Cache Manager - Listener Port	
Cache Manager - Listener Host	
Cache Manager - HA Manager Port	
Cache Manager - JMX Service Port	
Cache Manager - Connection Endpoints	
Cache Manager - Catalog Cluster Endpoints	
Cache Manager - Clients Default to SSL	
Cache Manager - SSL Protocol	
Cache Manager - SSL Alias	

Figure 12-34. SSL for external grids WM6461.0

Notes:

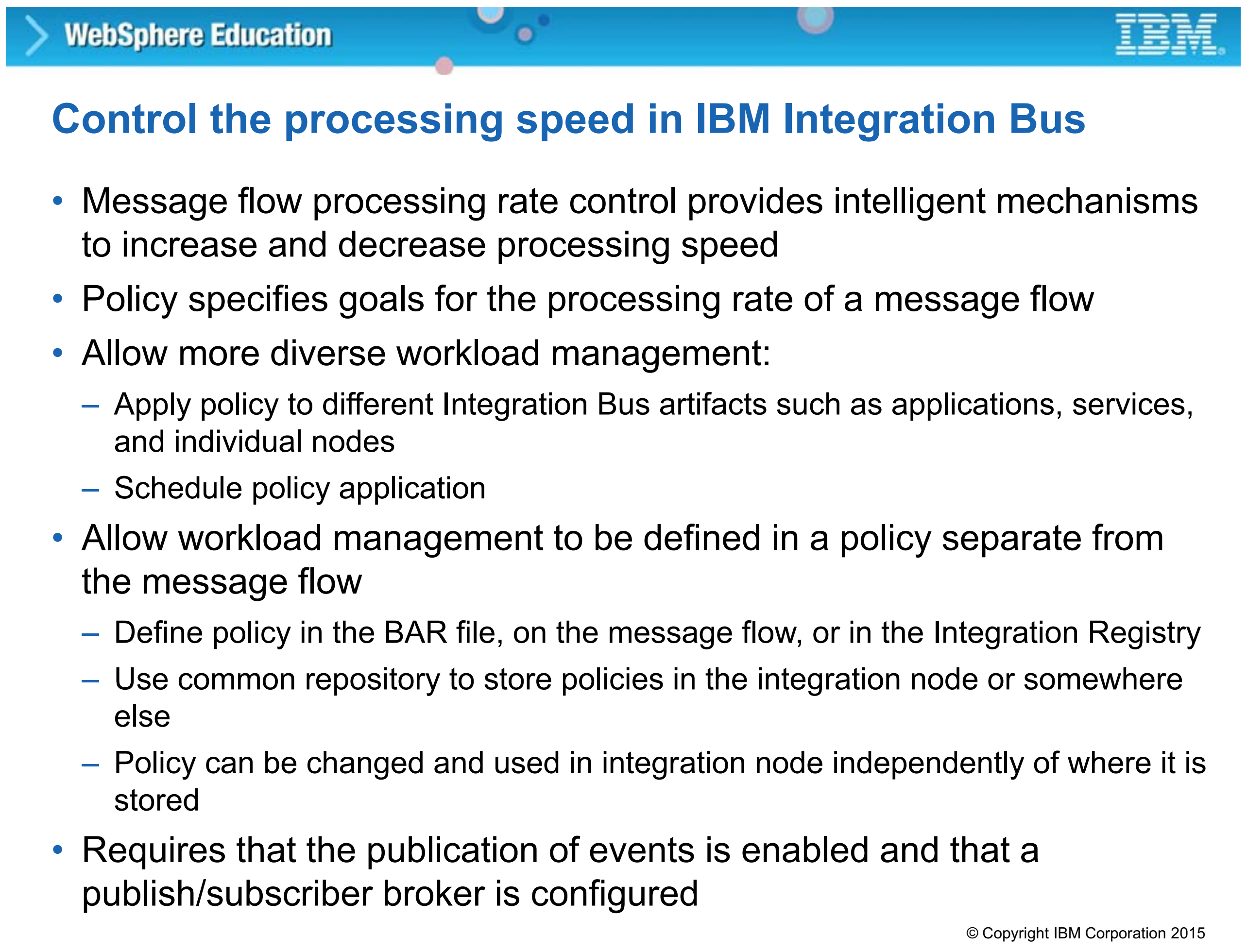
You can enable SSL for client connections to external WebSphere eXtreme Scale grids. To enable SSL communication, configure the keystore, truststore, passwords, and certificates.

To enable server authentication, import the public certificate from the WebSphere eXtreme Scale server into the integration node or integration server truststore. If the server requires client authentication, you must also create a private key in the integration node or integration server keystore that the WebSphere eXtreme Scale server trusts.

You then set properties on the integration server to enable SSL and specify the required protocol. To enable SSL in the IBM Integration web interface, set the **Cache Manager** properties on the integration server:

- To enable SSL, set **Clients Default to SSL** to `true`.
- To specify an SSL protocol, set **SSL Protocol** to a value that the IBM JSSE2 security provider recognizes.
- If the external grid requires client authentication and you have more than one trusted private key in the integration node keystore, set **SSL Alias** to the appropriate key.

12.4.Workload management



WebSphere Education

Control the processing speed in IBM Integration Bus

- Message flow processing rate control provides intelligent mechanisms to increase and decrease processing speed
- Policy specifies goals for the processing rate of a message flow
- Allow more diverse workload management:
  - Apply policy to different Integration Bus artifacts such as applications, services, and individual nodes
  - Schedule policy application
- Allow workload management to be defined in a policy separate from the message flow
  - Define policy in the BAR file, on the message flow, or in the Integration Registry
  - Use common repository to store policies in the integration node or somewhere else
  - Policy can be changed and used in integration node independently of where it is stored
- Requires that the publication of events is enabled and that a publish/subscriber broker is configured

Figure 12-35. Control the processing speed in IBM Integration Bus WM6461.0

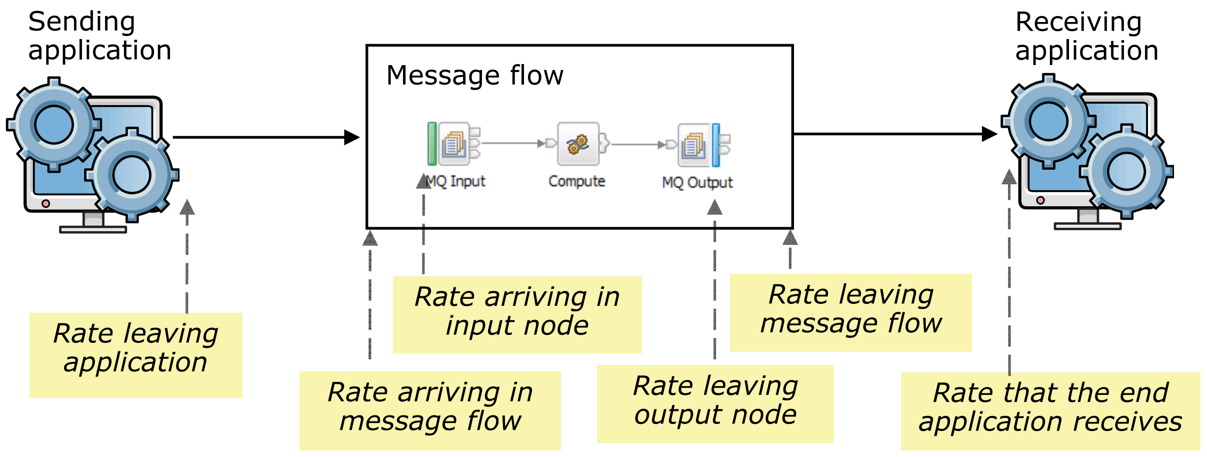
Notes:

Workload management allows system administrators to monitor and adjust the speed that messages are processes. It also controls the actions that are taken on unresponsive flows and threads.

The system administrator can specify a workload management policy within the Integration Registry for a message flow. The workload management policy encompasses all of the properties that are available under workload management in one place. It allows for easier tuning of message flow performance. If you use this method, you can change the values of attributes for a policy on the integration node, which then affects the behavior of a message flow without the need for redeployment.

WebSphere Education IBM

Message flow processing rate control terms



- There are various points in the flow of data from one application to another where the message rate can be measured and controlled
- **Rate arriving in message flow** controls the rate for the message flow
  - Effectively limits **Rate received in end application**
  - Includes the total rates of all input nodes of any type in the message flow

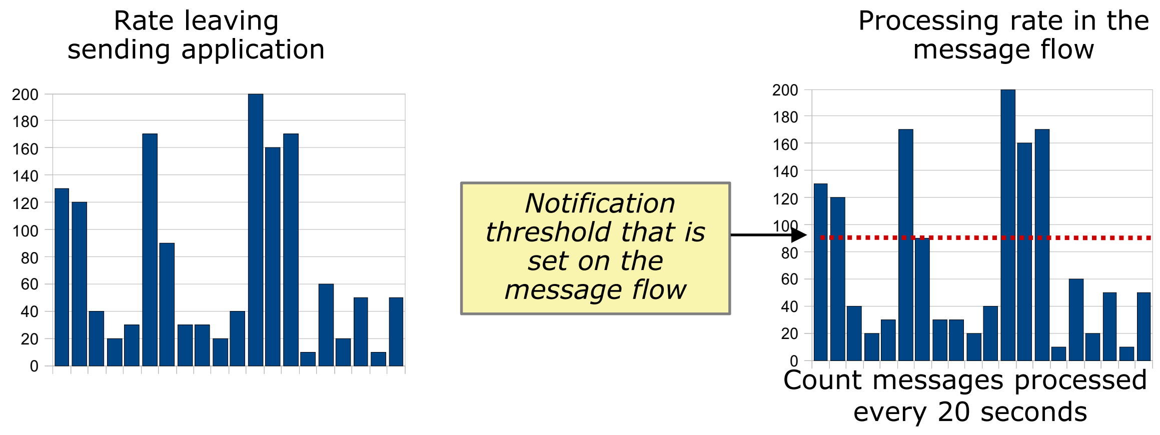
Figure 12-36. Message flow processing rate control terms WM6461.0

Notes:

The system administrator can set the maximum rate that an individual message flow can run at. The maximum rate is specified as the total number of input messages processed every second. When set, the number of input messages that are processed across the flow is measured. This measure is irrespective of the number of extra instances in use, the number of input nodes in the message flow, or the number of errors that occur. If necessary, a processing delay is introduced to keep the input message processing rate under the maximum flow rate setting.

WebSphere Education IBM

Message flow processing rate control: Notification



- If a sending application generates a message rate that is higher (or lower) than expected, configure the integration node to send a notification when a threshold is exceeded or dropped below the threshold
  - Publish/subscribe mechanism decouples notification from consumers
  - Write to Activity log
  - Write to User trace
- Action to reduce message rate up to the user:
  - Slow the sending application
  - Email administrator to investigate
  - Stop message flow to protect receiving application

Figure 12-37. Message flow processing rate control: Notification WM6461.0

Notes:

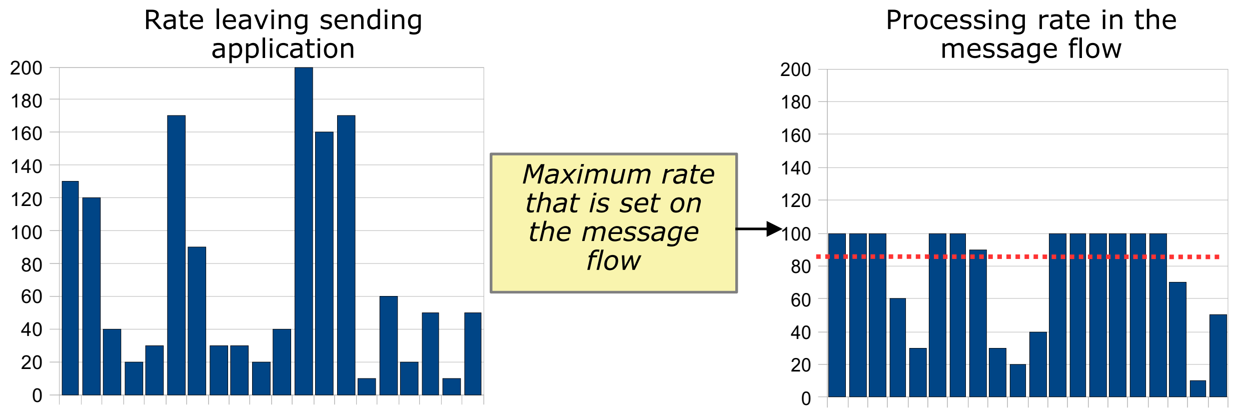
A common requirement is to be able to monitor the speed at which Integration Bus processes messages. Workload management allows the system administrator to express a notification threshold for individual message flows deployed.

- An *out of range* notification message is produced if the notification threshold is exceeded.
- A *back in range* notification message is produced if the notification threshold later drops back into range.

The action that is taken when the threshold is exceeded up to the user.

WebSphere Education

Message flow processing rate control: Delay



- Sending application generates a “bursty” message rate
- Receiving application that uses the messages can handle the average rate but not short-term fluctuations
- Configure message flow to:
  - Limit the rate on a message by message basis
  - Delay messages if the message flow is going too fast
- Time is measured before getting message
- Time is measured again before getting next message
- Delay is made if time difference is less than the time required to keep to the maximum rate

Figure 12-38. Message flow processing rate control: Delay WM6461.0

Notes:

The system administrator can set the maximum rate that an individual message flow can run at.

The system administrator can restrict the message flow run rate by setting the maximum rate property. The maximum rate is specified as the total number of input messages processed every second. The maximum rate value is divided equally among all threads that are running in the message flow irrespective of the number of input nodes within the flow.

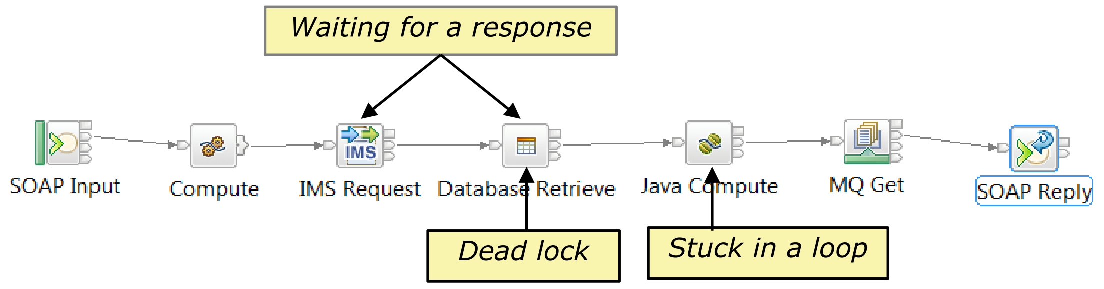
To calculate the number of threads within a specific message flow:

1. Count the number of input nodes within the flow.
2. Add on the number of extra instances that are specified for each input node.

If any thread exceeds its maximum rate allocation, a processing delay is introduced on that thread. The processing delay keeps the processing rate under the assigned maximum rate allocation.

WebSphere Education IBM

Unresponsive flows



- Message flow processing can become unresponsive when:
  - Waiting for a response from an external system
  - Processing a loop or a calculation that takes a long time
  - Deadlocked between two resources

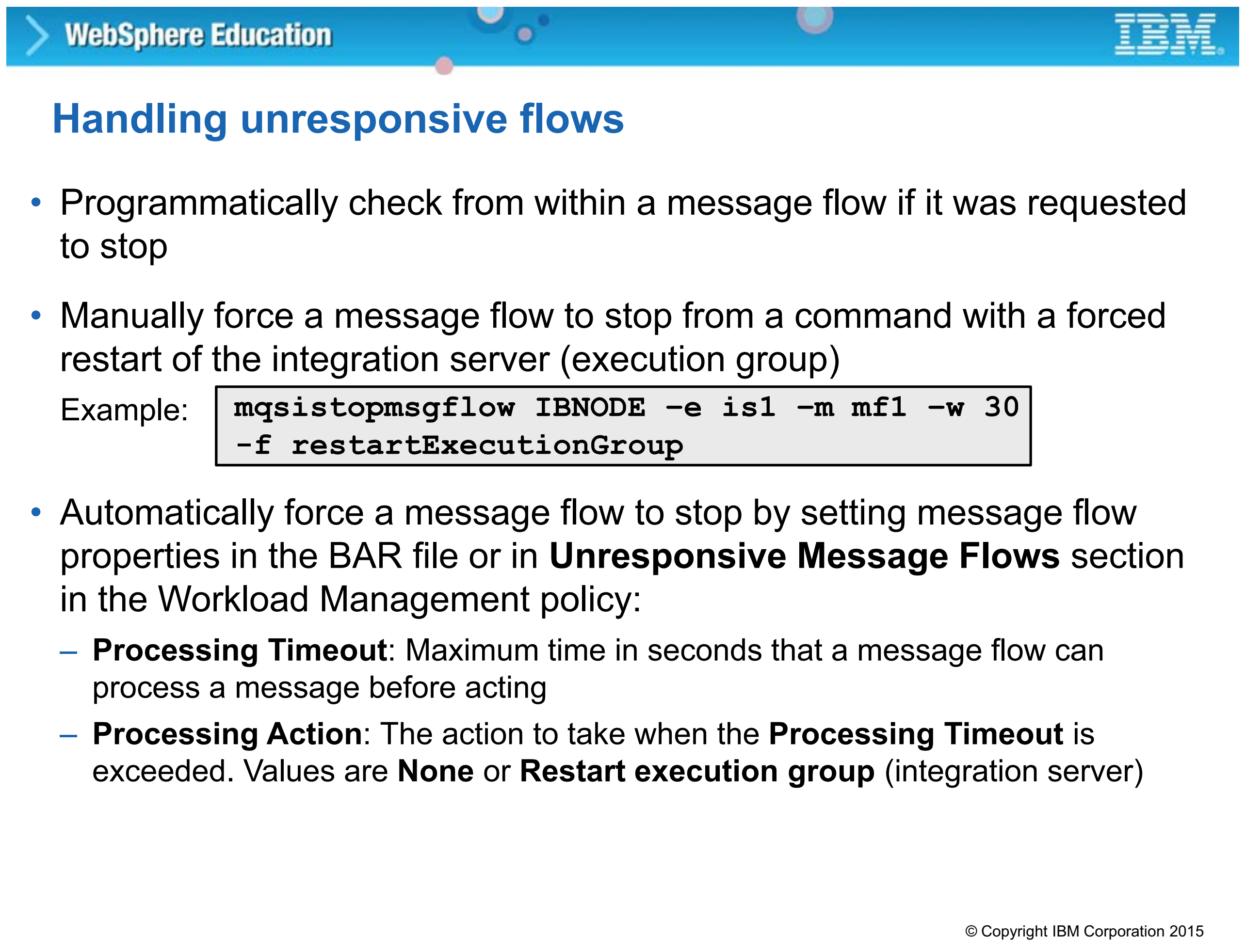
Figure 12-39. Unresponsive flows WM6461.0

Notes:

Under certain conditions, message flow processing can become unresponsive, for instance when:

- Waiting for a response from an external system
- Processing a loop or a calculation that takes a long time
- Deadlocked between two resources

Workload management allows system administrators to monitor a message flow to see whether it was requested to stop and to take the required action.



WebSphere Education

Handling unresponsive flows

- Programmatically check from within a message flow if it was requested to stop
- Manually force a message flow to stop from a command with a forced restart of the integration server (execution group)

  Example:

  ```
  mqsistopmsgflow IBNODE -e is1 -m mf1 -w 30
  -f restartExecutionGroup
  ```

- Automatically force a message flow to stop by setting message flow properties in the BAR file or in **Unresponsive Message Flows** section in the Workload Management policy:
  - **Processing Timeout**: Maximum time in seconds that a message flow can process a message before acting
  - **Processing Action**: The action to take when the **Processing Timeout** is exceeded. Values are **None** or **Restart execution group** (integration server)

© Copyright IBM Corporation 2015

Figure 12-40. Handling unresponsive flows WM6461.0

Notes:

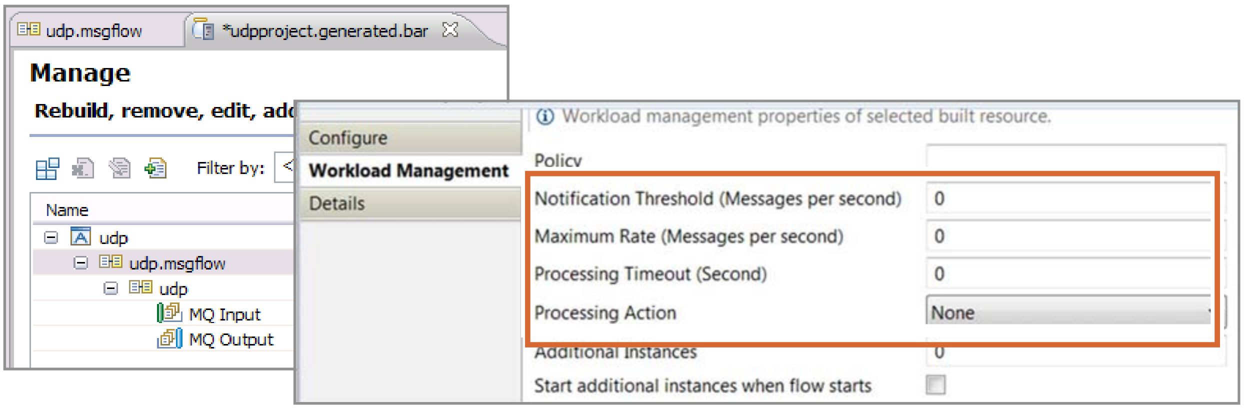
You can handle unresponsive flows programmatically, manually, or automatically.

- You can programmatically check from within a message flow if it was requested to stop by using the programming APIs (ESQL, Java, and .NET).
- You can manually force a message flow to stop by using a *force* option (**-f**) on the **`mqsistopmsgflow`** command.
- You can automatically force a message flow to stop by setting the **Processing Timeout** and **Processing Action** properties in the BAR file or the workload management policy.

WebSphere Education

Workload Management BAR file properties

- **Notification Threshold (Messages per second)**
  - Rate at which notification is published when it is exceeded
  - Default: No limit
- **Maximum Rate (Messages per second)**
  - Maximum rate a flow processes message
  - Default: No limit
- **Processing Timeout** and **Processing Action**


udp.msgflow
*udpproject.generated.bar
Manage
Rebuild, remove, edit, add
Filter by:
Name
udp
udp.msgflow
udp
MQ Input
MQ Output
Configure
Workload Management
Details
Workload management properties of selected built resource.

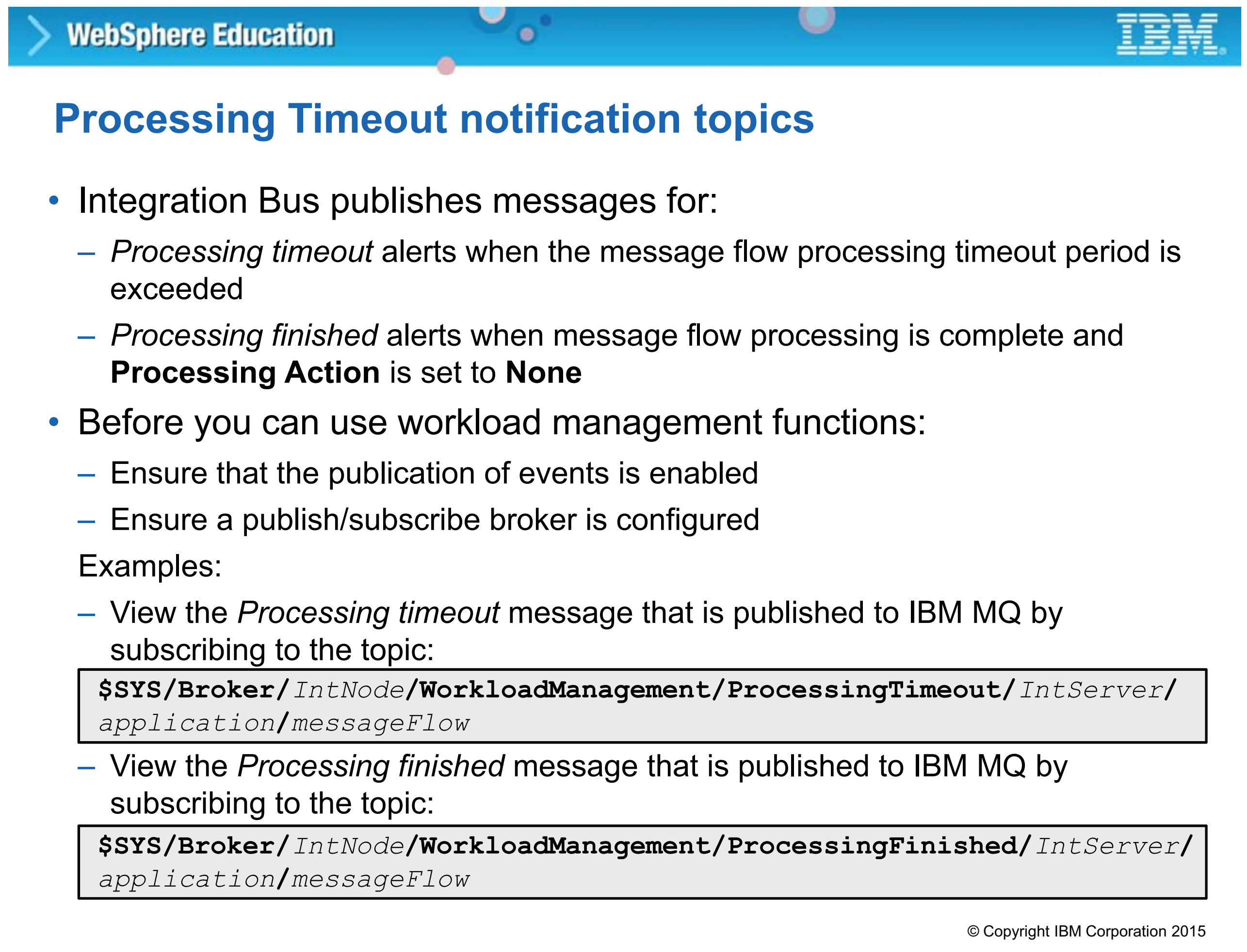
Policy	
Notification Threshold (Messages per second)	0
Maximum Rate (Messages per second)	0
Processing Timeout (Second)	0
Processing Action	None
Additional Instances	0
Start additional instances when flow starts	☐

Figure 12-41. Workload Management BAR file properties WM6461.0

Notes:

Workload management properties can be configured on the BAR file by using the BAR File editor in the Integration Toolkit or through the **`mqsiapplybaroverride`** command. The figure shows the properties in the BAR File editor.

- **Notification Threshold** is the rate that must be exceeded before a threshold notification is generated.
- **Maximum Rate** is the maximum rate of input messages processed every second that must be exceeded before a processing delay is introduced on that thread. The processing delay keeps the processing rate under the assigned maximum.
- **Processing Timeout** is the maximum time a message flow can process a message before running the action specified in **Processing Action**.
- **Processing Action** is the action to take when the **Processing Timeout** is exceeded. Currently, this property is restricted to **None** or **Restart execution group**.



Processing Timeout notification topics

- Integration Bus publishes messages for:
  - *Processing timeout* alerts when the message flow processing timeout period is exceeded
  - *Processing finished* alerts when message flow processing is complete and **Processing Action** is set to **None**
- Before you can use workload management functions:
  - Ensure that the publication of events is enabled
  - Ensure a publish/subscribe broker is configured

  Examples:
  - View the *Processing timeout* message that is published to IBM MQ by subscribing to the topic:

```
$SYS/Broker/IntNode/WorkloadManagement/ProcessingTimeout/IntServer/
application/messageFlow
```

  - View the *Processing finished* message that is published to IBM MQ by subscribing to the topic:

```
$SYS/Broker/IntNode/WorkloadManagement/ProcessingFinished/IntServer/
application/messageFlow
```

Figure 12-42. Processing Timeout notification topics WM6461.0

Notes:

Integration Bus publishes messages for various workload conditions that include the following publications:

- ***Processing Timeout*** alerts are published when the message flow processing timeout period is exceeded
- ***Processing Finished*** alerts are published when message flow processing is complete and no timeout action is specified (the **Processing Action** property is set to **None**)

The topic strings are available for subscriptions to receive the alerts. You can define subscriptions with IBM MQ Explorer. Alternatively you can write your own applications to subscribe to the publications for the integration servers, applications, and message flows that interest you.

You can view ***Processing Timeout*** alert message in IBM MQ by subscribing to the following topic:

```
$SYS/Broker/IntNode/WorkloadManagement/ProcessingTimeout/IntServer/
application/library/messageFlow
```

Where:

- *IntNode* is the name of the integration node

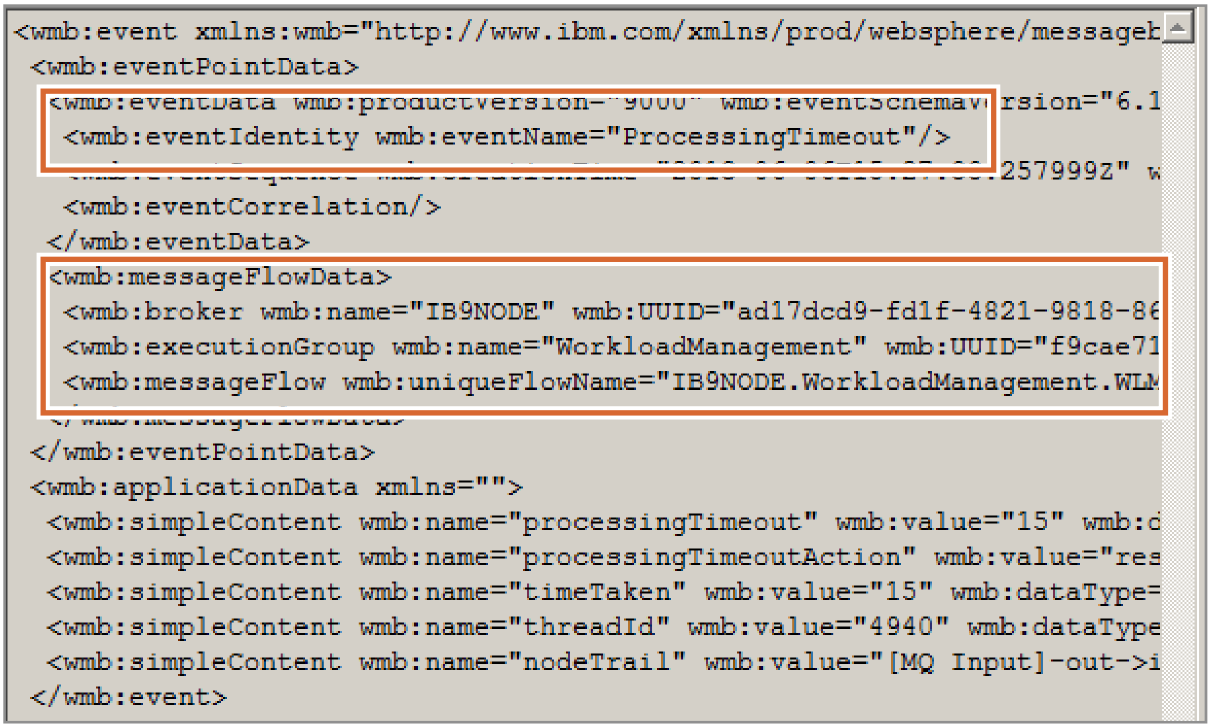
- *IntServer* is the name of the integration server on that integration node
- *application* is the name of the application on that integration server
- *library* is the name of the library on that application
- *messageFlow* is the name of the message flow that is deployed to the library

When the message flow is not contained in either an application or a library, the *application* or *library* parameters must be omitted along with their enclosing forward slash (/).

If the **Processing Action** property is set to **None** and processing of the message flow continues to completion, another event message is published to indicate that the processing is finished. You can view the message by subscribing to the following IBM MQ topic:

`$SYS/Broker/`*`IntNode`*`/WorkloadManagement/ProcessingFinished/`*`IntServer/application/library/messageFlow`*

If the **Processing Action** property is been set to **Restart the execution group,** the integration server is restarted. No further event messages are published from the message flow.



Processing Timeout alert example

```
<wmb:event xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messageb
 <wmb:eventPointData>
  <wmb:eventData wmb:productVersion="9000" wmb:eventSchemaVersion="6.1
   <wmb:eventIdentity wmb:eventName="ProcessingTimeout"/>
   <wmb:eventSequence wmb:creationTime="2013-06-06T15:27:00.257999Z" w
   <wmb:eventCorrelation/>
  </wmb:eventData>
  <wmb:messageFlowData>
   <wmb:broker wmb:name="IB9NODE" wmb:UUID="ad17dcd9-fd1f-4821-9818-86
   <wmb:executionGroup wmb:name="WorkloadManagement" wmb:UUID="f9cae71
   <wmb:messageFlow wmb:uniqueFlowName="IB9NODE.WorkloadManagement.WLM
  </wmb:messageFlowData>
 </wmb:eventPointData>
 <wmb:applicationData xmlns="">
  <wmb:simpleContent wmb:name="processingTimeout" wmb:value="15" wmb:d
  <wmb:simpleContent wmb:name="processingTimeoutAction" wmb:value="res
  <wmb:simpleContent wmb:name="timeTaken" wmb:value="15" wmb:dataType=
  <wmb:simpleContent wmb:name="threadId" wmb:value="4940" wmb:dataType
  <wmb:simpleContent wmb:name="nodeTrail" wmb:value="[MQ Input]-out->i
 </wmb:event>
```

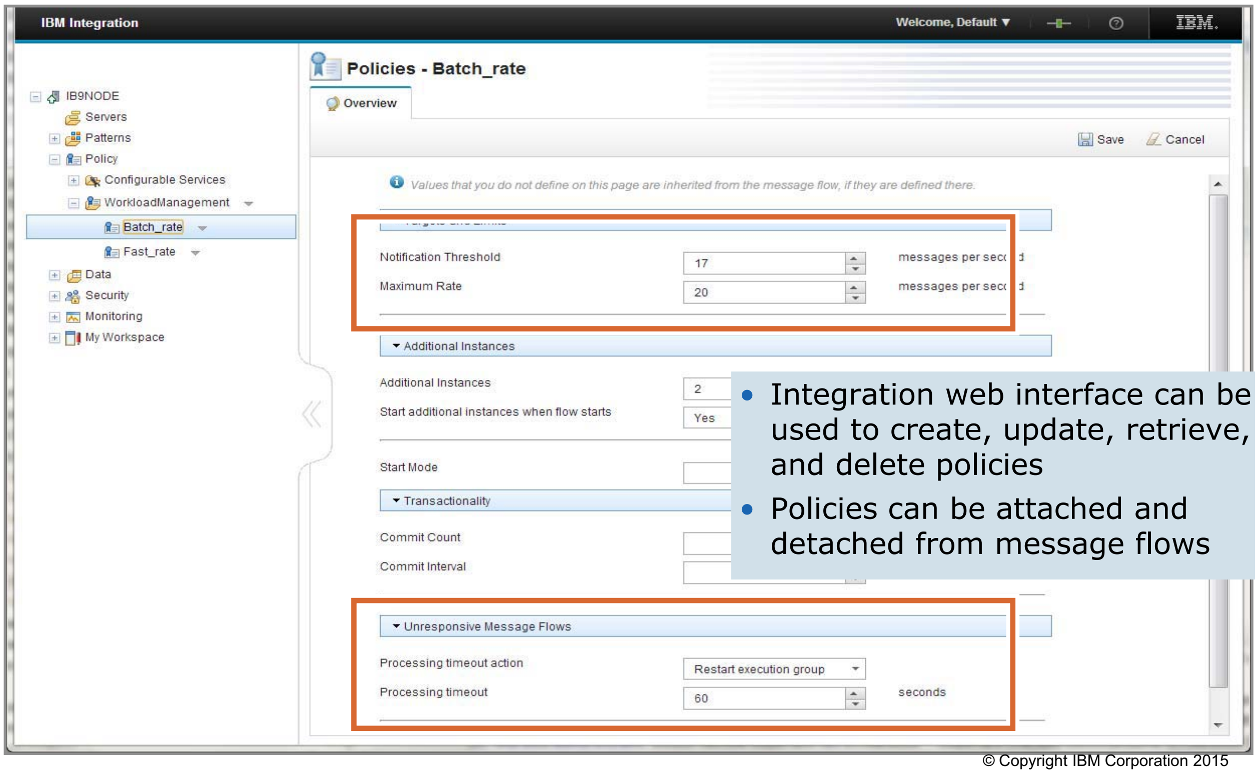
Figure 12-43. Processing Timeout alert example WM6461.0

Notes:

The figure shows an example of the XML message for a *Processing Timeout* event.

The XML message identifies the execution group (integration server) and message flow. The message also contains the processing timeout value for the message flow and the processing timeout action. It also reports the time that is taken, thread ID, and the node trail.

Using the Integration web interface to administer workload management policies



Values that you do not define on this page are inherited from the message flow, if they are defined there.

Notification Threshold	17	messages per second
Maximum Rate	20	messages per second

Additional Instances

Additional Instances	2
Start additional instances when flow starts	Yes
Start Mode	

Transactionality

Commit Count	
Commit Interval	

Unresponsive Message Flows

Processing timeout action	Restart execution group	
Processing timeout	60	seconds

- Integration web interface can be used to create, update, retrieve, and delete policies
- Policies can be attached and detached from message flows

Figure 12-44. Using the Integration web interface to administer workload management policies WM6461.0

Notes:

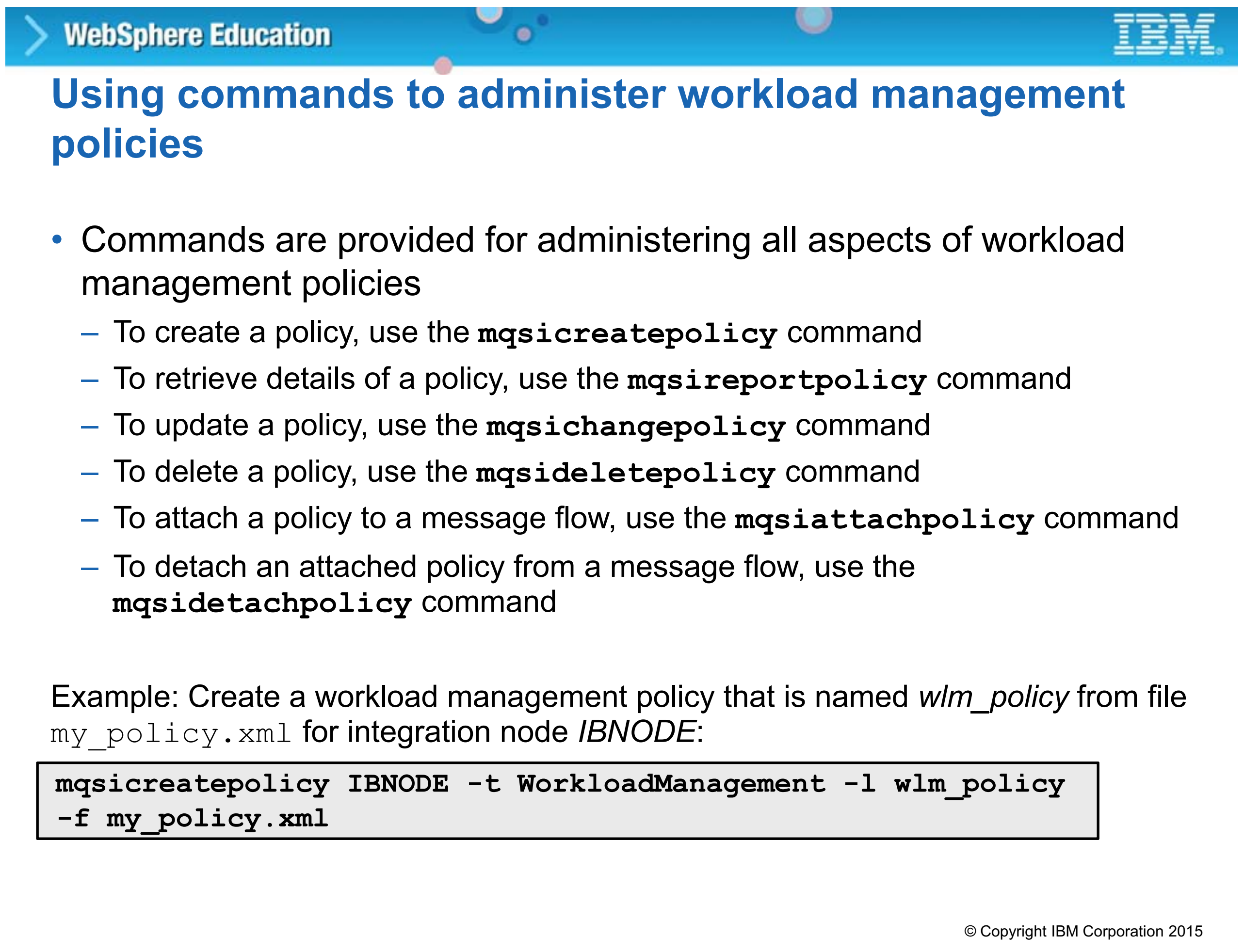
Instead of defining Workload Management properties on the BAR file, you can create policies so that message flows can refer to them at run time. If you use this method, you can change the values of attributes for a policy on the integration node, which then affects the behavior of a message flow without the need for redeployment.

The Workload Management policy encompasses all of the properties available under workload management in one place. These properties include the notification threshold and maximum rate, which allows for the easier tuning of message flow performance.

A policy can be set up and administered within the IBM Integration web interface.

When a policy is created and deployed, it adheres to the following precedence rules:

- Properties that are set in the policy take precedence over properties that are set as BAR file properties.
- If a property is not set in the policy, the equivalent property that is set on the BAR file takes effect.



WebSphere Education
IBM

Using commands to administer workload management policies

- Commands are provided for administering all aspects of workload management policies
  - To create a policy, use the **`mqsicreatepolicy`** command
  - To retrieve details of a policy, use the **`mqsireportpolicy`** command
  - To update a policy, use the **`mqsichangepolicy`** command
  - To delete a policy, use the **`mqsideletepolicy`** command
  - To attach a policy to a message flow, use the **`mqsiattachpolicy`** command
  - To detach an attached policy from a message flow, use the **`mqsidetachpolicy`** command

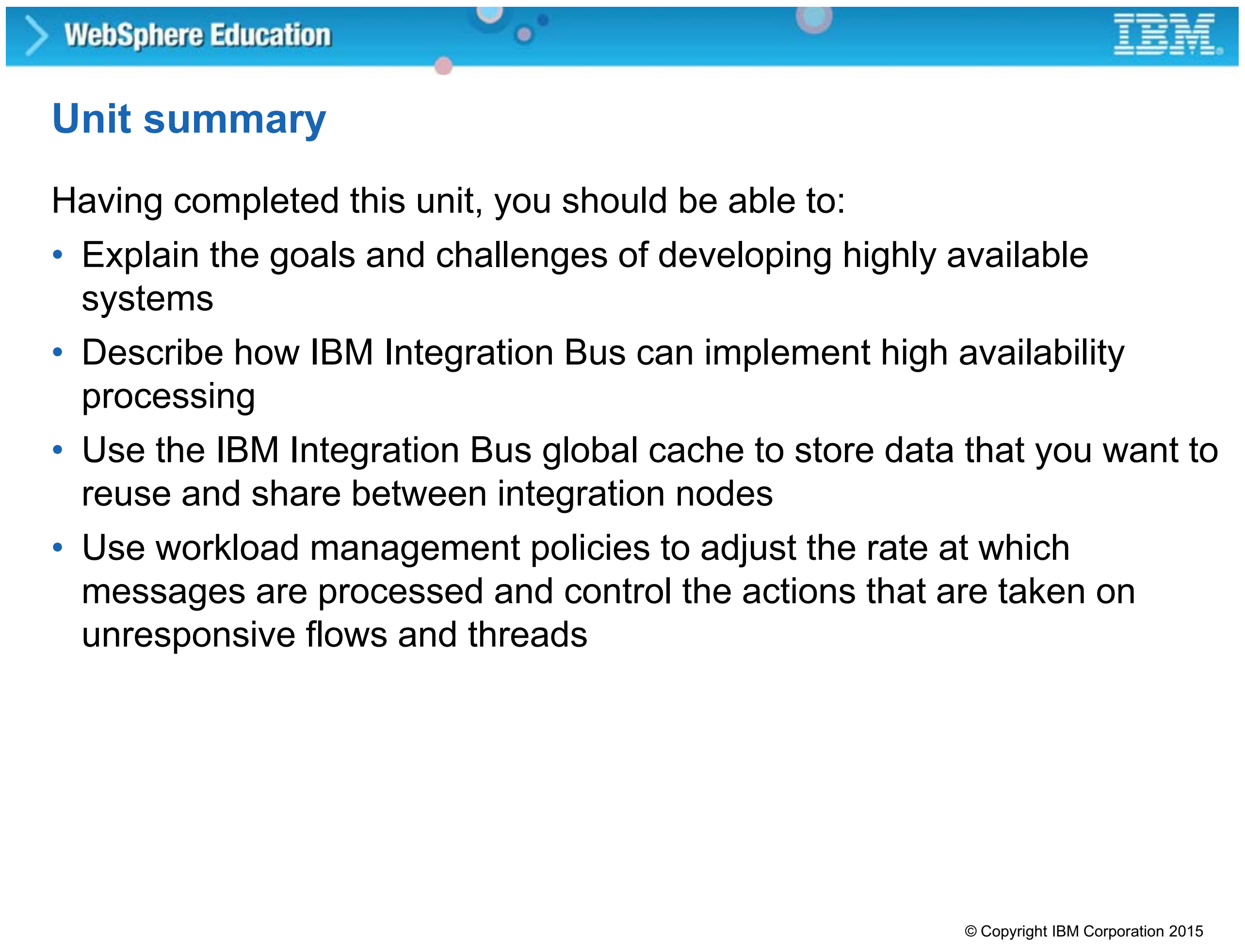
Example: Create a workload management policy that is named *wlm_policy* from file `my_policy.xml` for integration node *IBNODE*:

```
mqsicreatepolicy IBNODE -t WorkloadManagement -l wlm_policy
-f my_policy.xml
```

Figure 12-45. Using commands to administer workload management policies WM6461.0

Notes:

You can also use commands to administer all aspects of workload management policies.



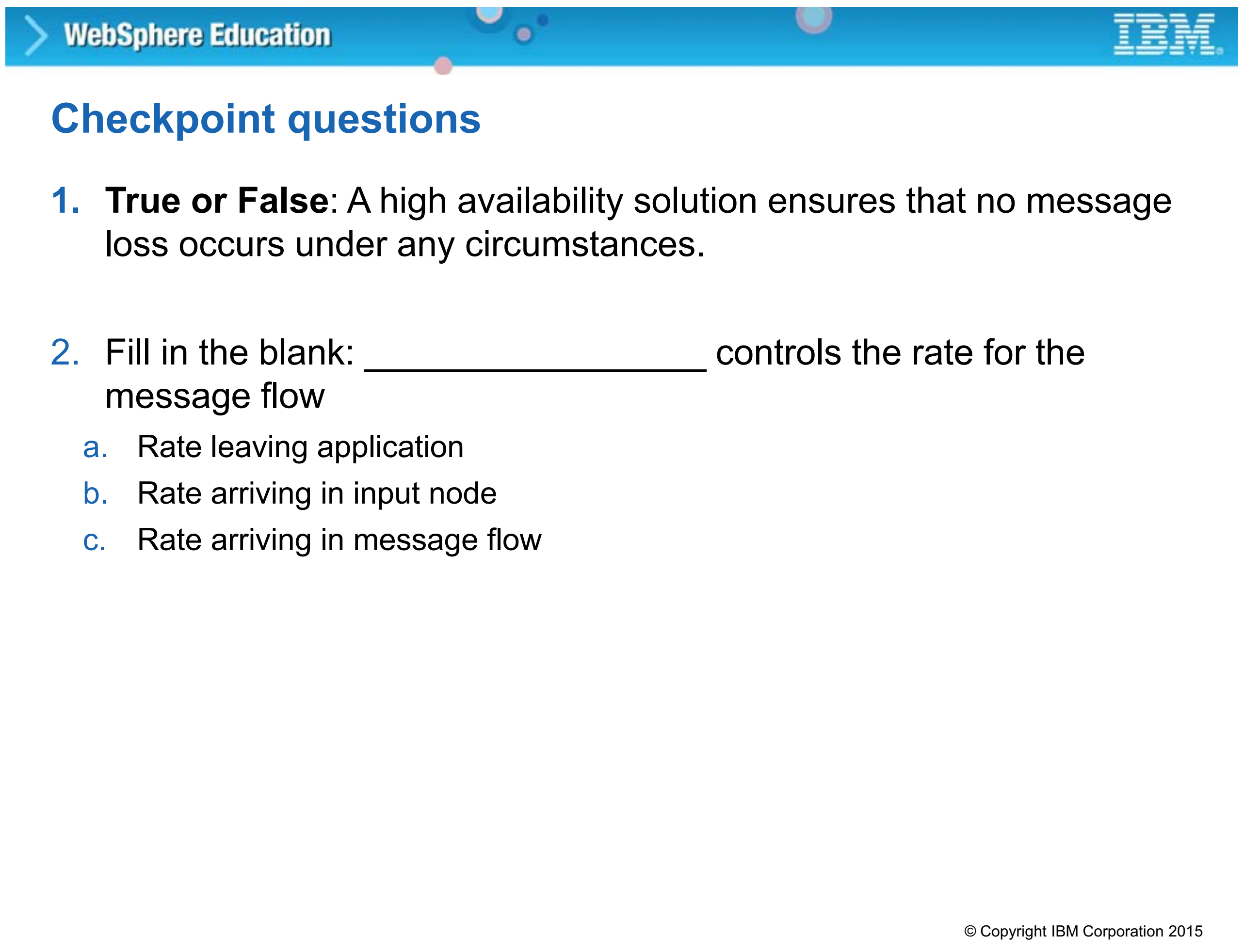
Unit summary

Having completed this unit, you should be able to:

- Explain the goals and challenges of developing highly available systems
- Describe how IBM Integration Bus can implement high availability processing
- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Use workload management policies to adjust the rate at which messages are processed and control the actions that are taken on unresponsive flows and threads

Figure 12-46. Unit summary WM6461.0

Notes:



WebSphere Education IBM

Checkpoint questions

1. **True or False**: A high availability solution ensures that no message loss occurs under any circumstances.

2. Fill in the blank: ________________ controls the rate for the message flow
   a. Rate leaving application
   b. Rate arriving in input node
   c. Rate arriving in message flow

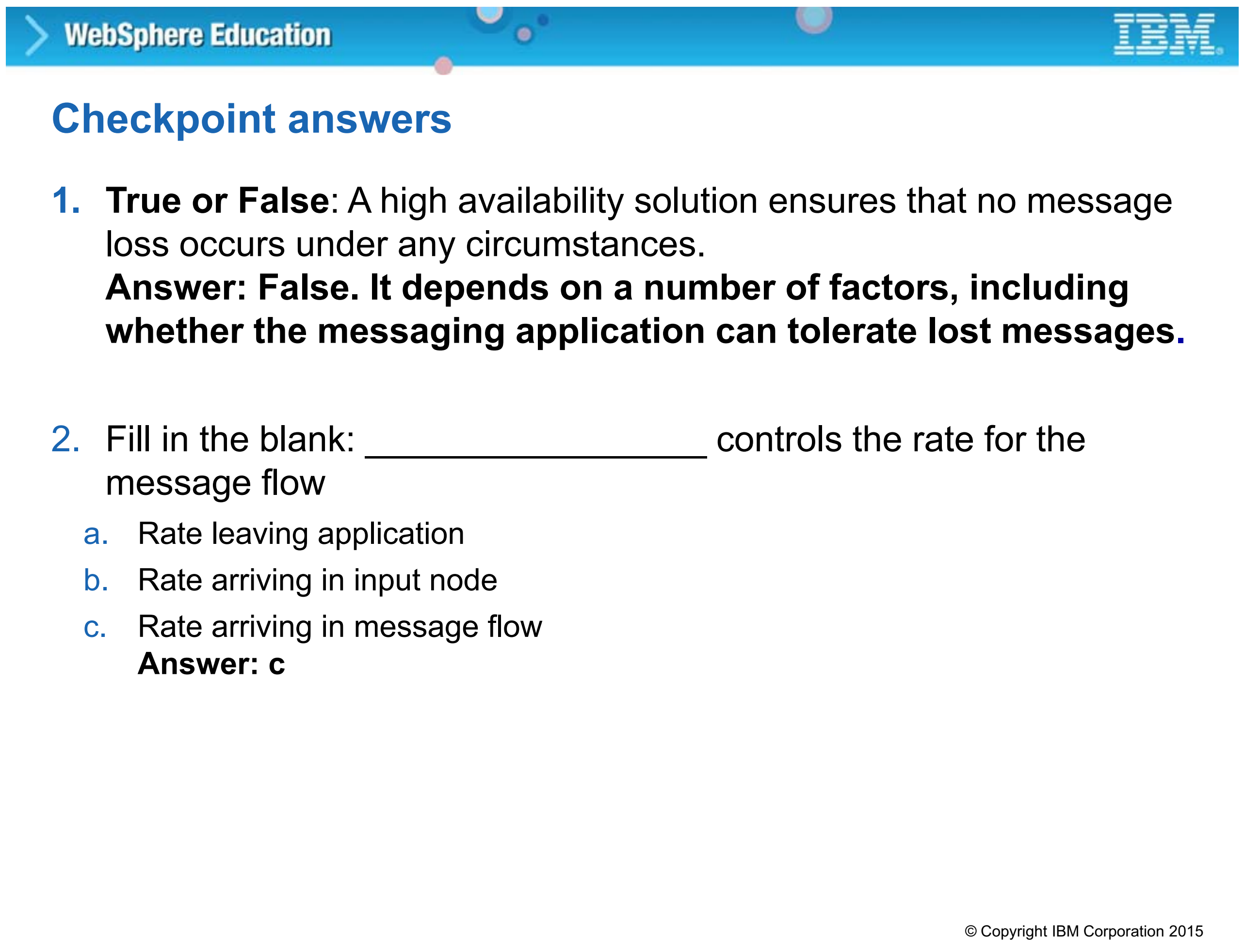
Figure 12-47. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education

IBM

Checkpoint answers

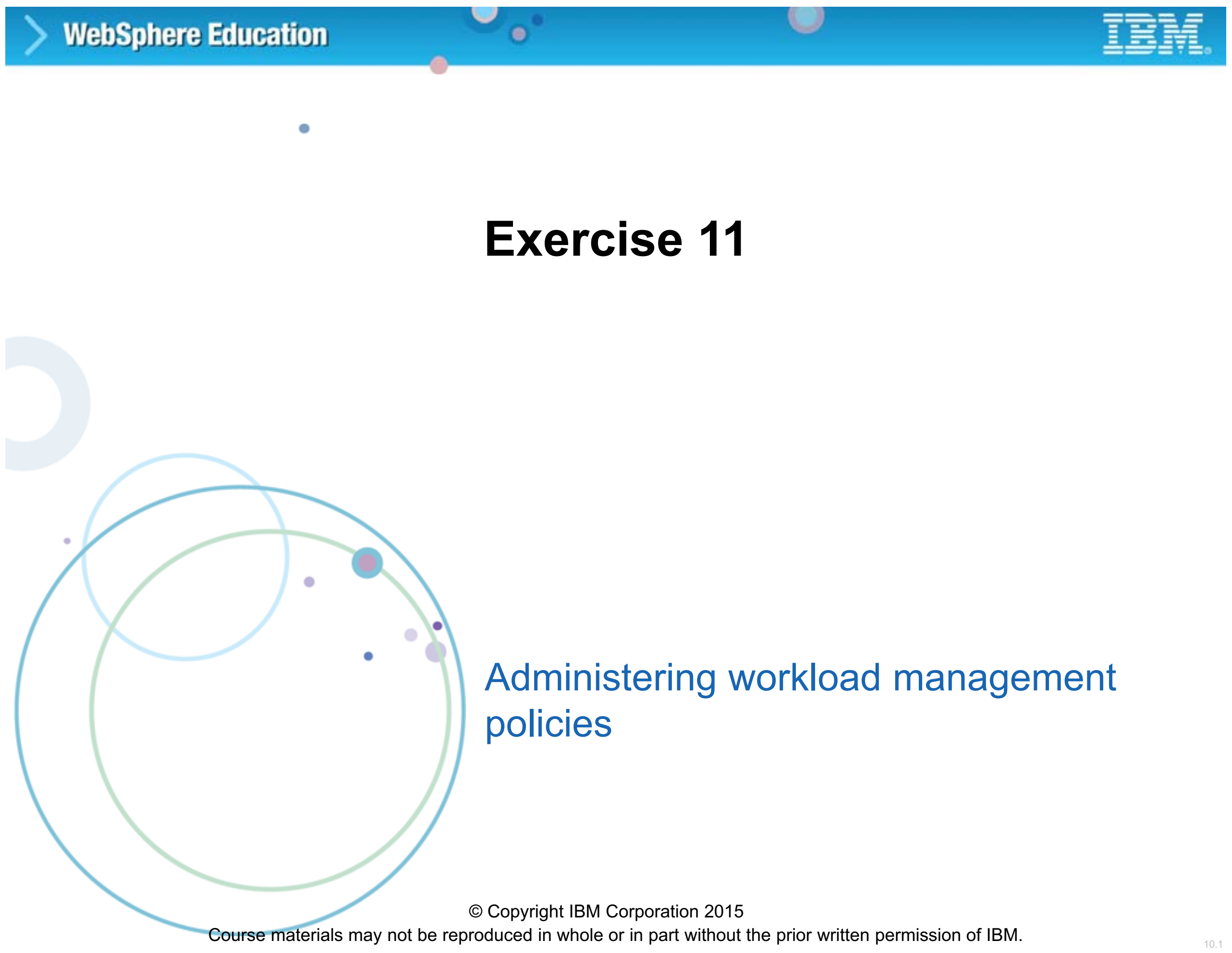
1. **True or False**: A high availability solution ensures that no message loss occurs under any circumstances.
   Answer: False. It depends on a number of factors, including whether the messaging application can tolerate lost messages.

2. Fill in the blank: ___________________ controls the rate for the message flow
   a. Rate leaving application
   b. Rate arriving in input node
   c. Rate arriving in message flow
   Answer: c

© Copyright IBM Corporation 2015

Figure 12-48. Checkpoint answers WM6461.0

Notes:


WebSphere Education
IBM.

Exercise 11

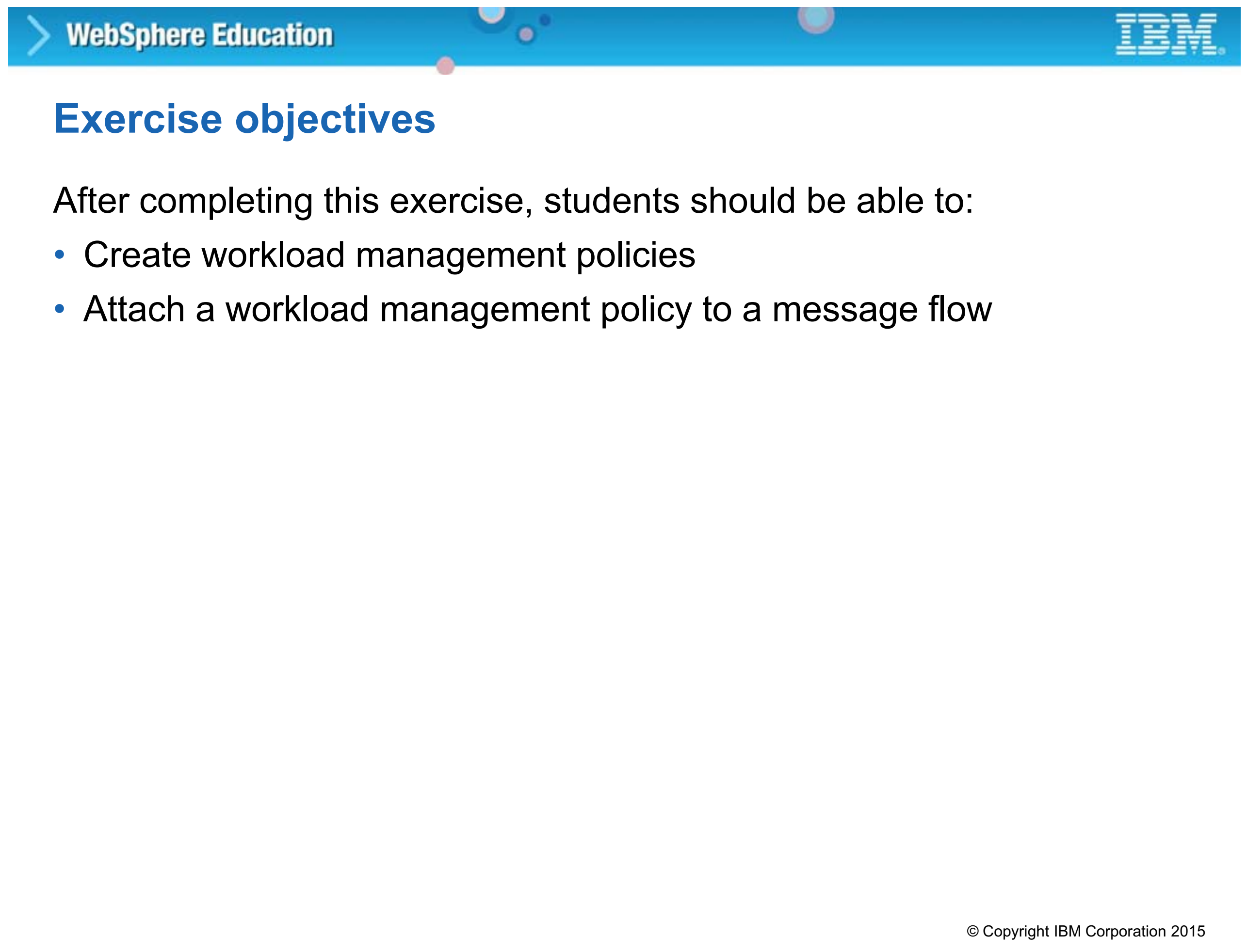
Administering workload management policies

10.1

Figure 12-49. Exercise 11 WM6461.0

Notes:

In this exercise, you create and attach workload management policies for a message flow.



WebSphere Education

Exercise objectives

After completing this exercise, students should be able to:

- Create workload management policies
- Attach a workload management policy to a message flow

© Copyright IBM Corporation 2015

Figure 12-50. Exercise objectives WM6461.0

Notes:

See the *Student Exercises Guide* for detailed instructions.

Unit 13. Monitoring, recording, and replaying message flow events

What this unit is about

This unit describes message flow event monitoring and how to create monitoring profiles that you can use to set runtime monitoring options. You also learn how to record event data to a database and use the IBM Integration web user interface to view events and replay messages.

What you should be able to do

After completing this unit, you should be able to:

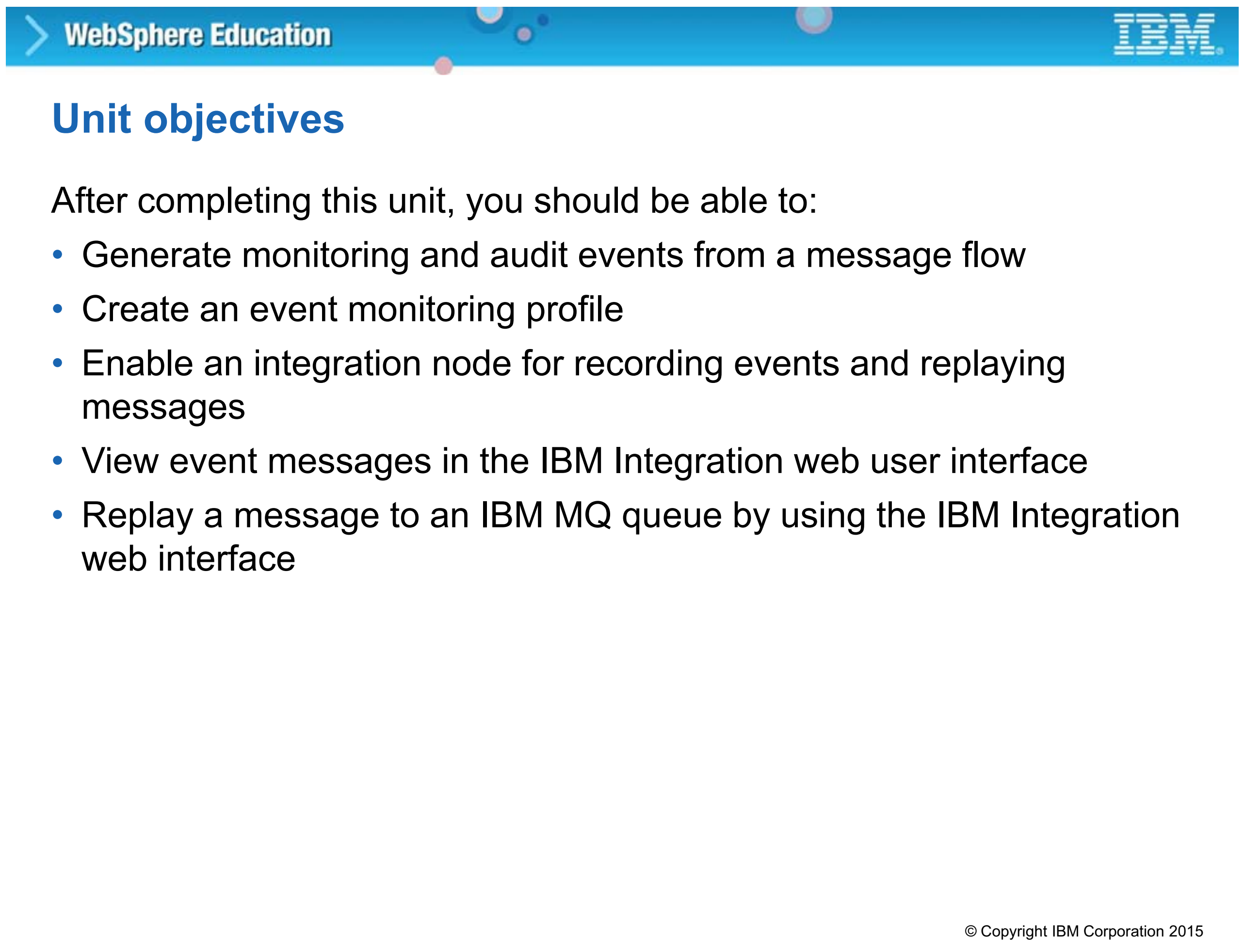
- Generate monitoring and audit events from a message flow
- Create an event monitoring profile
- Enable an integration node for recording events and replaying messages
- View event messages in the IBM Integration web user interface
- Replay a message to an IBM MQ queue by using the IBM Integration web interface

How you will check your progress

- Checkpoint questions
- Lab exercise: Recording and replaying message flow data

References

IBM Knowledge Center for IBM Integration Bus V10



WebSphere Education

Unit objectives

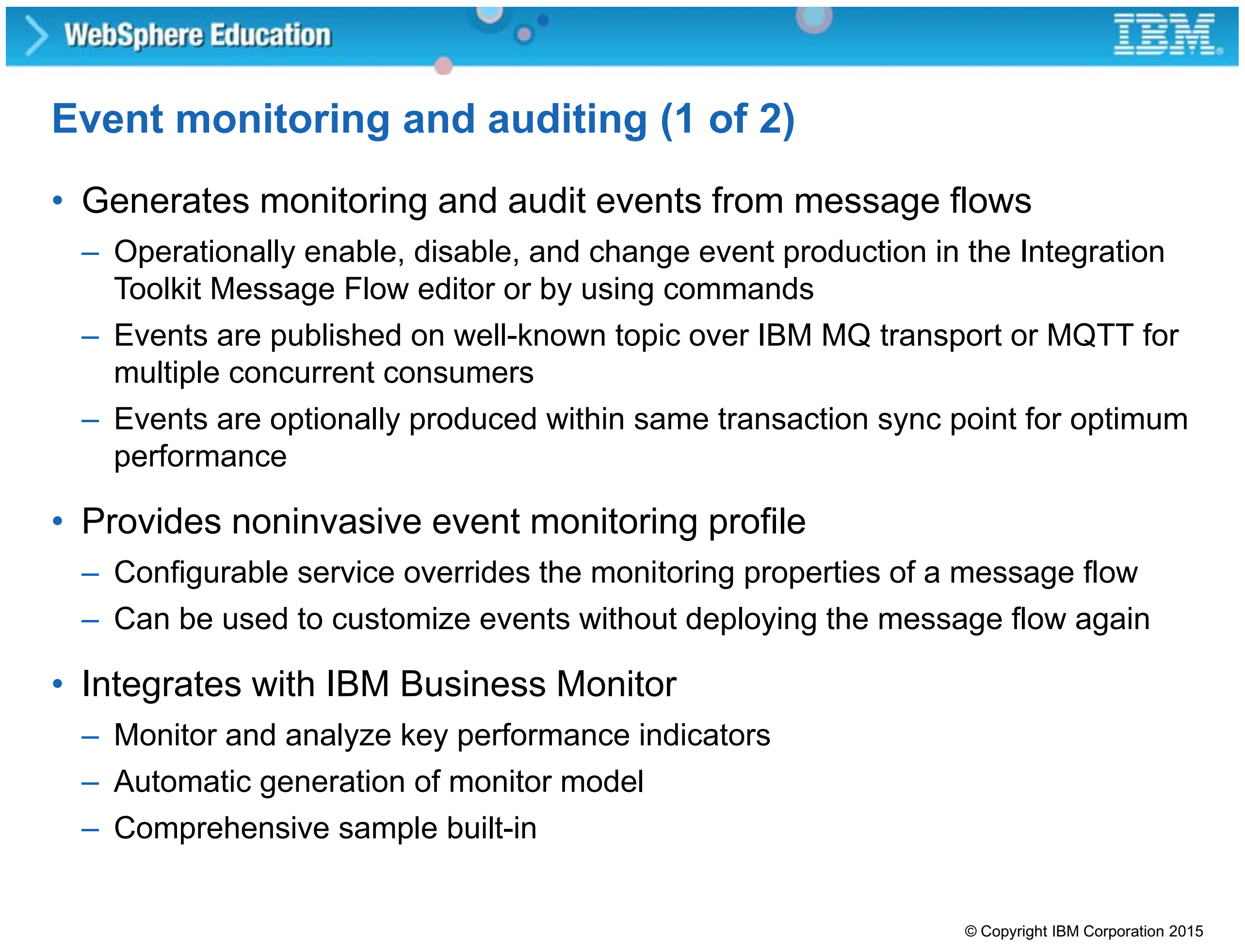
After completing this unit, you should be able to:

- Generate monitoring and audit events from a message flow
- Create an event monitoring profile
- Enable an integration node for recording events and replaying messages
- View event messages in the IBM Integration web user interface
- Replay a message to an IBM MQ queue by using the IBM Integration web interface

Figure 13-1. Unit objectives WM6461.0

Notes:

13.1. Event monitoring



WebSphere Education

IBM

Event monitoring and auditing (1 of 2)

- Generates monitoring and audit events from message flows
  - Operationally enable, disable, and change event production in the Integration Toolkit Message Flow editor or by using commands
  - Events are published on well-known topic over IBM MQ transport or MQTT for multiple concurrent consumers
  - Events are optionally produced within same transaction sync point for optimum performance
- Provides noninvasive event monitoring profile
  - Configurable service overrides the monitoring properties of a message flow
  - Can be used to customize events without deploying the message flow again
- Integrates with IBM Business Monitor
  - Monitor and analyze key performance indicators
  - Automatic generation of monitor model
  - Comprehensive sample built-in

Figure 13-2. Event monitoring and auditing (1 of 2) WM6461.0

Notes:

Message flows can be configured to emit events. Other applications can read and use these events for transaction monitoring, auditing, and business process monitoring. For example, it might be necessary to capture the first three fields of an input message at a certain point in the flow.

Events are published over a publish/subscribe network as a well-defined topic. You can also generate an event at design-time or operationally at any point throughout the flow, containing the entire payload or any piece of it.

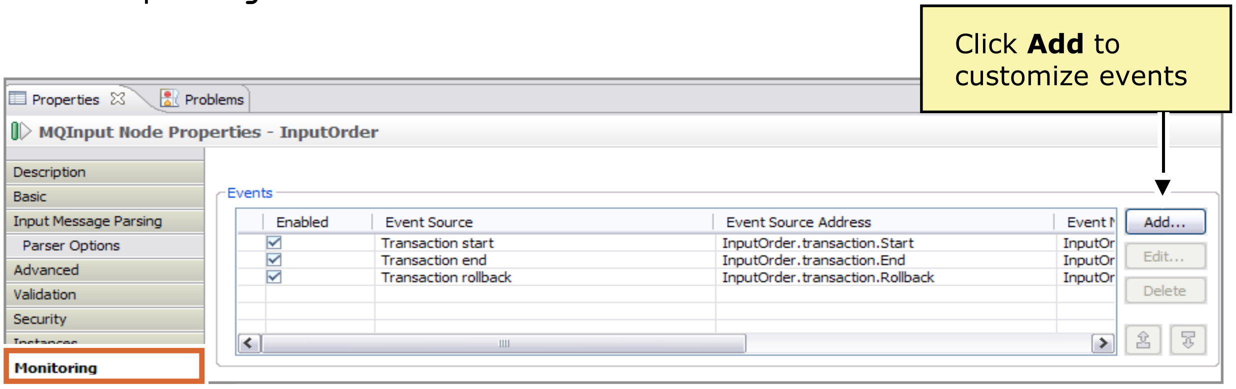
A monitoring profile configurable service can be used to customize events after a message flow is deployed, but without redeploying the flow.

IBM Business Monitor can be used to trend the Integration Bus event data. For example, you can analyze data volume changes, or break out the information by geographical area. A message driven bean captures the publication and pushes it into a common event infrastructure for integration with Business Monitor.

WebSphere Education IBM

Configuring event monitoring in the Integration Toolkit

- Every message flow node has a **Monitoring** tab to enable monitoring events
  - Transaction start, end, and rollback events
  - An event whenever a message is propagated from any terminal of any node
- Configure payload data, content style, identity, correlation, and sequencing data
- Event filters to limit exact conditions for event creation
  Example: `msg.Price>100`



Enabled	Event Source	Event Source Address	Event N
✓	Transaction start	InputOrder.transaction.Start	InputOr
✓	Transaction end	InputOrder.transaction.End	InputOr
✓	Transaction rollback	InputOrder.transaction.Rollback	InputOr

Figure 13-3. Configuring event monitoring in the Integration Toolkit WM6461.0

Notes:

In a development environment, the monitoring requirements are defined in the Integration Toolkit on the **Monitoring** tab on the message flow node properties.

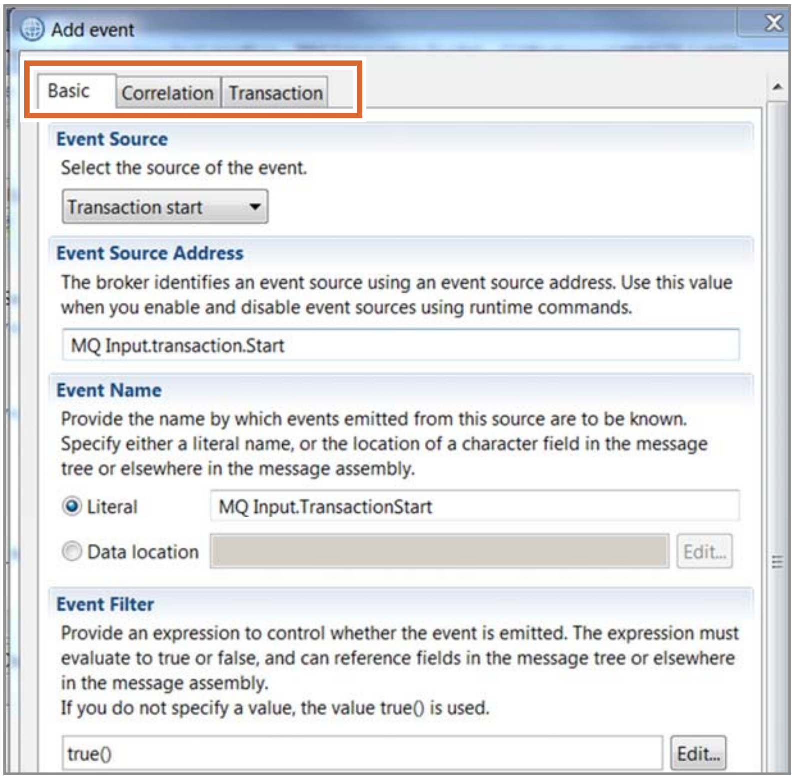
Developers can use the **Monitoring** properties to declare the event to see from a node, such as an input event, by checking the box next to the event. In the example, the MQ Input node has monitoring enabled at all points for a transaction: transaction start, transaction end, transaction rollback, Out terminal, Catch terminal, and Failure terminal.

It is also possible to declare sequencing information, correlating information, what the payload looks like, certain fields, and other information.

Event filters can be defined to generate events only when a certain condition applies, such as when a field in the message exceeds a specified value.

WebSphere Education

Customizing events



Event Source
Select the source of the event.
Transaction start

Event Source Address
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.
MQ Input.transaction.Start

Event Name
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.
[✓ Literal] MQ Input.TransactionStart
Data location Edit...

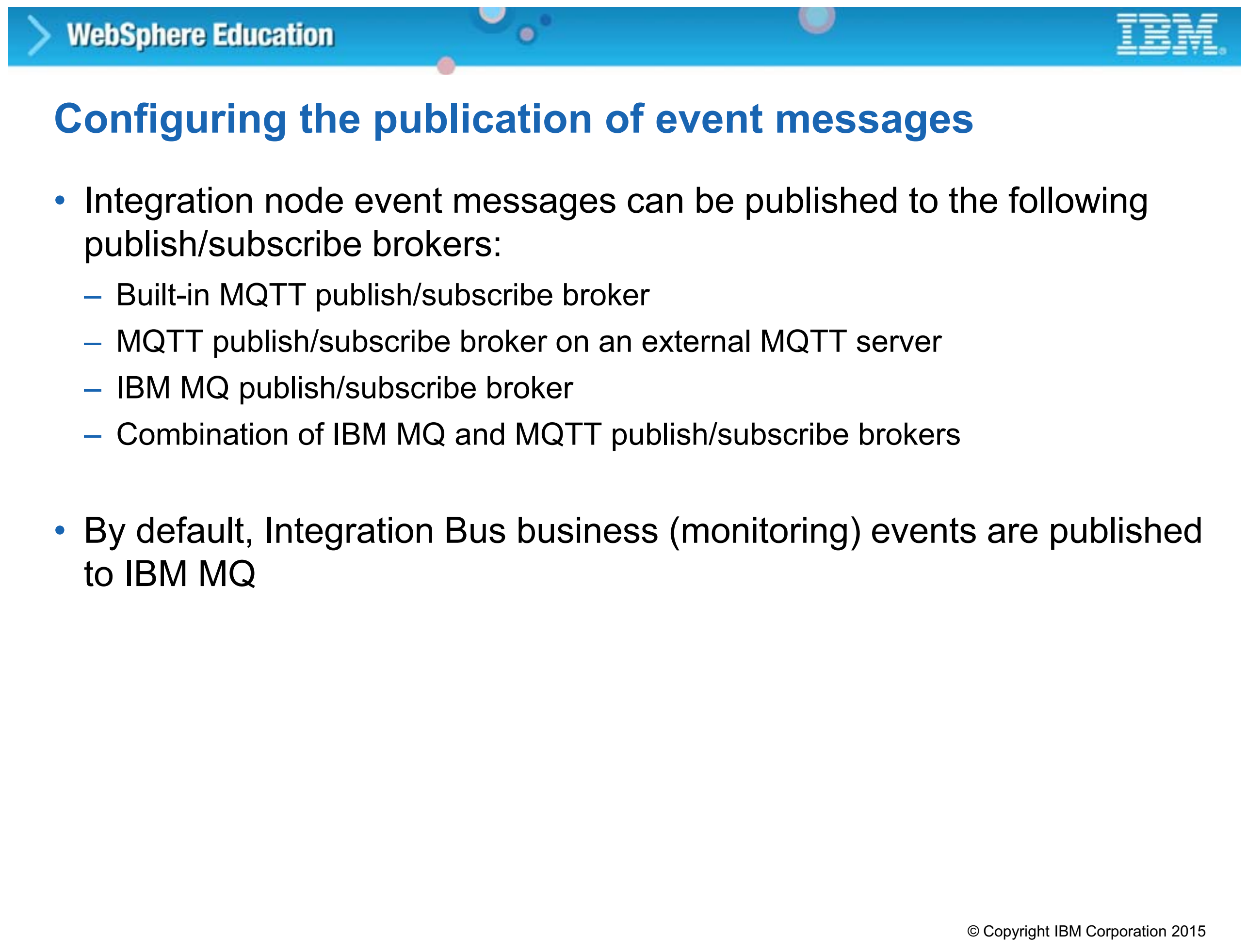
Event Filter
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly.
If you do not specify a value, the value true() is used.
true() Edit...

- Use the **Basic** tab to modify event properties and add event filters, and identify the event payload
- Use the **Correlation** tab when events must be correlated with events from a local or external process
- Use the **Transaction** tab to control the emission of monitoring events by a message flow by selecting one of the following options:
  - Coordinate the event emission with the message flow transaction
  - Events emission is an independent unit of work
  - Event emission is not in a unit of work

Figure 13-4. Customizing events WM6461.0

Notes:

The developer can customize events by clicking **Add** on the **Monitoring properties** tab and then specifying properties on the **Basic**, **Correlation**, and **Transaction** tab.



WebSphere Education

Configuring the publication of event messages

- Integration node event messages can be published to the following publish/subscribe brokers:
  - Built-in MQTT publish/subscribe broker
  - MQTT publish/subscribe broker on an external MQTT server
  - IBM MQ publish/subscribe broker
  - Combination of IBM MQ and MQTT publish/subscribe brokers
- By default, Integration Bus business (monitoring) events are published to IBM MQ

© Copyright IBM Corporation 2015

Figure 13-5. Configuring the publication of event messages WM6461.0

Notes:

You can configure the publication of event messages, including whether messages are published for an event message group. You can also define the publish/subscribe broker to which the messages are published. The topic is named with the integration node name, integration server, and the name that is given to the event as it is generated.

Integration node event messages can be published to the following publish/subscribe brokers:

- The built-in MQTT publish/subscribe broker
- An MQTT publish/subscribe broker on an external MQTT server
- An IBM MQ publish/subscribe broker
- A combination of IBM MQ and MQTT publish/subscribe brokers

The default publish/subscribe broker that is used for each group of event messages depends on your IBM Integration Bus deployment. Event messages are categorized as operational events, administration events, and business events.

- If IBM MQ is not installed, or IBM MQ is installed but a queue manager is not specified on the integration node, the messages are published to the following locations by default:
  - Operational events and administration events are published to the built-in MQTT publish/subscribe broker.
  - Business events are not published. If you want to publish Business events, you must enable the publication of the `BusinessEvents` group to the built-in MQTT publish/subscribe broker.
- If IBM MQ is installed and a queue manager is specified on the integration node, all events are published to the IBM MQ publish/subscribe broker by default.

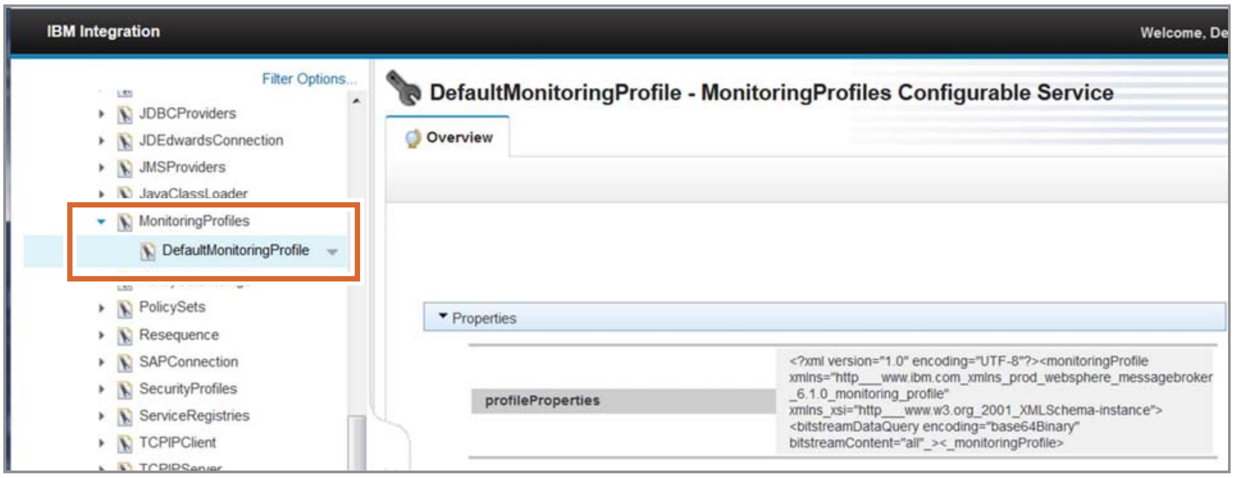
WebSphere Education IBM.

Event monitoring profiles

- Configurable service for customizing events after a message flow is deployed, but without redeploying the flow:

  Name: **`DefaultMonitoringProfile`**

  Property: **`profileProperties`**



profileProperties <?xml version="1.0" encoding="UTF-8"?><monitoringProfile xmlns="http___www.ibm.com_xmlns_prod_websphere_messagebroker_6.1.0_monitoring_profile" xmlns_xsi="http___www.w3.org_2001_XMLSchema-instance"><bitstreamDataQuery encoding="base64Binary" bitstreamContent="all" ><_monitoringProfile>

Figure 13-6. Event monitoring profiles WM6461.0

Notes:

You can use a monitoring profile configurable service to customize events after a message flow is deployed without deploying the message flow again.

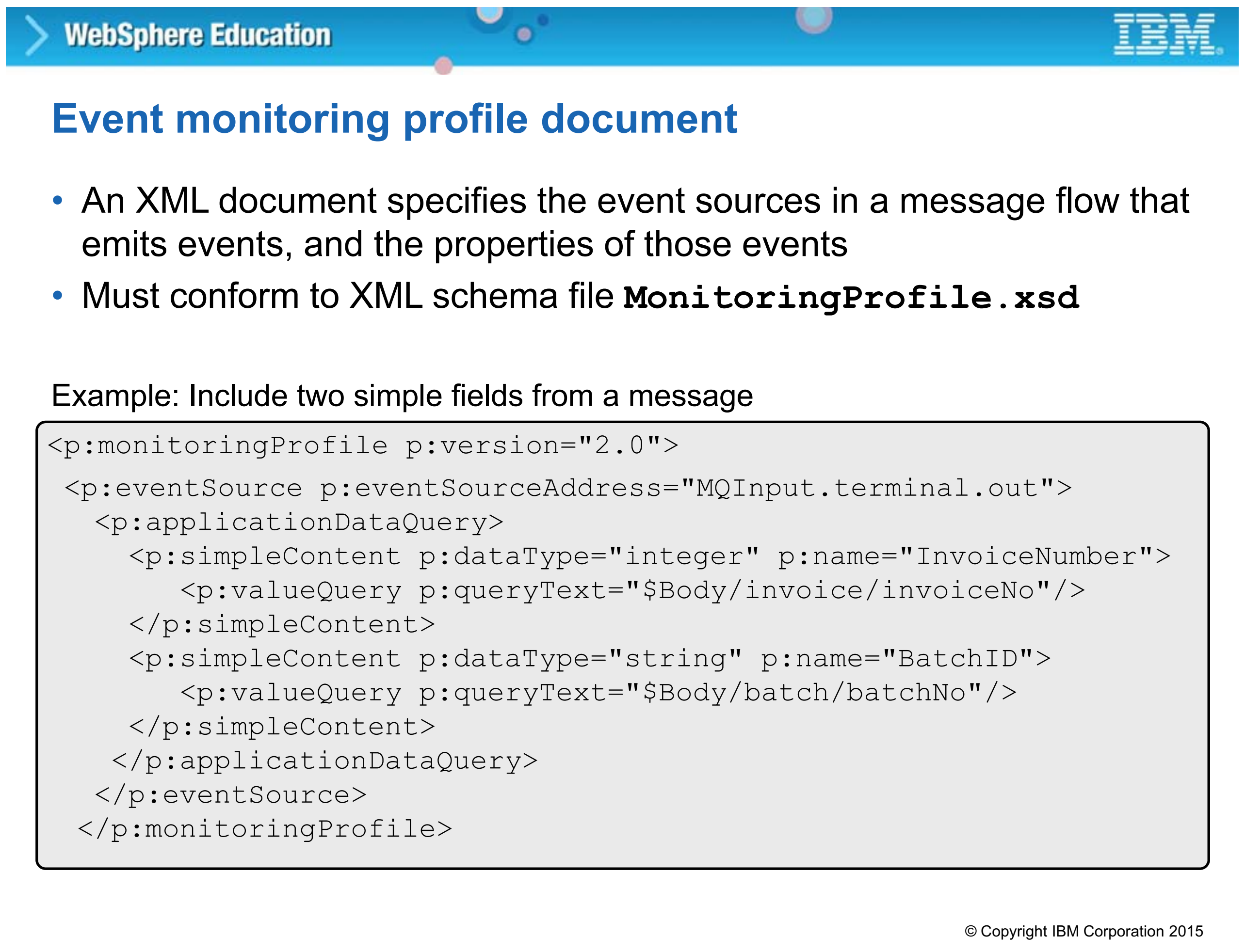
The monitoring profile configurable service has two parameters:

- The configurable service name
- The name of the monitoring profile file

The Business Monitor sample that is provided with Integration Bus includes the monitoring profile schema file (`MonitoringProfile.xsd`).

Hint

If the deployed message flow event monitoring properties were configured with the Integration Toolkit, use the `mqsireportflowmonitoring` command to create the equivalent monitoring profile the message flow. Use this profile as a starting point for creating other monitoring profiles.



WebSphere Education IBM

Event monitoring profile document

- An XML document specifies the event sources in a message flow that emits events, and the properties of those events
- Must conform to XML schema file **`MonitoringProfile.xsd`**

Example: Include two simple fields from a message

```
<p:monitoringProfile p:version="2.0">
 <p:eventSource p:eventSourceAddress="MQInput.terminal.out">
   <p:applicationDataQuery>
     <p:simpleContent p:dataType="integer" p:name="InvoiceNumber">
        <p:valueQuery p:queryText="$Body/invoice/invoiceNo"/>
     </p:simpleContent>
     <p:simpleContent p:dataType="string" p:name="BatchID">
        <p:valueQuery p:queryText="$Body/batch/batchNo"/>
     </p:simpleContent>
    </p:applicationDataQuery>
   </p:eventSource>
  </p:monitoringProfile>
```

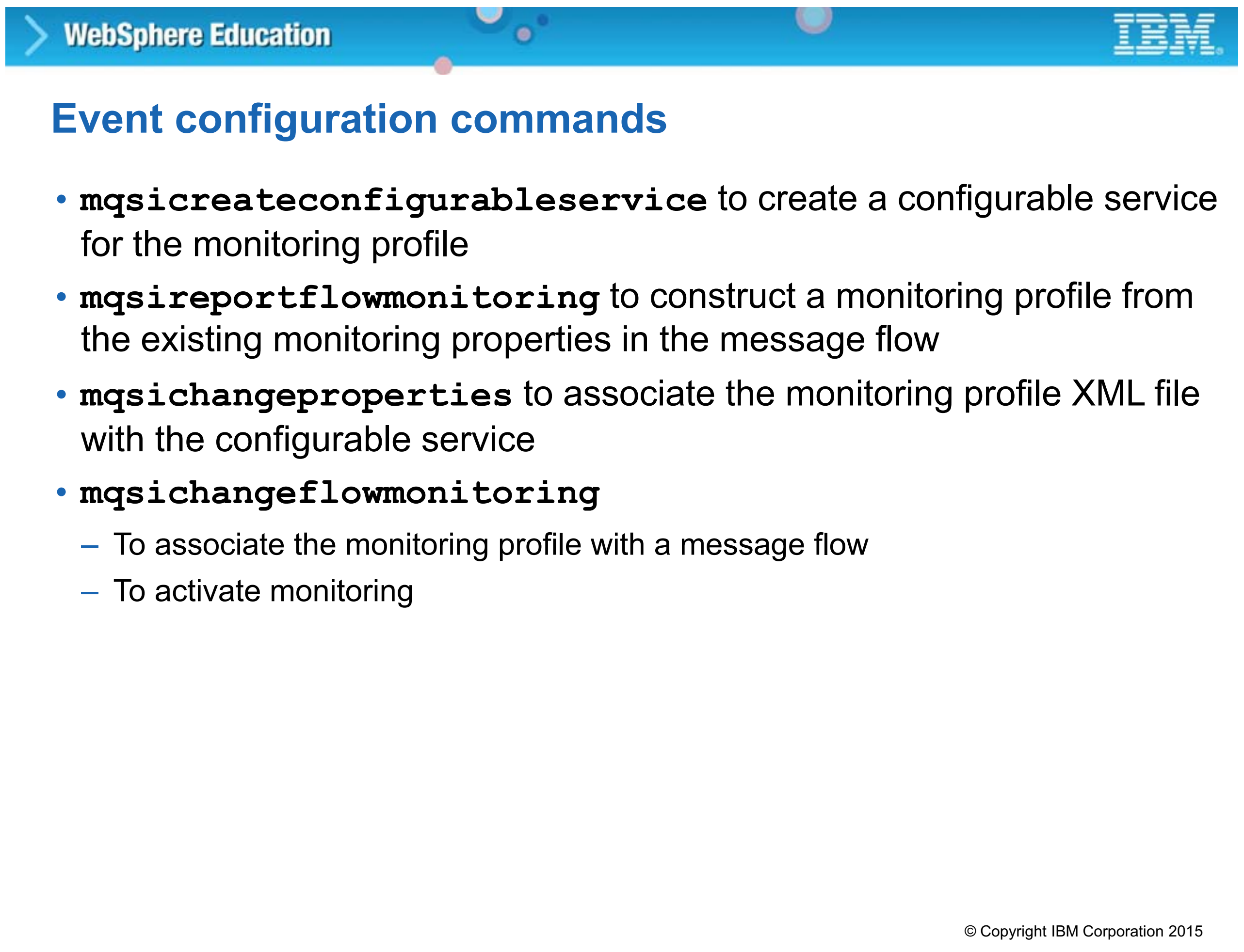
© Copyright IBM Corporation 2015

Figure 13-7. Event monitoring profile document WM6461.0

Notes:

The figure shows an example of an event monitoring profile.

The profile in the example captures the invoice number and batch ID from the body of the logical message. The message is captured at the MQInput node Out terminal (the event source).



WebSphere Education

Event configuration commands

- `mqsicreateconfigurableservice` to create a configurable service for the monitoring profile
- `mqsireportflowmonitoring` to construct a monitoring profile from the existing monitoring properties in the message flow
- `mqsichangeproperties` to associate the monitoring profile XML file with the configurable service
- `mqsichangeflowmonitoring`
  - To associate the monitoring profile with a message flow
  - To activate monitoring

© Copyright IBM Corporation 2015

Figure 13-8. Event configuration commands WM6461.0

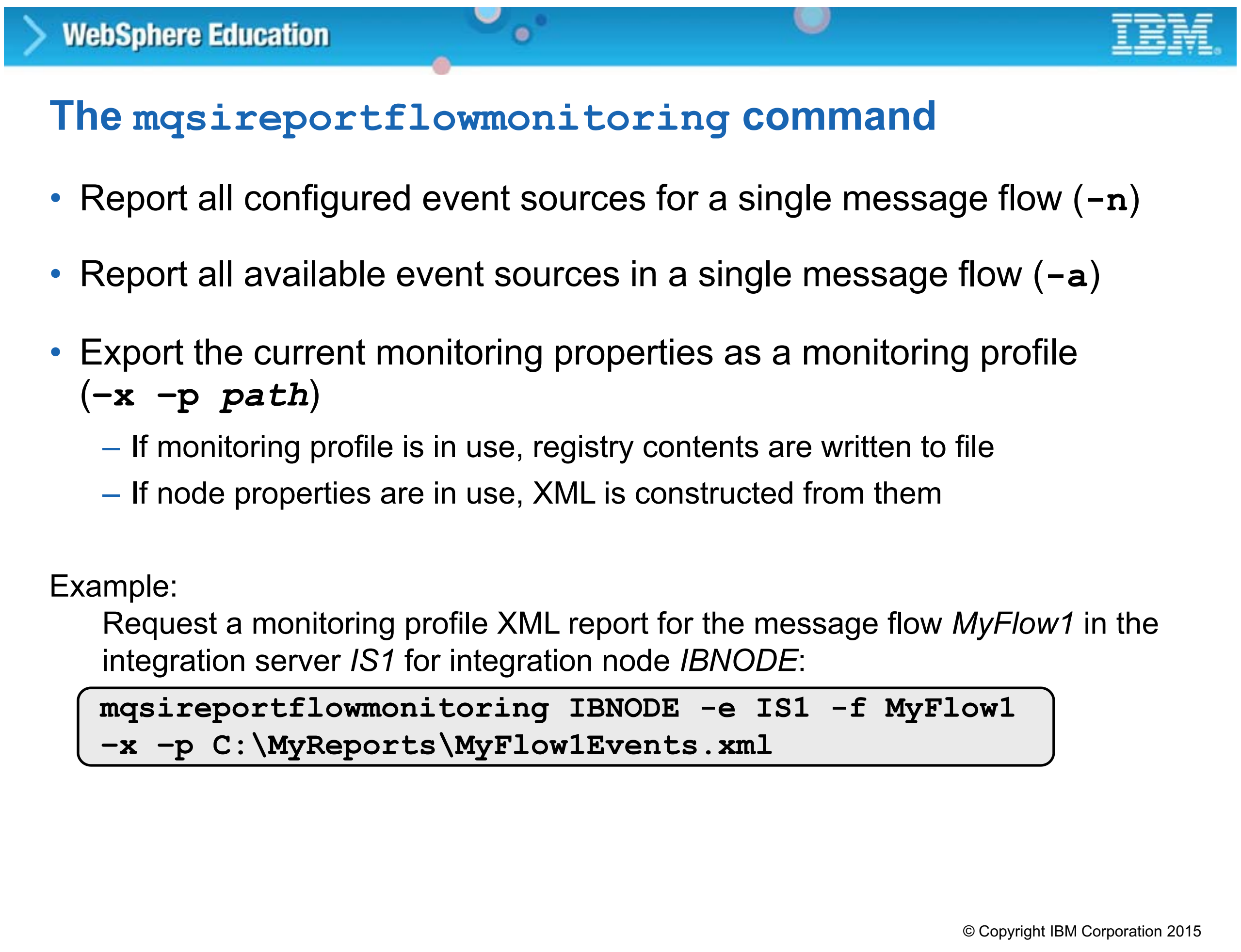
Notes:

Event monitoring can be configured from a command, without using the Integration Toolkit.

First, create an event monitoring configurable service by using the `mqsicreateconfigurableservice` command or the IBM Integration web interface.

Next, generate a base profile that you can use as a template by using the `mqsireportflowmonitoring` command.

When the monitoring profile is ready, load it using the `mqsichangeproperties` command. Finally, associate the monitoring profile with a message flow by using the `mqsichangeflowmonitoring` command or the IBM Integration web interface.



WebSphere Education IBM

The `mqsireportflowmonitoring` command

- Report all configured event sources for a single message flow (**`-n`**)
- Report all available event sources in a single message flow (**`-a`**)
- Export the current monitoring properties as a monitoring profile (**`-x -p`** ***`path`***)
  - If monitoring profile is in use, registry contents are written to file
  - If node properties are in use, XML is constructed from them

Example:

Request a monitoring profile XML report for the message flow *MyFlow1* in the integration server *IS1* for integration node *IBNODE*:

```
mqsireportflowmonitoring IBNODE -e IS1 -f MyFlow1
-x -p C:\MyReports\MyFlow1Events.xml
```

Figure 13-9. The mqsireportflowmonitoring command WM6461.0

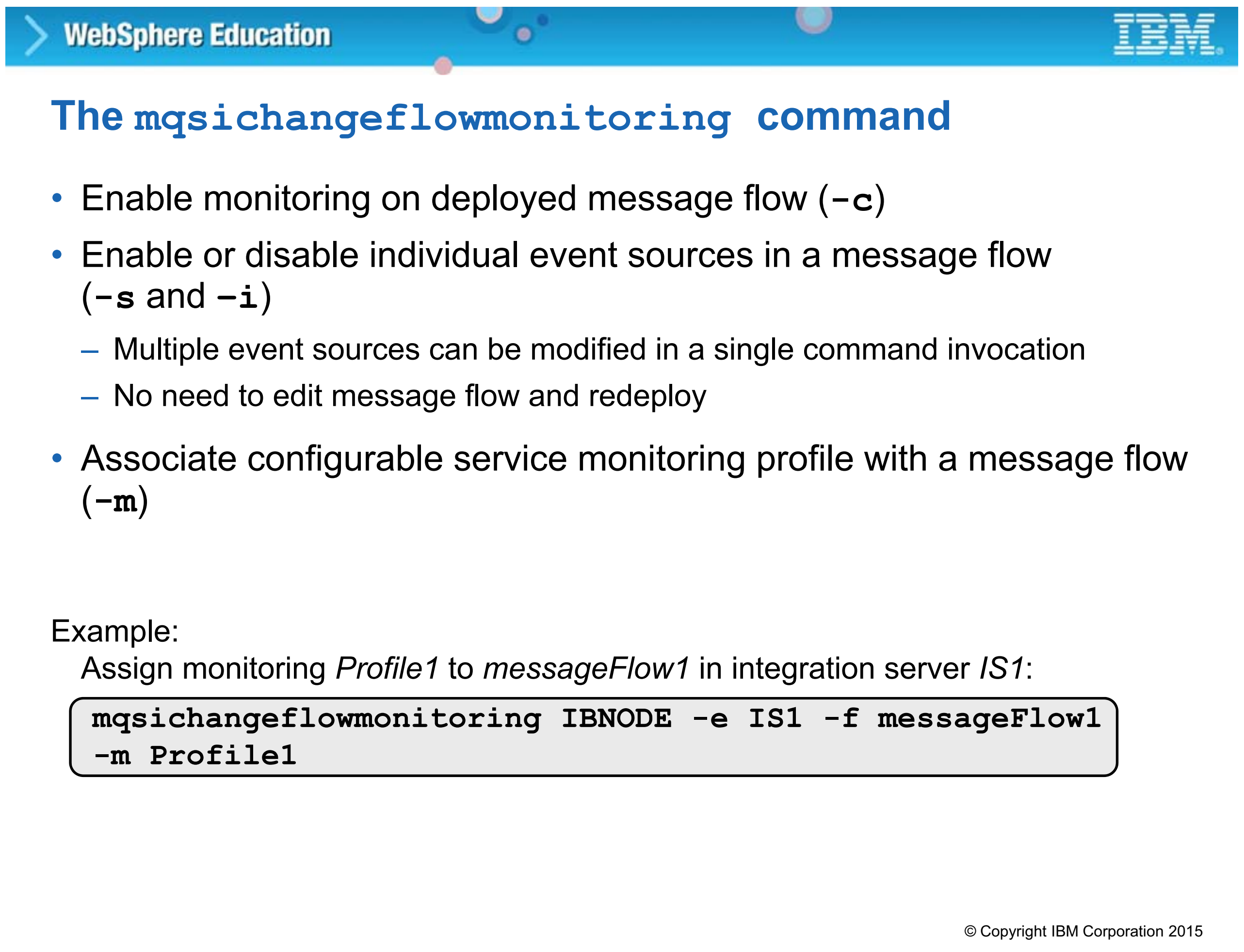
Notes:

The `mqsireportflowmonitoring` command can be used to construct a monitoring profile, rather than handcrafting it in a schema editor.

- The –n flag is equivalent to selecting the message flow canvas.
- The –a flag is useful when you must discover the event source addresses of the available event sources.
- The –x and –p flags are used to extract the monitoring profile to an XML file. If you extract the current configuration, you can modify it and reactivate monitoring by using the `mqsichangeflowmonitoring` command.

The command in the example can also generate the monitoring profile and check the event settings after a profile is deployed.

For the complete command syntax, see the product documentation.



The `mqsichangeflowmonitoring` command

- Enable monitoring on deployed message flow (**`-c`**)
- Enable or disable individual event sources in a message flow (**`-s`** and **`-i`**)
  - Multiple event sources can be modified in a single command invocation
  - No need to edit message flow and redeploy
- Associate configurable service monitoring profile with a message flow (**`-m`**)

Example:

Assign monitoring *Profile1* to *messageFlow1* in integration server *IS1*:

```
mqsichangeflowmonitoring IBNODE -e IS1 -f messageFlow1
-m Profile1
```

© Copyright IBM Corporation 2015

Figure 13-10. The mqsichangeflowmonitoring command WM6461.0

Notes:

The `mqsichangeflowmonitoring` command assigns a monitoring profile to message flow.

Whether you are configuring monitoring by using the Integration Toolkit or a command, you must activate monitoring with the `mqsichangeflowmonitoring` command with the –c flag. If you redeploy a flow, you must reactivate monitoring if you delete the existing flow, and then deploy.

You can use the `–m` flag to change the monitoring configuration without redeploying the message flow.

You can also use this command to directly enable event sources. In the following example:

`-s` precedes a comma-separated list of the event sources to be enabled or disabled.

`-i enabled` controls monitoring for the specified event source.

```
mqsichangeflowmonitoring IBNODE -e default -f myMessageFlow
-s "SOAP Input1.terminal.out,MQOutput1.terminal.in" -i enabled
```

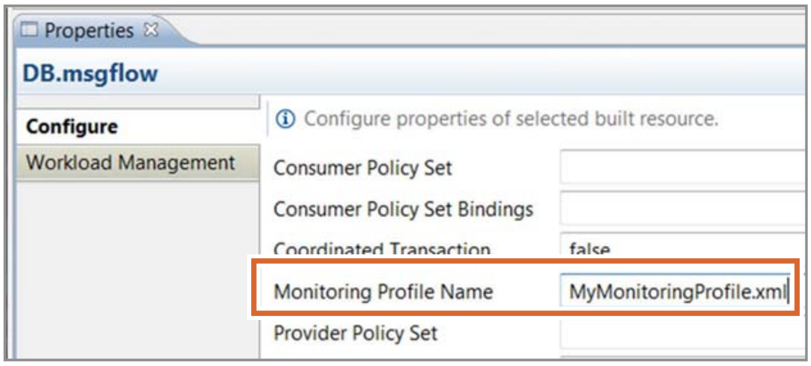
For the complete command syntax, see the product documentation.

WebSphere Education IBM

Apply a monitoring profile to a BAR file

Use the Integration Toolkit BAR File editor to apply a monitoring profile to one or more message flows:

1. Double-click the BAR file to open the BAR File editor.
2. Click the **Manage** tab.
3. Select the message flow to display the **Properties** view.
4. In the **Monitoring Profile Name** field, enter the name of a monitoring profile.
5. Save the BAR file.
6. Deploy the BAR file.
7. Activate monitoring for the flow with `mqsichangeflowmonitoring`



Consumer Policy Set	
Consumer Policy Set Bindings	
Coordinated Transaction	false
Monitoring Profile Name	MyMonitoringProfile.xml
Provider Policy Set	

© Copyright IBM Corporation 2015

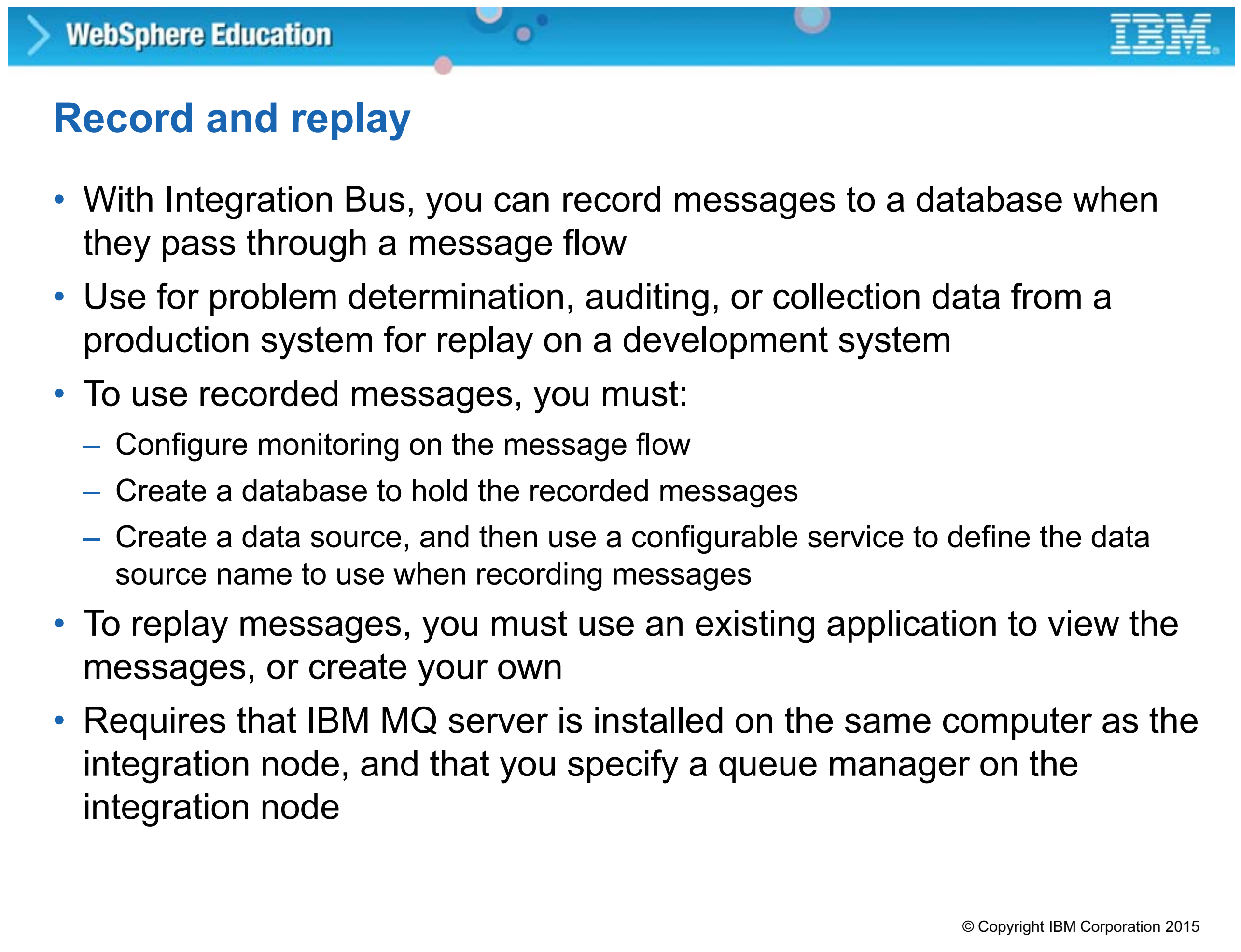
Figure 13-11. Apply a monitoring profile to a BAR file WM6461.0

Notes:

You can apply a monitoring profile by using a command or by modifying the BAR file in the Integration Toolkit.

The figure lists the steps for adding a monitoring profile to a BAR file by using the BAR File editor in the Integration Toolkit.

13.2.Record and replay



WebSphere Education IBM

Record and replay

- With Integration Bus, you can record messages to a database when they pass through a message flow
- Use for problem determination, auditing, or collection data from a production system for replay on a development system
- To use recorded messages, you must:
  – Configure monitoring on the message flow
  – Create a database to hold the recorded messages
  – Create a data source, and then use a configurable service to define the data source name to use when recording messages
- To replay messages, you must use an existing application to view the messages, or create your own
- Requires that IBM MQ server is installed on the same computer as the integration node, and that you specify a queue manager on the integration node

Figure 13-12. Record and replay WM6461.0

Notes:

If you configured your message flow to emit event messages, the monitoring events publish selected message data. With the record and replay feature of IBM Integration Bus, you can view or replay (resubmit for processing) the message data.

Integration Bus event messages are published to a topic on a publish/subscribe broker. The “record” function subscribes to the published monitoring data, and stores the data in a database. You can then view the data through the IBM Integration web interface, or replay it by resending the message to an IBM MQ queue, for example.

Record and replay configuration

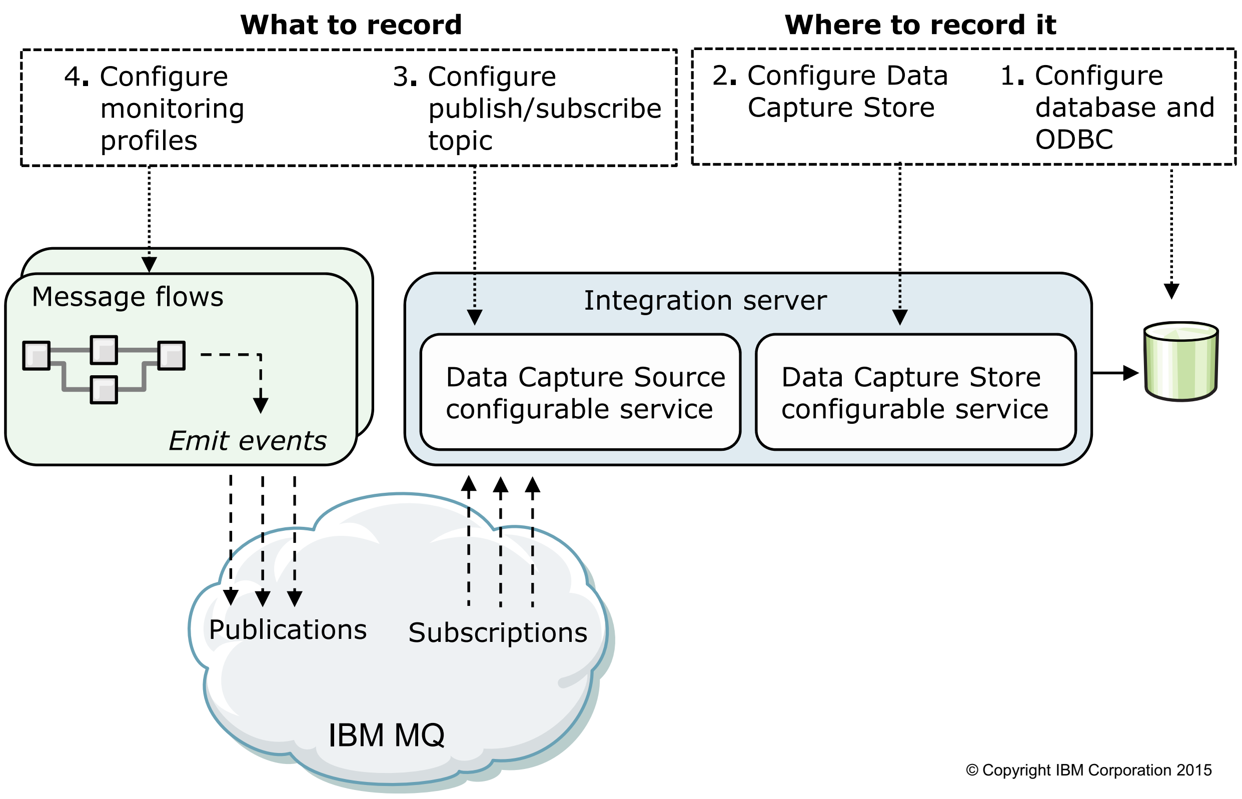

What to record
Where to record it
4. Configure monitoring profiles
3. Configure publish/subscribe topic
2. Configure Data Capture Store
1. Configure database and ODBC
Message flows
Emit events
Integration server
Data Capture Source configurable service
Data Capture Store configurable service
Publications
Subscriptions
IBM MQ
© Copyright IBM Corporation 2015

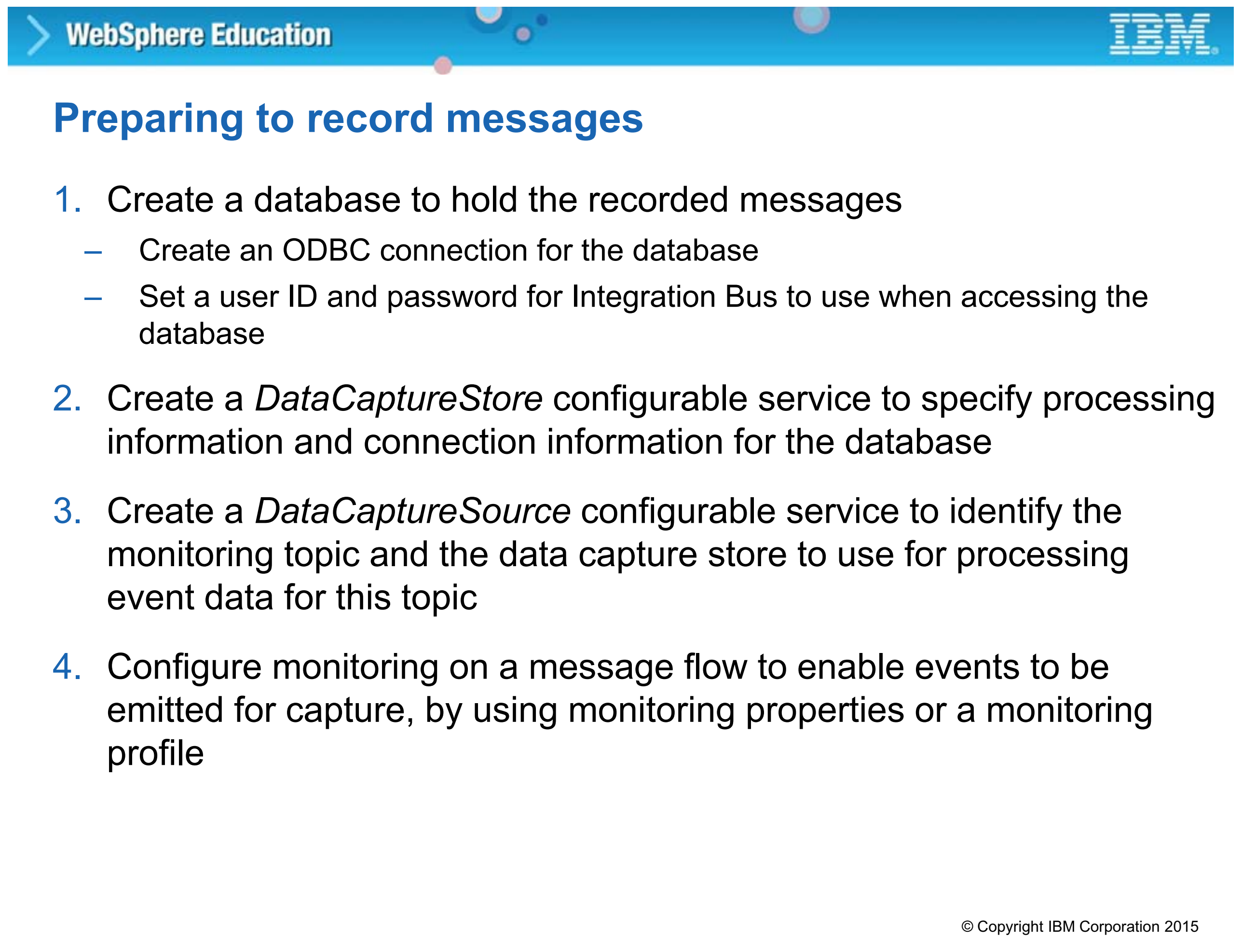
Figure 13-13. Record and replay configuration WM6461.0

Notes:

This figure summarizes the configuration for record and replay.

To configure Integration Bus to record data, complete the following steps. The sequence of these steps is important. If they are not completed exactly as shown, BIP Message BIP2194 is generated on start.

1. Create and configure the database, and define an ODBC definition for the data source name (DSN).
2. To define how and where data is stored, create a DataCaptureStore configurable service. This configurable service specifies the Integration Bus runtime properties for data processing and for connecting to the database.
3. Specify a publish/subscribe topic that identifies the source of the data that you want to capture. To identify the source of the data, create a DataCaptureSource configurable service. You use this configurable service to specify the monitoring topic that identifies the messages flows from which your data comes. This configurable service also identifies the data capture store to use for storing this data.
4. To generate the data that you want to record, configure monitoring on your message flows.



WebSphere Education

Preparing to record messages

1. Create a database to hold the recorded messages
   - Create an ODBC connection for the database
   - Set a user ID and password for Integration Bus to use when accessing the database
2. Create a *DataCaptureStore* configurable service to specify processing information and connection information for the database
3. Create a *DataCaptureSource* configurable service to identify the monitoring topic and the data capture store to use for processing event data for this topic
4. Configure monitoring on a message flow to enable events to be emitted for capture, by using monitoring properties or a monitoring profile

Figure 13-14. Preparing to record messages WM6461.0

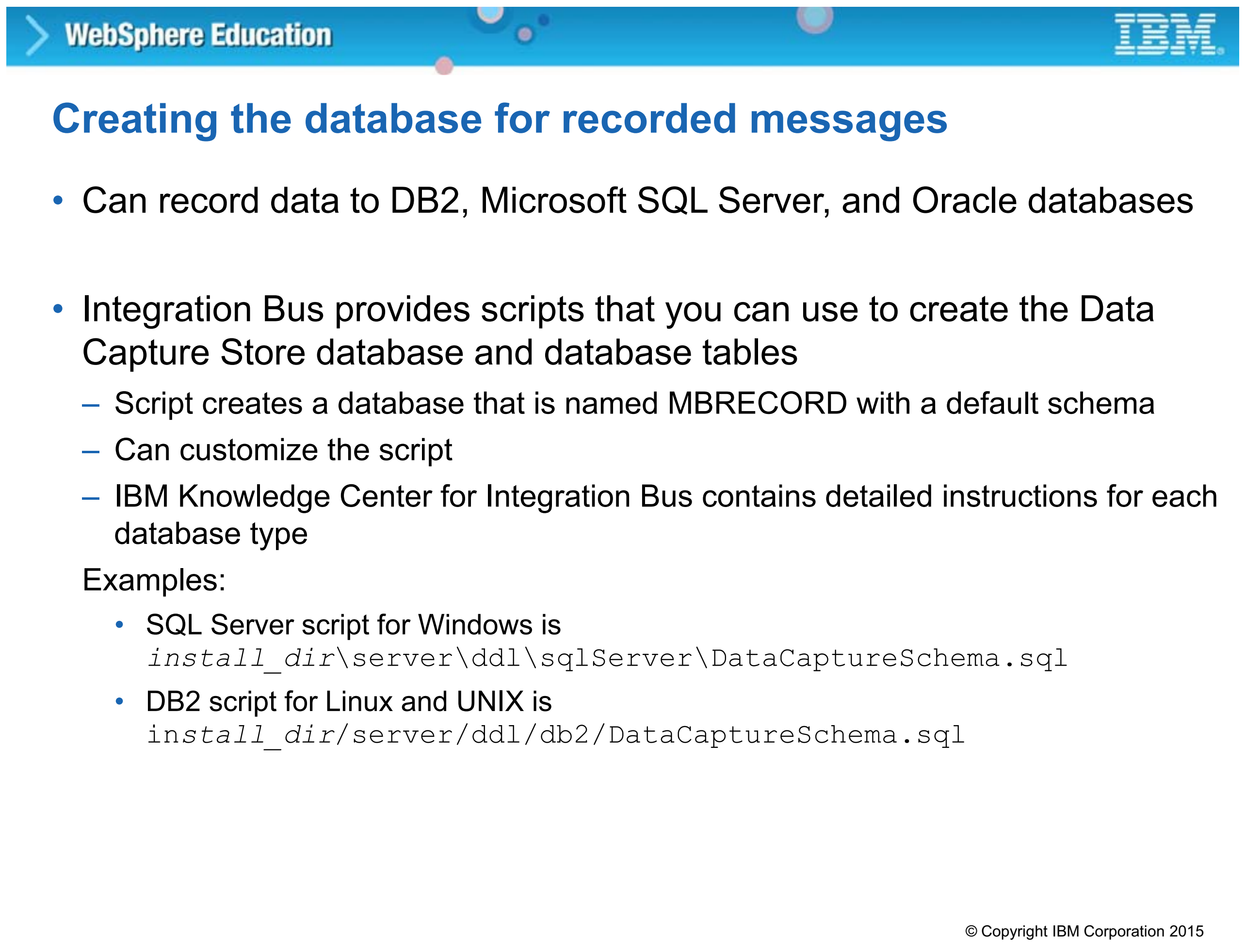
Notes:

The figure lists the steps that must be completed before you can record messages.

First, you must create a database to hold the recorded messages. You must also create an ODBC connection for the database.

You can restrict the users who can view and replay data for an integration node by enabling security. If you do not enable security, all users can complete all actions against an integration node and all integration servers.

The DataCaptureStore configurable service identifies the database that holds the messages. The DataCaptureSource configurable service identifies the monitoring topics and data capture store. Multiple instances of the DataCaptureSource configurable service can use the same DataCaptureStore configurable service.



WebSphere Education IBM

Creating the database for recorded messages

- Can record data to DB2, Microsoft SQL Server, and Oracle databases
- Integration Bus provides scripts that you can use to create the Data Capture Store database and database tables
  - Script creates a database that is named MBRECORD with a default schema
  - Can customize the script
  - IBM Knowledge Center for Integration Bus contains detailed instructions for each database type

  Examples:
  - SQL Server script for Windows is `install_dir\server\ddl\sqlServer\DataCaptureSchema.sql`
  - DB2 script for Linux and UNIX is `install_dir/server/ddl/db2/DataCaptureSchema.sql`

© Copyright IBM Corporation 2015

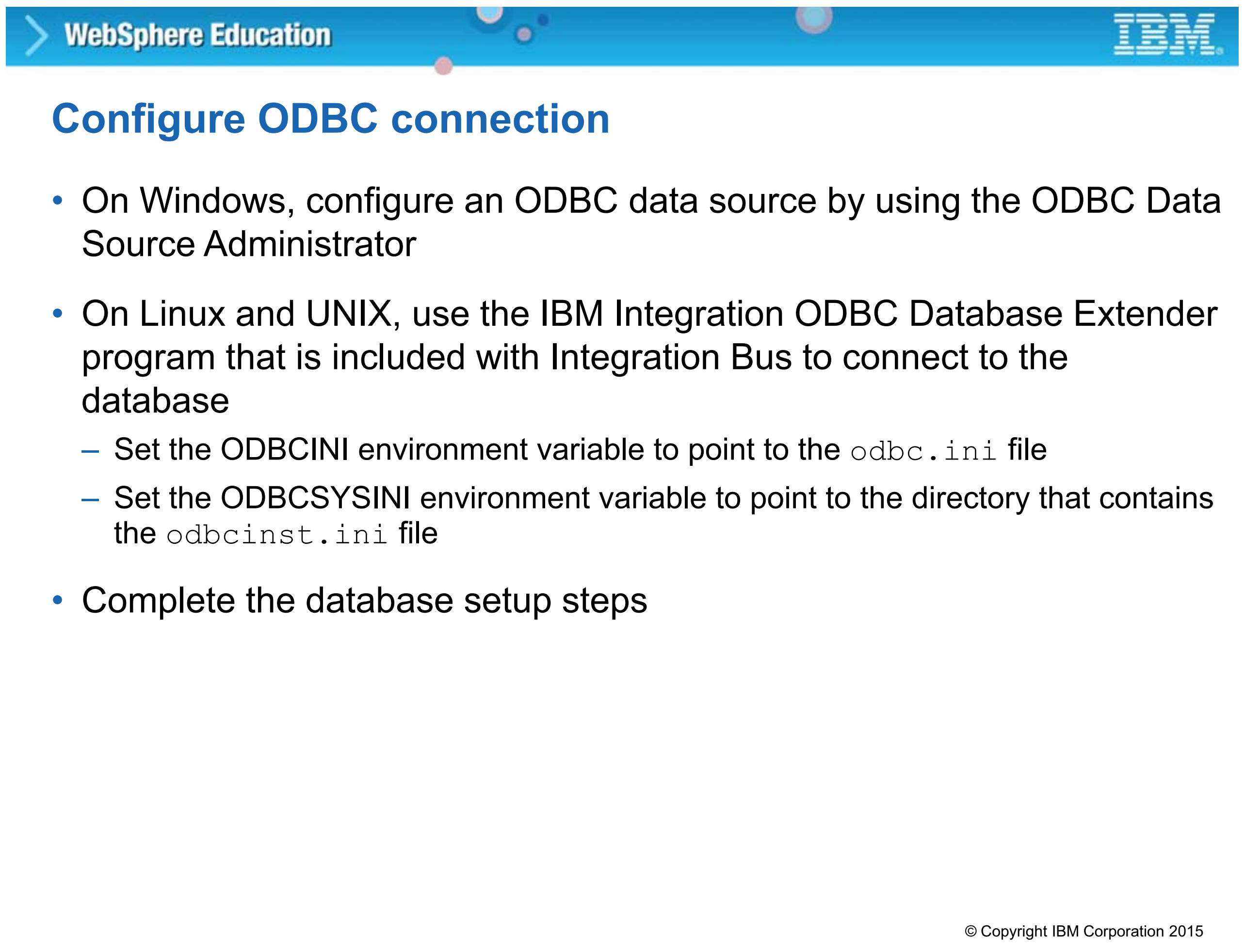
Figure 13-15. Creating the database for recorded messages WM6461.0

Notes:

You can record data to DB2, Microsoft SQL Server, and Oracle databases.

A script is provided with Integration Bus that you can use to create the database and database tables. You can run this script unmodified, or you can customize it.

The script creates a database that is called MBRECORD with a default schema.



Configure ODBC connection

- On Windows, configure an ODBC data source by using the ODBC Data Source Administrator
- On Linux and UNIX, use the IBM Integration ODBC Database Extender program that is included with Integration Bus to connect to the database
  - Set the ODBCINI environment variable to point to the `odbc.ini` file
  - Set the ODBCSYSINI environment variable to point to the directory that contains the `odbcinst.ini` file
- Complete the database setup steps

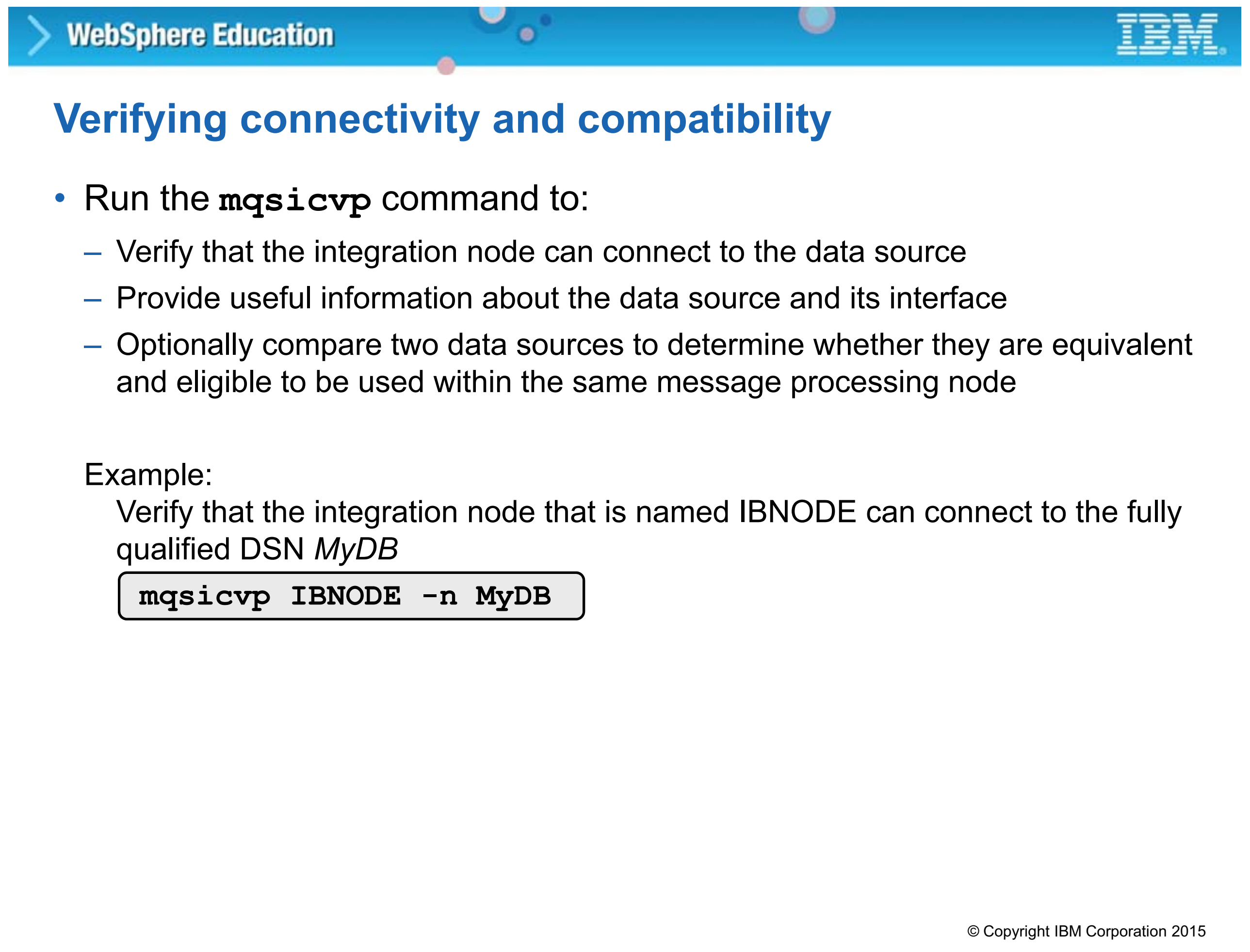
Figure 13-16. Configure ODBC connection WM6461.0

Notes:

After creating the database, you create an ODBC definition to identify the DSN for the database. The steps for creating an ODBC definition vary based on the operating system.

Linux and UNIX use the IBM Integration ODBC Database Extender program to connect to the database. In IBM Integration Bus Version 10, the IBM Integration ODBC Database Extender program is installed as part of the Integration Bus installation.

Test the connection to your database by using the `mqsicvp` command.



WebSphere Education

Verifying connectivity and compatibility

- Run the **`mqsicvp`** command to:
  - Verify that the integration node can connect to the data source
  - Provide useful information about the data source and its interface
  - Optionally compare two data sources to determine whether they are equivalent and eligible to be used within the same message processing node

Example:

Verify that the integration node that is named IBNODE can connect to the fully qualified DSN *MyDB*

```
mqsicvp IBNODE -n MyDB
```

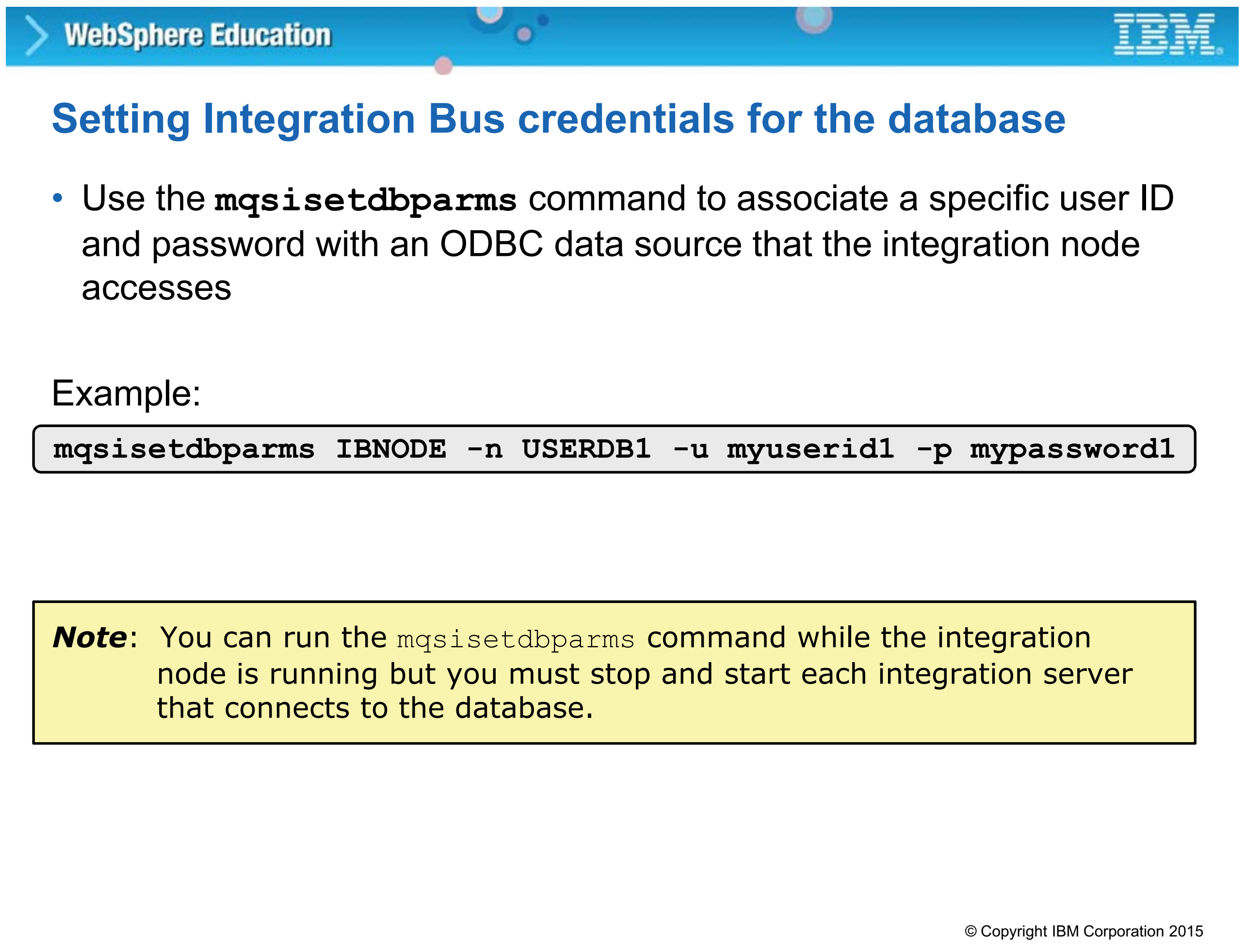
Figure 13-17. Verifying connectivity and compatibility WM6461.0

Notes:

The `mqsicvp` command provides database information and acts as an ODBC connection test tool.

When you use the `mqsicvp` command as an ODBC test tool, the command generates an informational message on successful connection. This message provides the name of the data source, database type, and version. If a secondary data source is supplied, the `mqsicvp` command issues a second informational message about the secondary data source.

In addition to verifying connectivity, the `mqsicvp` command can also determine whether there are any problems with the data types. This command can also determine whether the database is a suitable replacement for another database. This feature is especially useful when moving message flow applications from a development to a production environment.



WebSphere Education

Setting Integration Bus credentials for the database

- Use the **`mqsisetdbparms`** command to associate a specific user ID and password with an ODBC data source that the integration node accesses

Example:

```
mqsisetdbparms IBNODE -n USERDB1 -u myuserid1 -p mypassword1
```

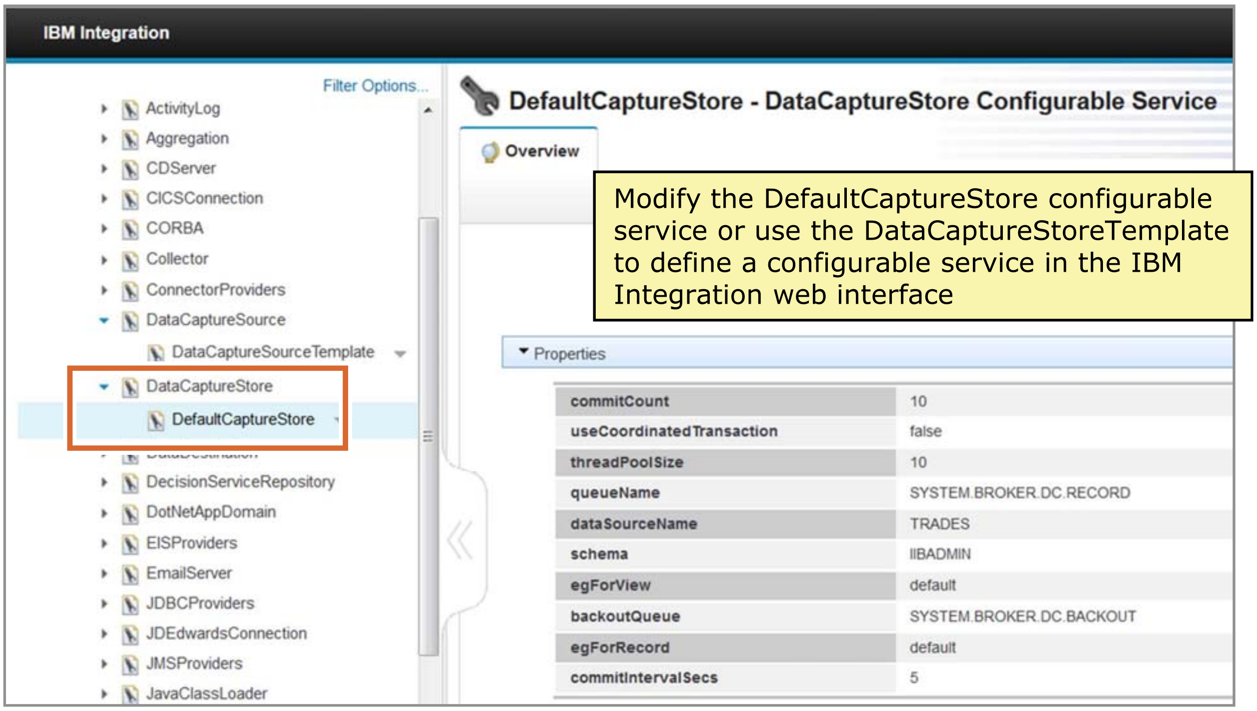
Note: You can run the `mqsisetdbparms` command while the integration node is running but you must stop and start each integration server that connects to the database.

Figure 13-18. Setting Integration Bus credentials for the database WM6461.0

Notes:

You use the `mqsisetdbparms` command to set a user identifier and password for the integration node to use when connecting to the event database.

DataCaptureStore configurable service



DefaultCaptureStore - DataCaptureStore Configurable Service

Modify the DefaultCaptureStore configurable service or use the DataCaptureStoreTemplate to define a configurable service in the IBM Integration web interface

commitCount	10
useCoordinatedTransaction	false
threadPoolSize	10
queueName	SYSTEM.BROKER.DC.RECORD
dataSourceName	TRADES
schema	IIBADMIN
egForView	default
backoutQueue	SYSTEM.BROKER.DC.BACKOUT
egForRecord	default
commitIntervalSecs	5

© Copyright IBM Corporation 2015

Figure 13-19. DataCaptureStore configurable service WM6461.0

Notes:

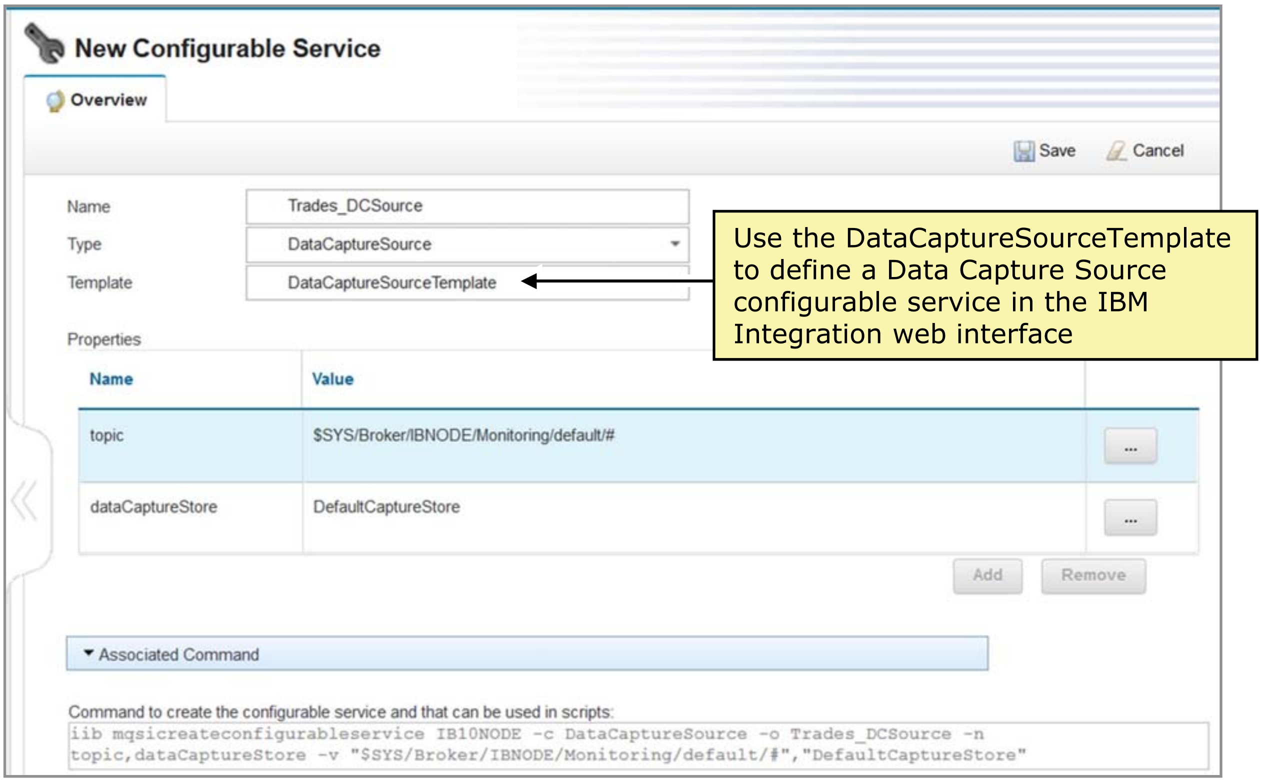
To define how and where data is stored, create a DataCaptureStore configurable service. This configurable service specifies the Integration Bus runtime properties for data processing and for connecting to the event database.

You can use the provided DefaultCaptureStore configurable service or create your own configurable service of type DataCaptureStore. You can use the IBM Integration web user interface to create the configurable service. Alternatively, use the `mqsicreateconfigurableservice` command.

Your record and replay topology can include more than one integration node. If you deploy the message flows for which you want to capture data to one integration node, and use a different integration node to record the data, you must connect the two integration nodes.

WebSphere Education

DataCaptureSource configurable service



Name: Trades_DCSource

Type: DataCaptureSource

Template: DataCaptureSourceTemplate

Properties

Name	Value
topic	$SYS/Broker/IBNODE/Monitoring/default/#
dataCaptureStore	DefaultCaptureStore

Associated Command

Command to create the configurable service and that can be used in scripts:

```
iib mqsicreateconfigurableservice IB10NODE -c DataCaptureSource -o Trades_DCSource -n
topic,dataCaptureStore -v "$SYS/Broker/IBNODE/Monitoring/default/#","DefaultCaptureStore"
```

Use the DataCaptureSourceTemplate to define a Data Capture Source configurable service in the IBM Integration web interface

Figure 13-20. DataCaptureSource configurable service WM6461.0

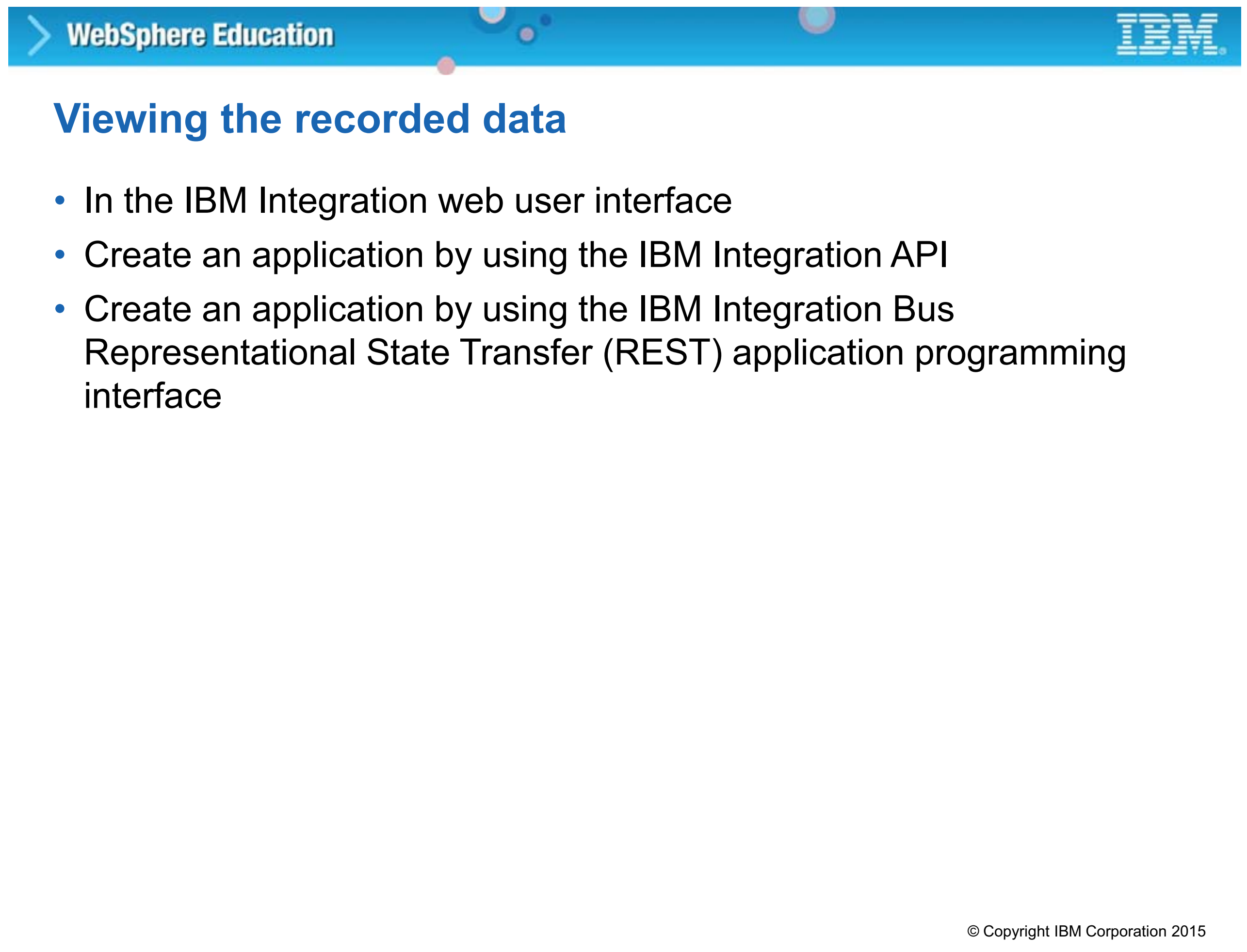
Notes:

To identify the source of the data, create a DataCaptureSource configurable service. You use this configurable service to specify the monitoring topic that identifies the messages flows from which your data comes. You also use this configurable service to identify the data capture store to use for storing this data.

Multiple instances of the DataCaptureSource configurable service can use the same DataCaptureStore configurable service.

You can use the IBM Integration web user interface or the `mqsicreateconfigurableservice` command to create the configurable service. For example:

```
mqsicreateconfigurableservice IBNODE -c DataCaptureSource
-o Trades_DCSOurce -n dataCaptureStore,topic
-v “MyDataCaptureStore”,”$SYS/Broker/IBNODE/Monitoring/IS1/#”
```



Viewing the recorded data

- In the IBM Integration web user interface
- Create an application by using the IBM Integration API
- Create an application by using the IBM Integration Bus Representational State Transfer (REST) application programming interface

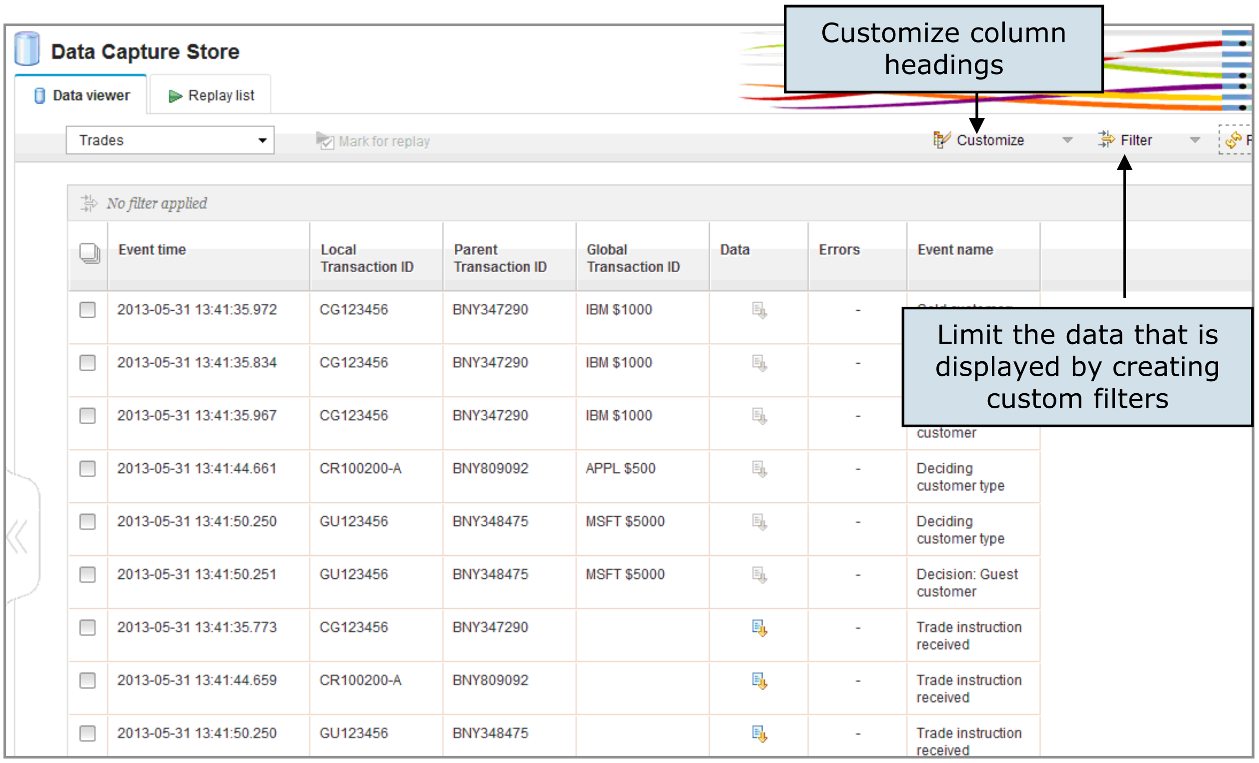
Figure 13-21. Viewing the recorded data WM6461.0

Notes:

You can use the IBM Integration web interface to review the recorded data, update and requeue the message, or delete the message.

You can also use the IBM Integration API or the IBM Integration REST API to write your own message viewer application to fit your requirements.

IBM Integration web user interface Data viewer



	Event time	Local Transaction ID	Parent Transaction ID	Global Transaction ID	Data	Errors	Event name
☐	2013-05-31 13:41:35.972	CG123456	BNY347290	IBM $1000		-	[illegible]
☐	2013-05-31 13:41:35.834	CG123456	BNY347290	IBM $1000		-	[illegible]
☐	2013-05-31 13:41:35.967	CG123456	BNY347290	IBM $1000		-	customer
☐	2013-05-31 13:41:44.661	CR100200-A	BNY809092	APPL $500		-	Deciding customer type
☐	2013-05-31 13:41:50.250	GU123456	BNY348475	MSFT $5000		-	Deciding customer type
☐	2013-05-31 13:41:50.251	GU123456	BNY348475	MSFT $5000		-	Decision: Guest customer
☐	2013-05-31 13:41:35.773	CG123456	BNY347290			-	Trade instruction received
☐	2013-05-31 13:41:44.659	CR100200-A	BNY809092			-	Trade instruction received
☐	2013-05-31 13:41:50.250	GU123456	BNY348475			-	Trade instruction received

© Copyright IBM Corporation 2015

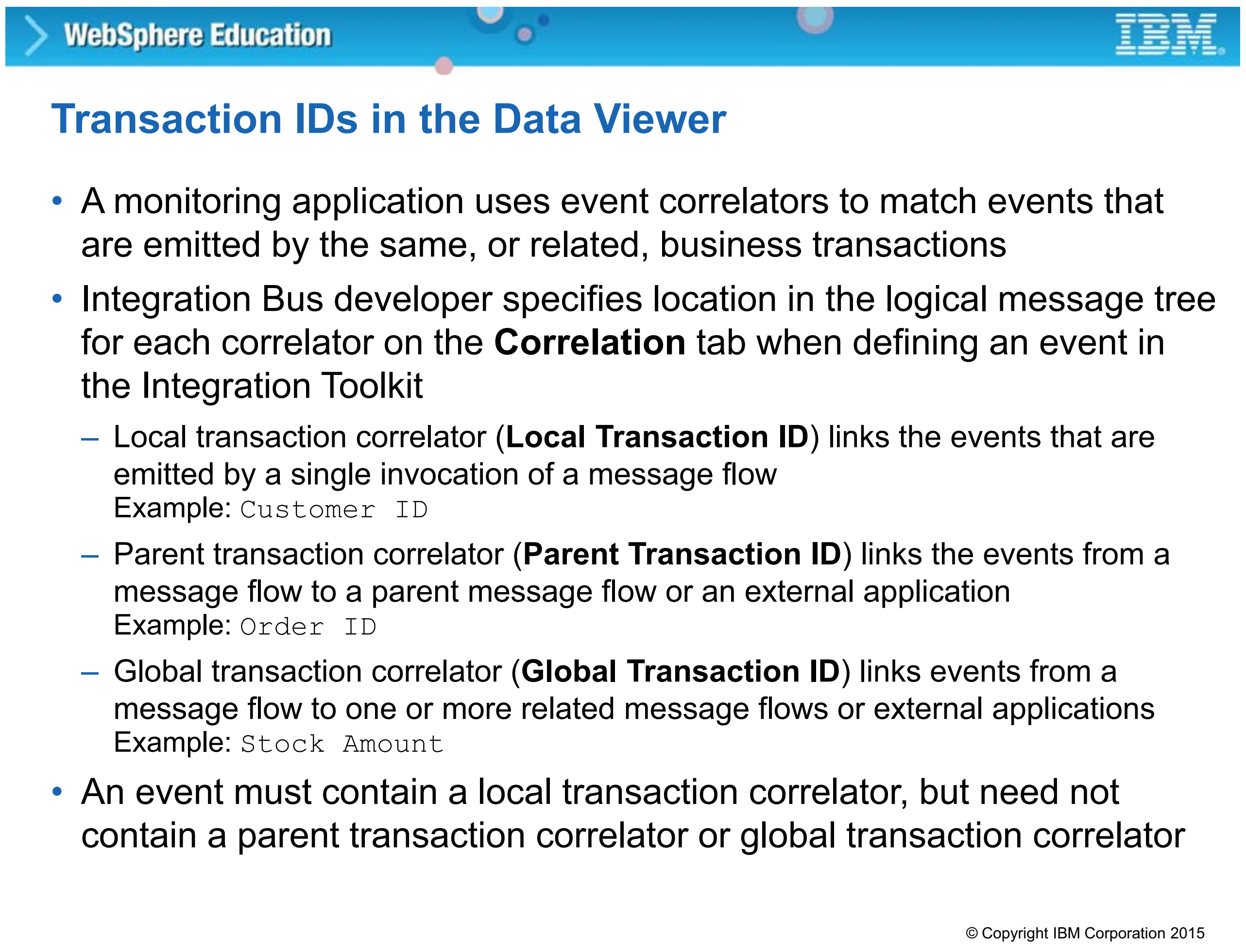
Figure 13-22. IBM Integration web user interface Data viewer WM6461.0

Notes:

The **Data viewer** tab in the IBM Integration web interface contains the record and replay monitoring events. You can customize the headings so that they are more descriptive.

- To change the name of each column, double-click the column name, and then enter another name. These changes are stored in the Integration Registry, and are retained uniquely for each data capture store. All users who view data from the same data capture store see the changes that this user made. If you want to record and view data with different headings, record the events in a separate data capture store.
- You can select or clear any of the recorded fields that are shown in the table.
- You can override the width of the column by using the divider bars.

You can also limit the data that is displayed in the **Data viewer** by using the **Filter** function.



WebSphere Education

Transaction IDs in the Data Viewer

- A monitoring application uses event correlators to match events that are emitted by the same, or related, business transactions
- Integration Bus developer specifies location in the logical message tree for each correlator on the **Correlation** tab when defining an event in the Integration Toolkit
  - Local transaction correlator (**Local Transaction ID**) links the events that are emitted by a single invocation of a message flow
    Example: `Customer ID`
  - Parent transaction correlator (**Parent Transaction ID**) links the events from a message flow to a parent message flow or an external application
    Example: `Order ID`
  - Global transaction correlator (**Global Transaction ID**) links events from a message flow to one or more related message flows or external applications
    Example: `Stock Amount`
- An event must contain a local transaction correlator, but need not contain a parent transaction correlator or global transaction correlator

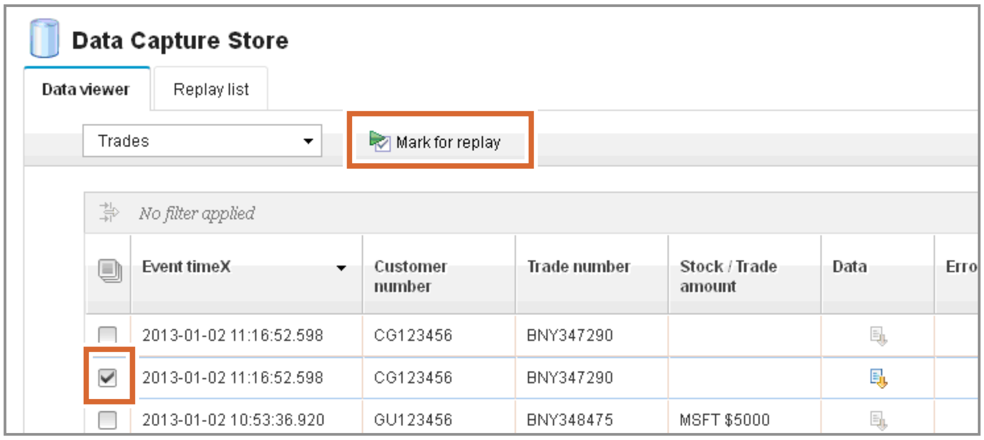
Figure 13-23. Transaction IDs in the Data Viewer WM6461.0

Notes:

WebSphere Education IBM

Replaying messages

1. Select the Data Capture Store that contains the data to replay.
2. Select the messages to replay by selecting the check box next to each message.
3. Click **Mark for replay** to add messages to the replay list.
4. Click the **Replay list** tab.
5. Select the destination for the messages.
6. To replay all messages, click **Replay all**.
   To replay a single message, click the **Replay** icon in the row.



☐	Event timeX	Customer number	Trade number	Stock / Trade amount	Data	Erro
☐	2013-01-02 11:16:52.598	CG123456	BNY347290			
☑	2013-01-02 11:16:52.598	CG123456	BNY347290			
☐	2013-01-02 10:53:36.920	GU123456	BNY348475	MSFT $5000		

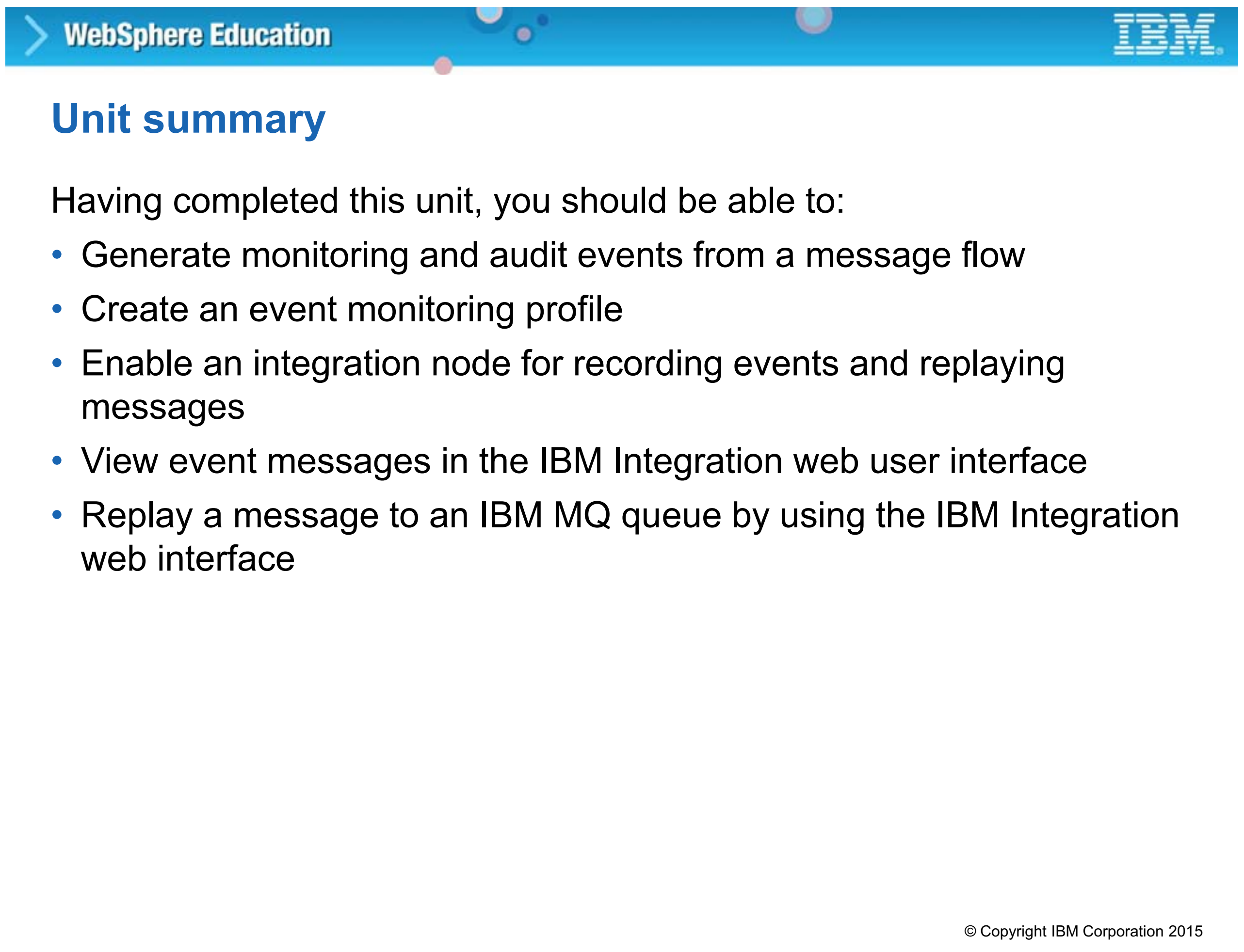
Figure 13-24. Replaying messages WM6461.0

Notes:

The figure lists the steps for replaying messages by using the IBM Integration web interface.

Each row that is displayed represents a recorded message. To sort these rows into a particular order, click a column heading.

You can also use a filter to help find rows in which you are interested. Filtering is case-sensitive and allows the use of wildcards. To search for an exact match, enclose the search string in quotation marks. You can also search for a substring, and you can use an asterisk (*) to match zero or more characters.



WebSphere Education
IBM

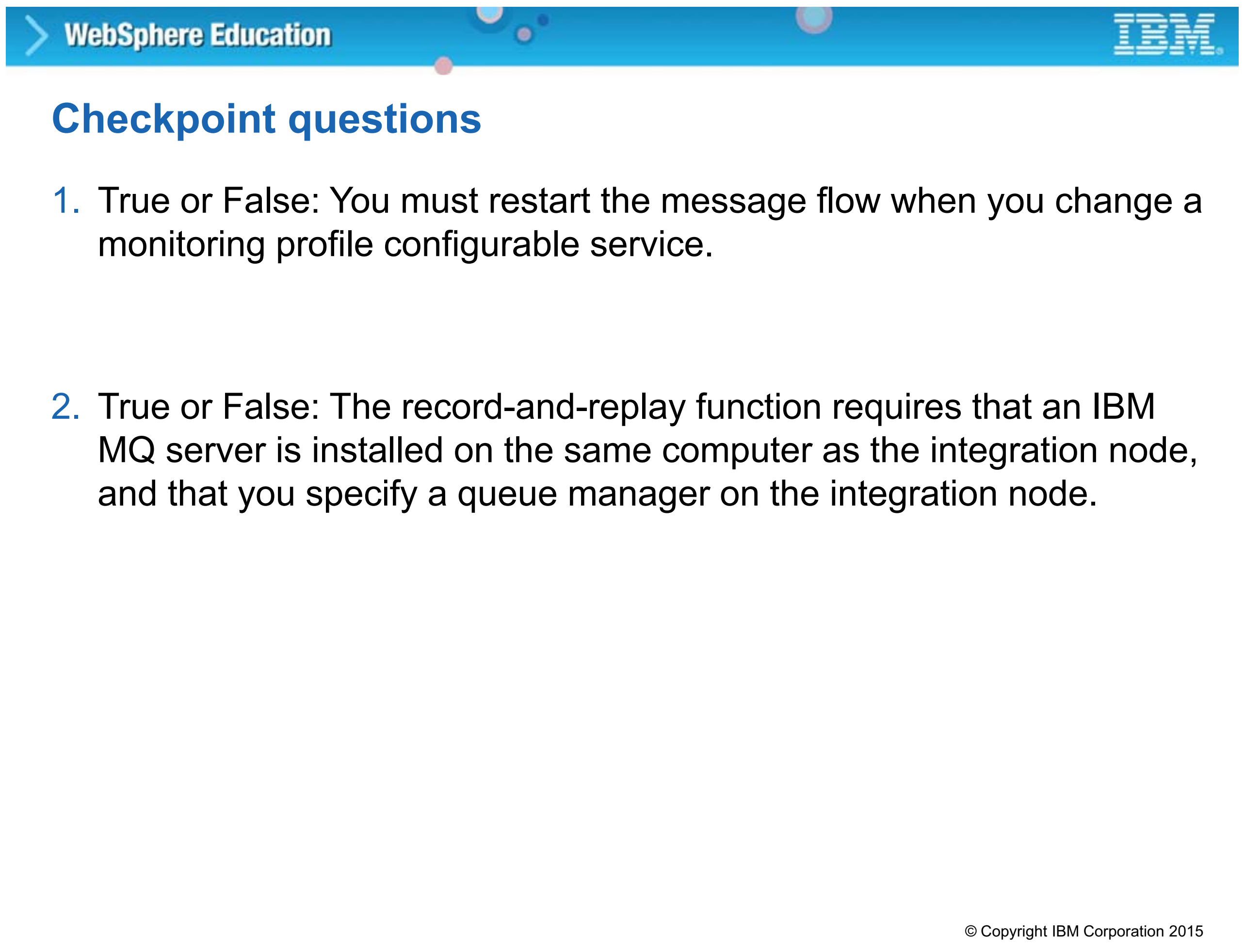
Unit summary

Having completed this unit, you should be able to:

- Generate monitoring and audit events from a message flow
- Create an event monitoring profile
- Enable an integration node for recording events and replaying messages
- View event messages in the IBM Integration web user interface
- Replay a message to an IBM MQ queue by using the IBM Integration web interface

Figure 13-25. Unit summary WM6461.0

Notes:



Checkpoint questions

1. True or False: You must restart the message flow when you change a monitoring profile configurable service.

2. True or False: The record-and-replay function requires that an IBM MQ server is installed on the same computer as the integration node, and that you specify a queue manager on the integration node.

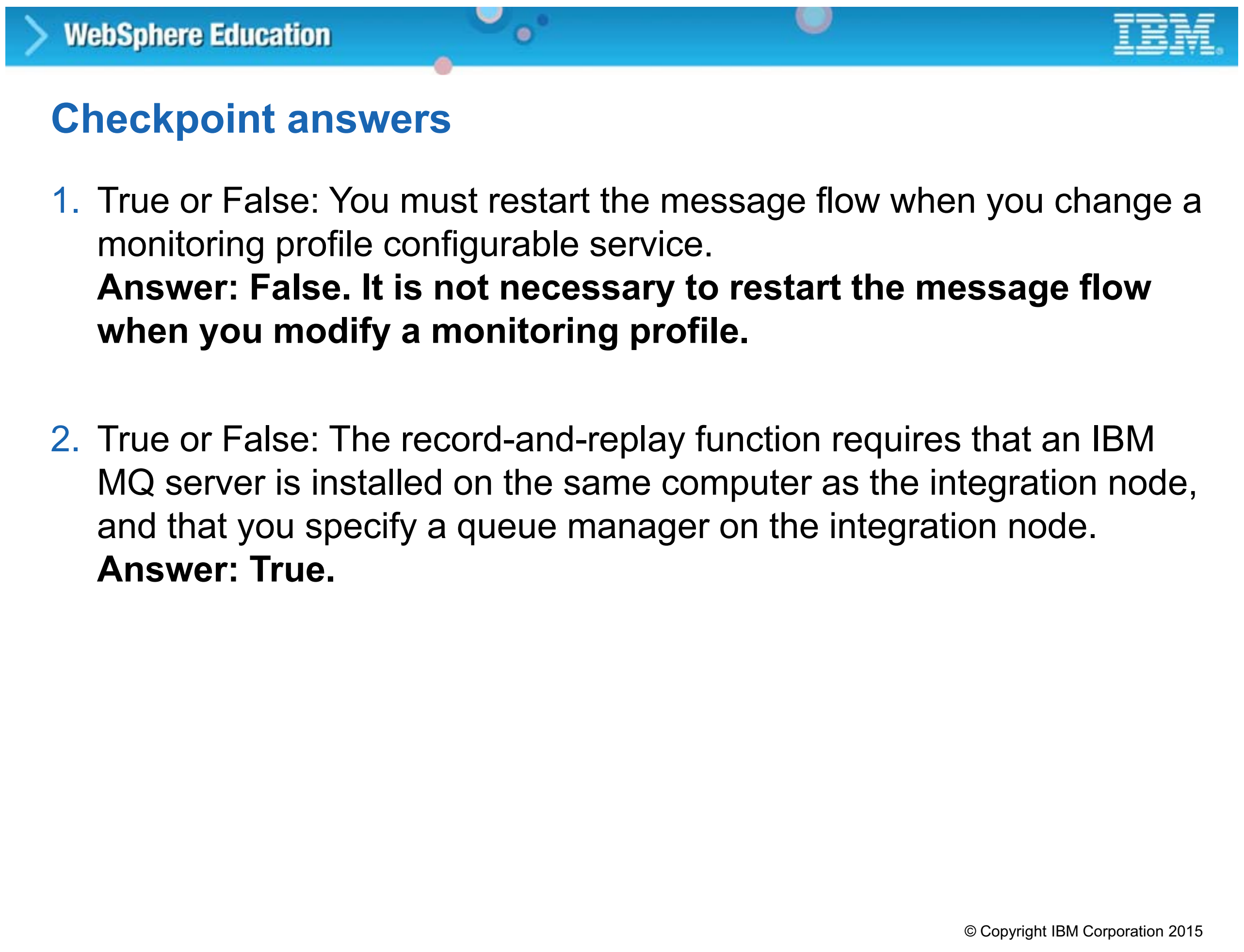
Figure 13-26. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education

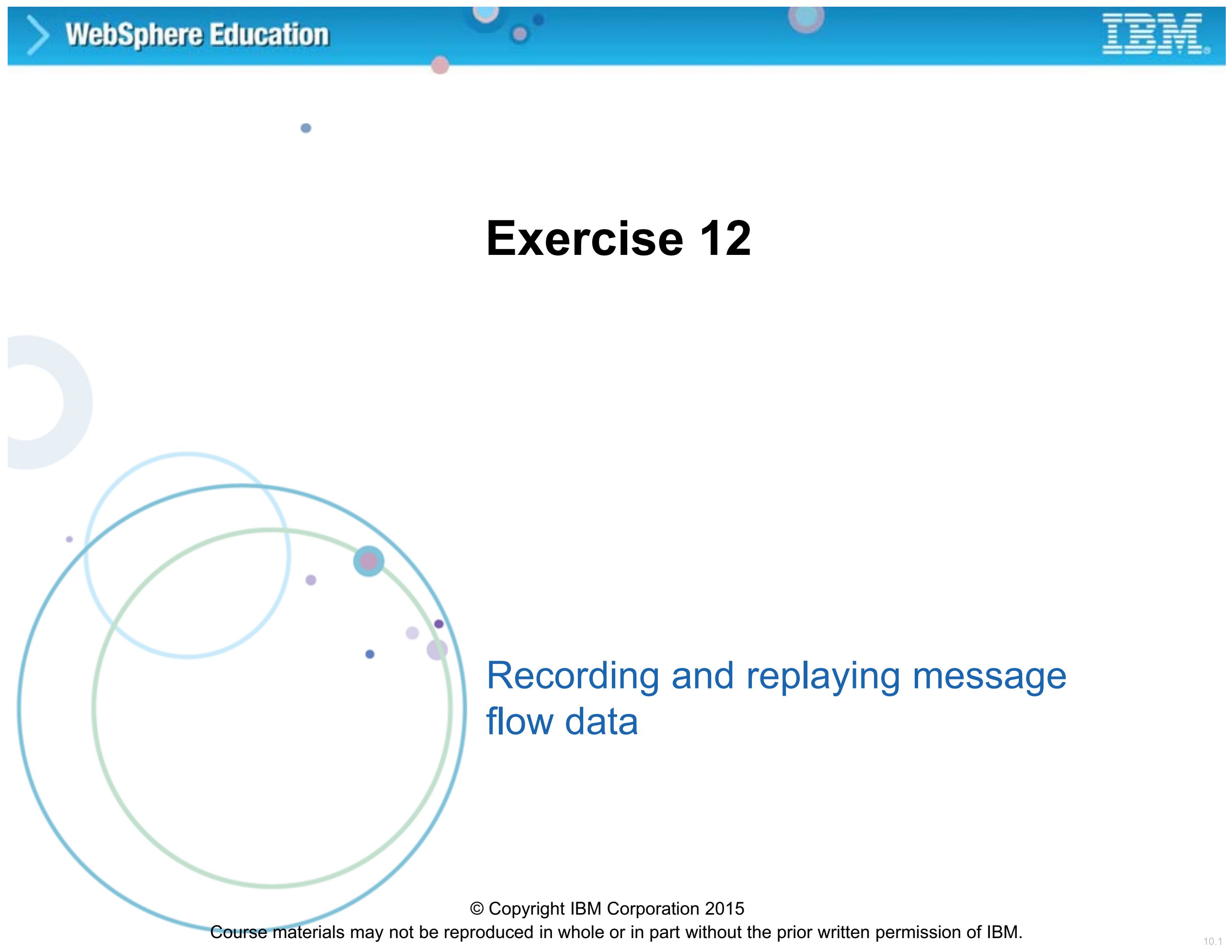
Checkpoint answers

1. True or False: You must restart the message flow when you change a monitoring profile configurable service.
   Answer: False. It is not necessary to restart the message flow when you modify a monitoring profile.

2. True or False: The record-and-replay function requires that an IBM MQ server is installed on the same computer as the integration node, and that you specify a queue manager on the integration node.
   Answer: True.

Figure 13-27. Checkpoint answers WM6461.0

Notes:



WebSphere Education

Exercise 12

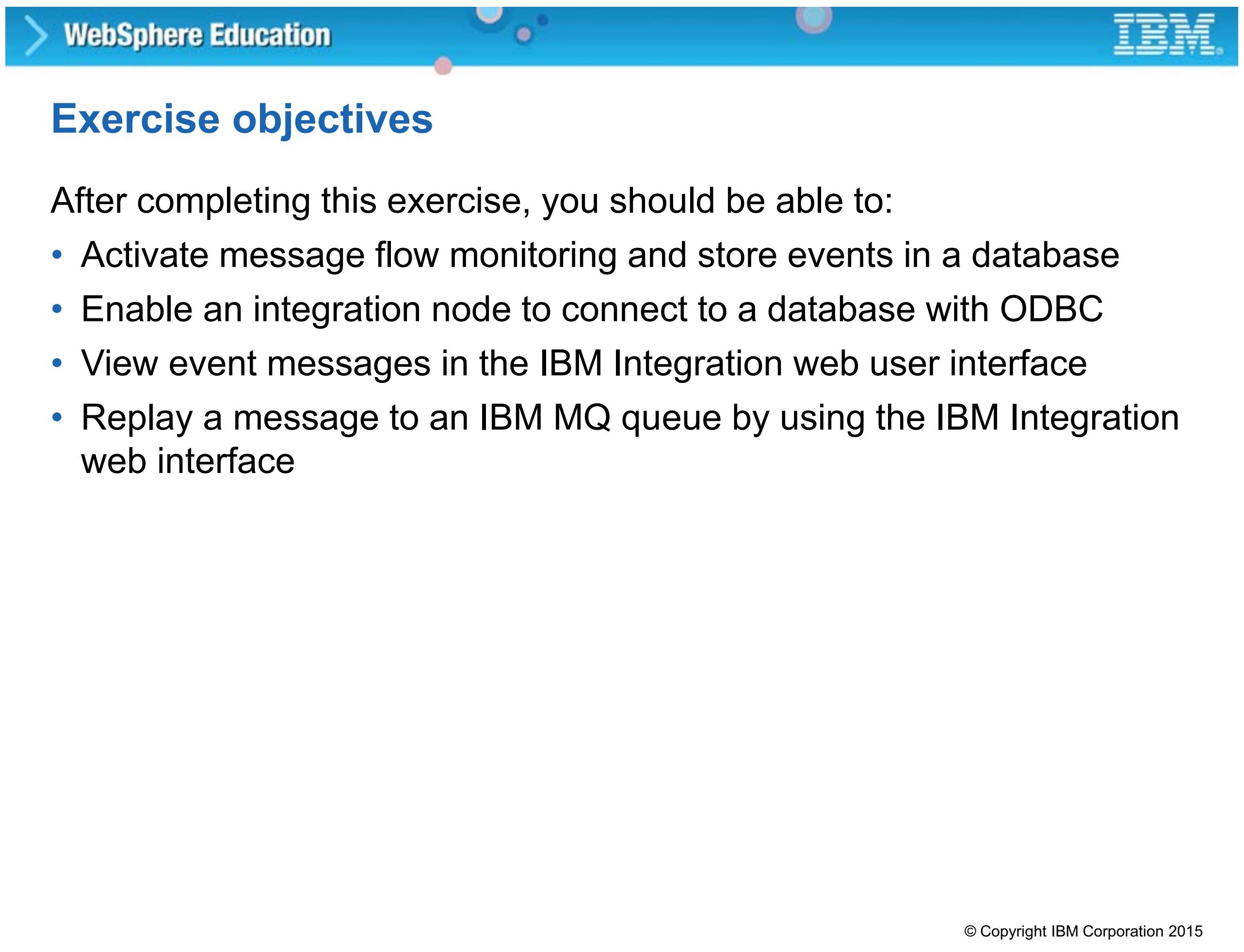
Recording and replaying message flow data

10.1

Figure 13-28. Exercise 12 WM6461.0

Notes:

In this exercise, you define message flow events and then record them in a database. You also replay messages and capture and report failed events.



WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Activate message flow monitoring and store events in a database
- Enable an integration node to connect to a database with ODBC
- View event messages in the IBM Integration web user interface
- Replay a message to an IBM MQ queue by using the IBM Integration web interface

Figure 13-29. Exercise objectives WM6461.0

Notes:

See the *Student Exercise Guide* for detailed instructions.

Unit 14. Extending IBM Integration Bus

What this unit is about

In this unit, you learn how to configure the IBM Integration Bus runtime environment to allow message flows to connect to a database, use JMS as a message transport, and read and write files with secure FTP (SFTP). The unit also describes how to use SupportPacs to extend IBM Integration Bus capability.

What you should be able to do

After completing this unit, you should be able to:

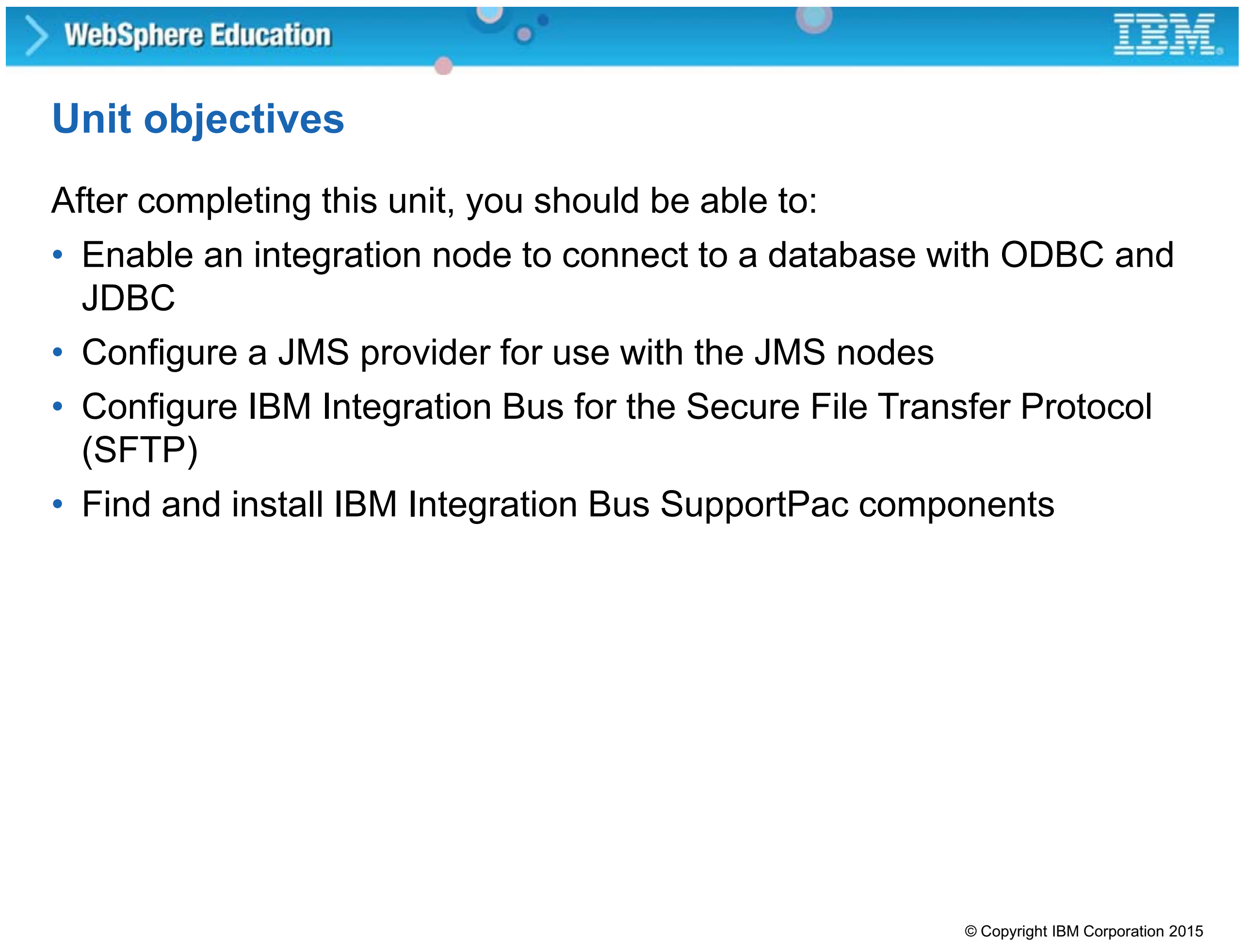
- Enable an integration node to connect to a database with ODBC and JDBC
- Configure a JMS provider for use with the JMS nodes
- Configure IBM Integration Bus for the Secure File Transfer Protocol (SFTP)
- Find and install IBM Integration Bus SupportPac components

How you will check your progress

- Checkpoint questions

References

IBM Knowledge Center for IBM Integration Bus V10



WebSphere Education IBM

Unit objectives

After completing this unit, you should be able to:

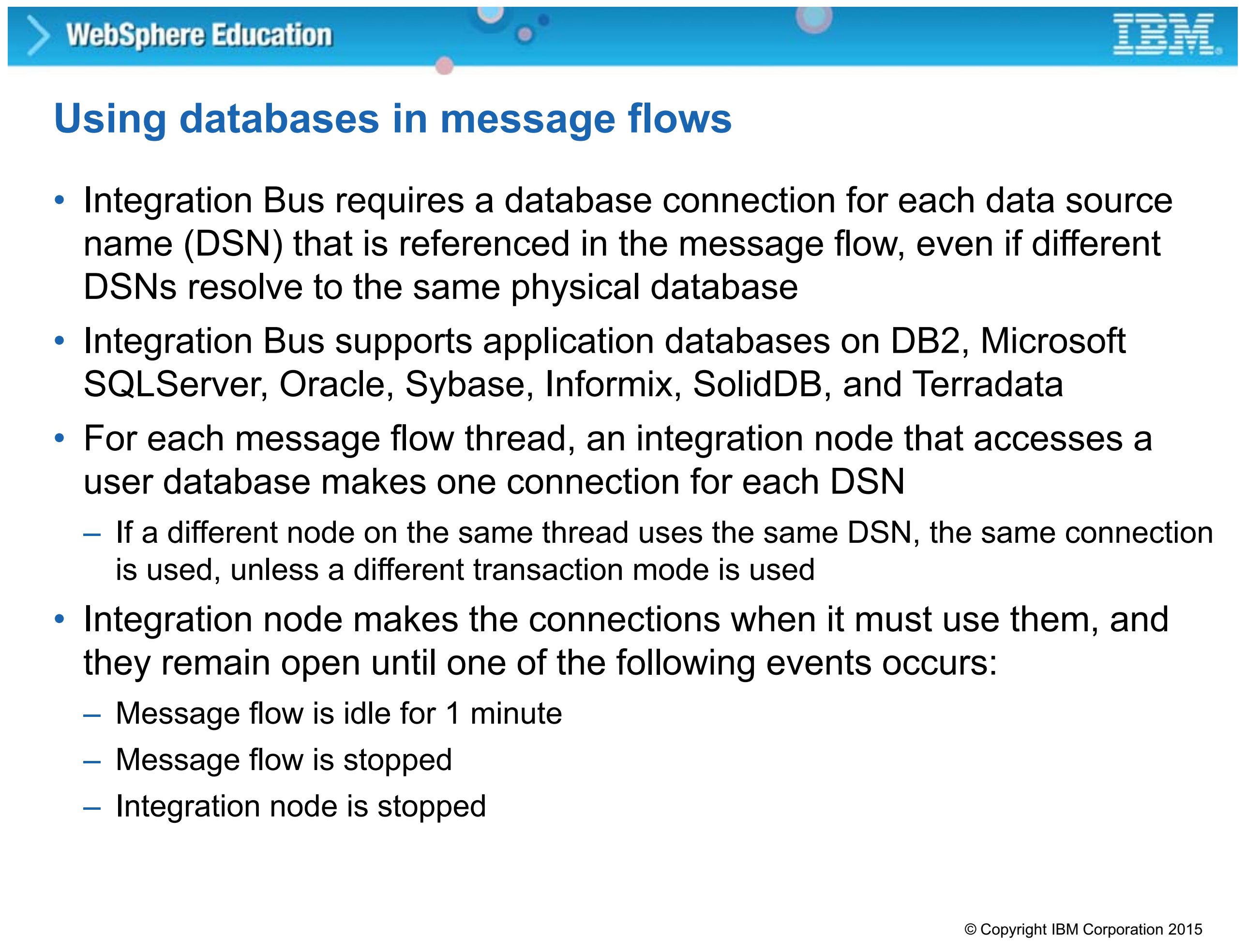
- Enable an integration node to connect to a database with ODBC and JDBC
- Configure a JMS provider for use with the JMS nodes
- Configure IBM Integration Bus for the Secure File Transfer Protocol (SFTP)
- Find and install IBM Integration Bus SupportPac components

© Copyright IBM Corporation 2015

Figure 14-1. Unit objectives WM6461.0

Notes:

14.1. Database connectivity



Using databases in message flows

- Integration Bus requires a database connection for each data source name (DSN) that is referenced in the message flow, even if different DSNs resolve to the same physical database
- Integration Bus supports application databases on DB2, Microsoft SQLServer, Oracle, Sybase, Informix, SolidDB, and Terradata
- For each message flow thread, an integration node that accesses a user database makes one connection for each DSN
  - If a different node on the same thread uses the same DSN, the same connection is used, unless a different transaction mode is used
- Integration node makes the connections when it must use them, and they remain open until one of the following events occurs:
  - Message flow is idle for 1 minute
  - Message flow is stopped
  - Integration node is stopped

Figure 14-2. Using databases in message flows WM6461.0

Notes:

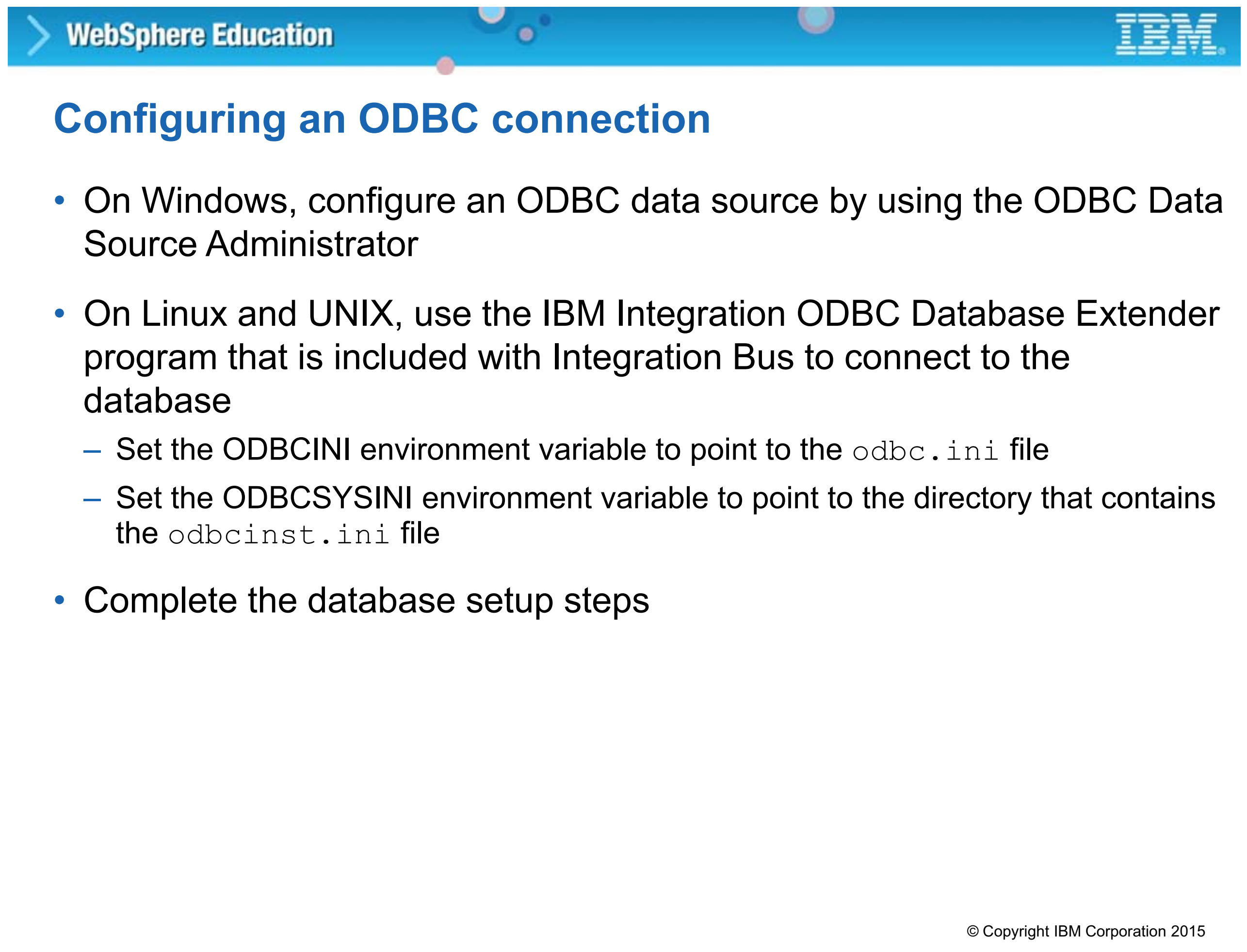
Developers can create and configure message flows that access databases to enhance or influence the operation of the message flow. A message flow can also modify the contents of a database by adding new information or removing or replacing existing information.

Accessing user databases in a message flow requires the database definitions in the workspace. The IBM Integration Toolkit can connect to the database, discover the definition (schema), and then store it in the workspace. The administrator must define a data source name (DSN) for each database and also configure any security requirements.

When you start an integration node, and while it is running, it opens connections to application databases. The integration node automatically handles connection pooling and manages the message flow as a single transaction. The integration node makes the connections when it needs to use them, and they remain open until a flow has no work. When a message flow is idle for approximately 1 minute, or if the message flow completes, the integration node closes the connection.

Determine the number of database connections that the integration node requires, for capacity-planning and resource-planning purposes. The integration node makes a database connection through the ODBC data source name (DSN). A connection is made for each DSN even if a different DSN resolves to the same physical database. For example, by default DB2 limits the number of concurrent connections to a database to the value of the `maxappls` configuration parameter. The default for `maxappls` is 40. If the number of connections that the integration node might require exceeds the value for `maxappls`, increase the value of this parameter.

Always check the database documentation for information about connections and the limits or restrictions that might apply.



Configuring an ODBC connection

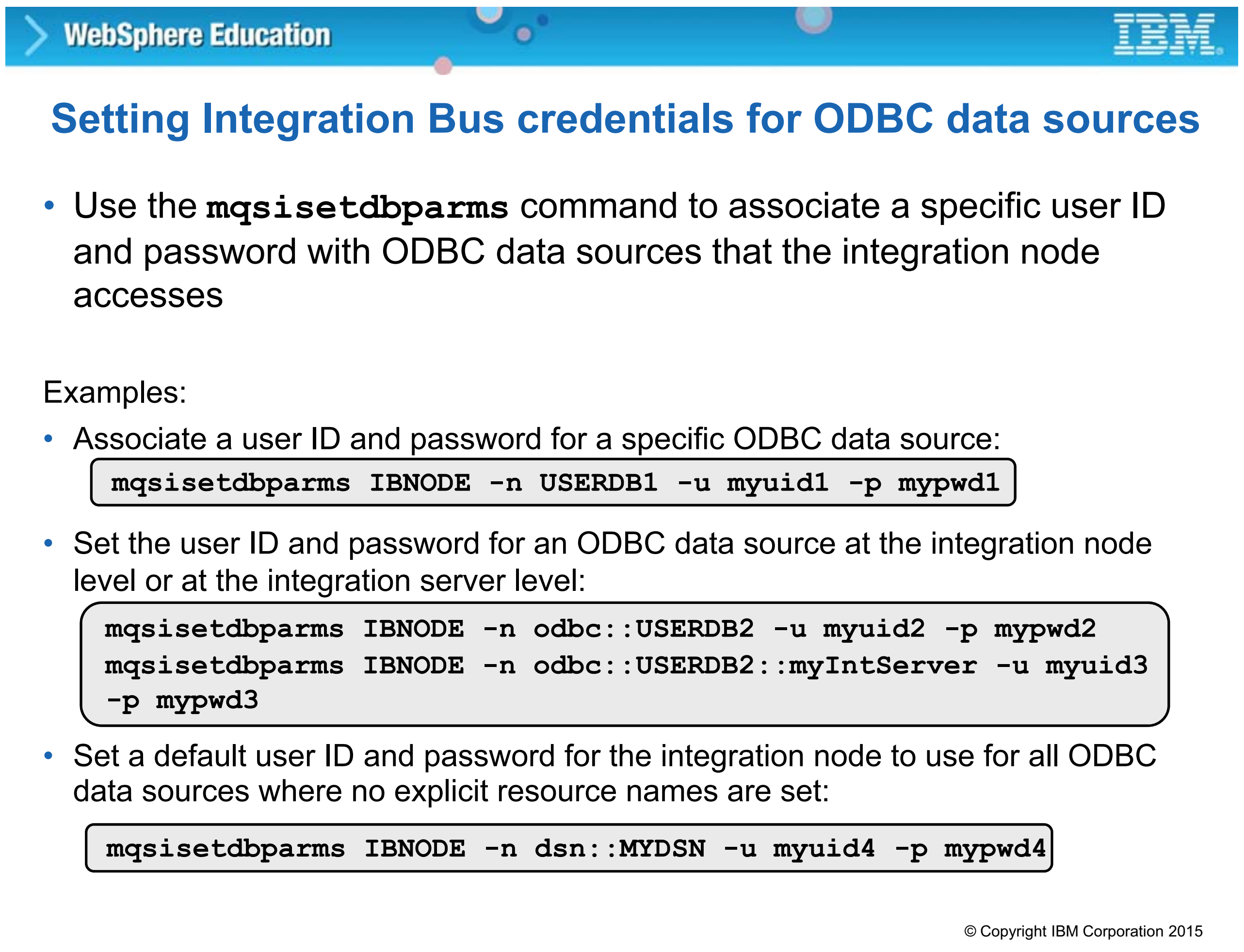
- On Windows, configure an ODBC data source by using the ODBC Data Source Administrator
- On Linux and UNIX, use the IBM Integration ODBC Database Extender program that is included with Integration Bus to connect to the database
  - Set the ODBCINI environment variable to point to the `odbc.ini` file
  - Set the ODBCSYSINI environment variable to point to the directory that contains the `odbcinst.ini` file
- Complete the database setup steps

© Copyright IBM Corporation 2015

Figure 14-3. Configuring an ODBC connection WM6461.0

Notes:

Integration Bus provides different connection types to application databases, depending on the nodes. Use ODBC and integration node managed JDBC type 4 connections when possible to save on manual programming.



WebSphere Education

Setting Integration Bus credentials for ODBC data sources

- Use the **`mqsisetdbparms`** command to associate a specific user ID and password with ODBC data sources that the integration node accesses

Examples:

- Associate a user ID and password for a specific ODBC data source:

```
mqsisetdbparms IBNODE -n USERDB1 -u myuid1 -p mypwd1
```

- Set the user ID and password for an ODBC data source at the integration node level or at the integration server level:

```
mqsisetdbparms IBNODE -n odbc::USERDB2 -u myuid2 -p mypwd2
mqsisetdbparms IBNODE -n odbc::USERDB2::myIntServer -u myuid3
-p mypwd3
```

- Set a default user ID and password for the integration node to use for all ODBC data sources where no explicit resource names are set:

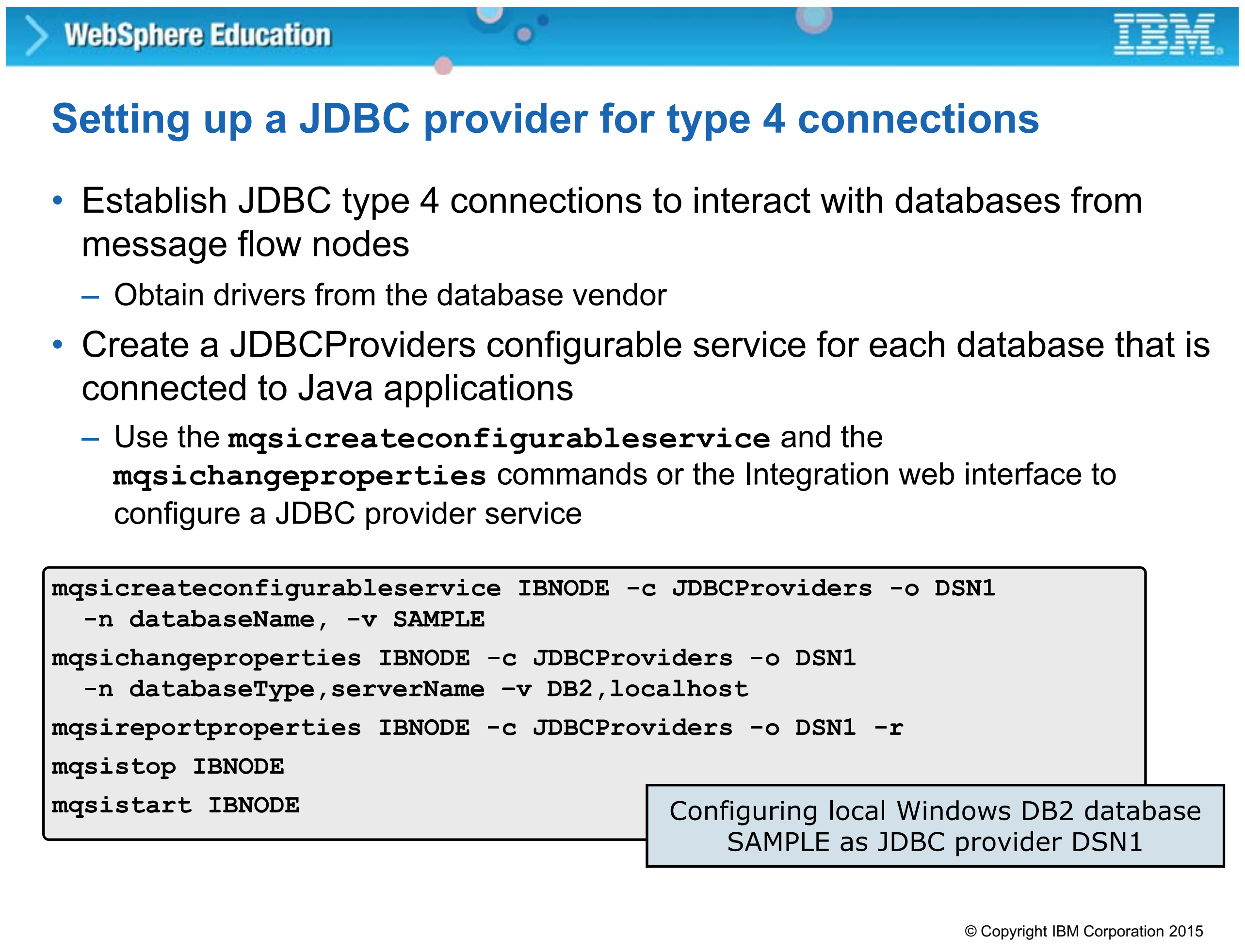
```
mqsisetdbparms IBNODE -n dsn::MYDSN -u myuid4 -p mypwd4
```

Figure 14-4. Setting Integration Bus credentials for ODBC data sources WM6461.0

Notes:

You can use the `mqsisetdbparms` command to associate a specific user ID and password with an ODBC data source.

The figure includes some examples of the `mqsisetdbparms` command.



WebSphere Education

Setting up a JDBC provider for type 4 connections

- Establish JDBC type 4 connections to interact with databases from message flow nodes
  - Obtain drivers from the database vendor
- Create a JDBCProviders configurable service for each database that is connected to Java applications
  - Use the **`mqsicreateconfigurableservice`** and the **`mqsichangeproperties`** commands or the Integration web interface to configure a JDBC provider service

```
mqsicreateconfigurableservice IBNODE -c JDBCProviders -o DSN1
  -n databaseName, -v SAMPLE
mqsichangeproperties IBNODE -c JDBCProviders -o DSN1
  -n databaseType,serverName –v DB2,localhost
mqsireportproperties IBNODE -c JDBCProviders -o DSN1 -r
mqsistop IBNODE
mqsistart IBNODE
```

Configuring local Windows DB2 database SAMPLE as JDBC provider DSN1

Figure 14-5. Setting up a JDBC provider for type 4 connections WM6461.0

Notes:

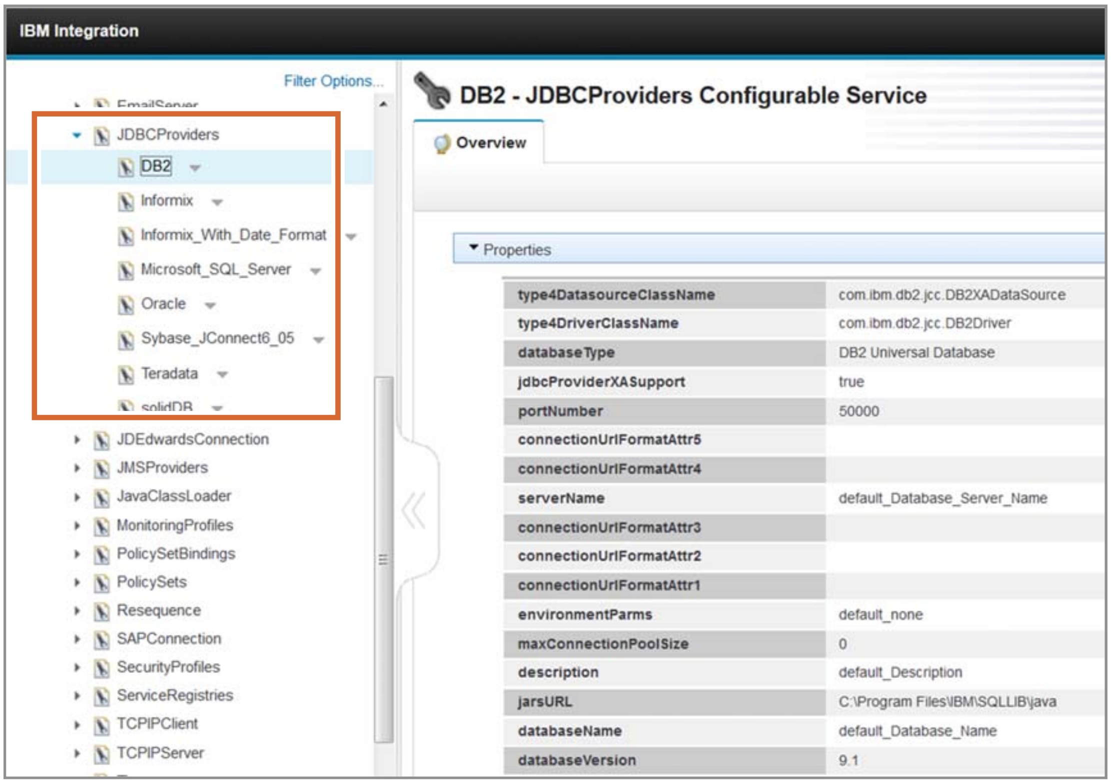
When the message flow includes a message flow node that interacts with a database, the integration node must establish connections with the database. Define a JDBCProvider configurable service to provide the integration node with the information that it needs to complete the connection.

The example in the figure shows how to configure a local Windows DB2 database SAMPLE as JDBCProvider DSN1.

You might need to change more properties specific to your database and environment. Some values depend on how and where the database is installed. For example, the property `jarsURL` identifies the location of the JAR files that the database vendor provides.

WebSphere Education

JDBCProviders configurable services


IBM Integration

Filter Options...

- EmailServer
- JDBCProviders
  - DB2
  - Informix
  - Informix_With_Date_Format
  - Microsoft_SQL_Server
  - Oracle
  - Sybase_JConnect6_05
  - Teradata
  - solidDB
- JDEdwardsConnection
- JMSProviders
- JavaClassLoader
- MonitoringProfiles
- PolicySetBindings
- PolicySets
- Resequence
- SAPConnection
- SecurityProfiles
- ServiceRegistries
- TCPIPClient
- TCPIPServer

DB2 - JDBCProviders Configurable Service

Overview

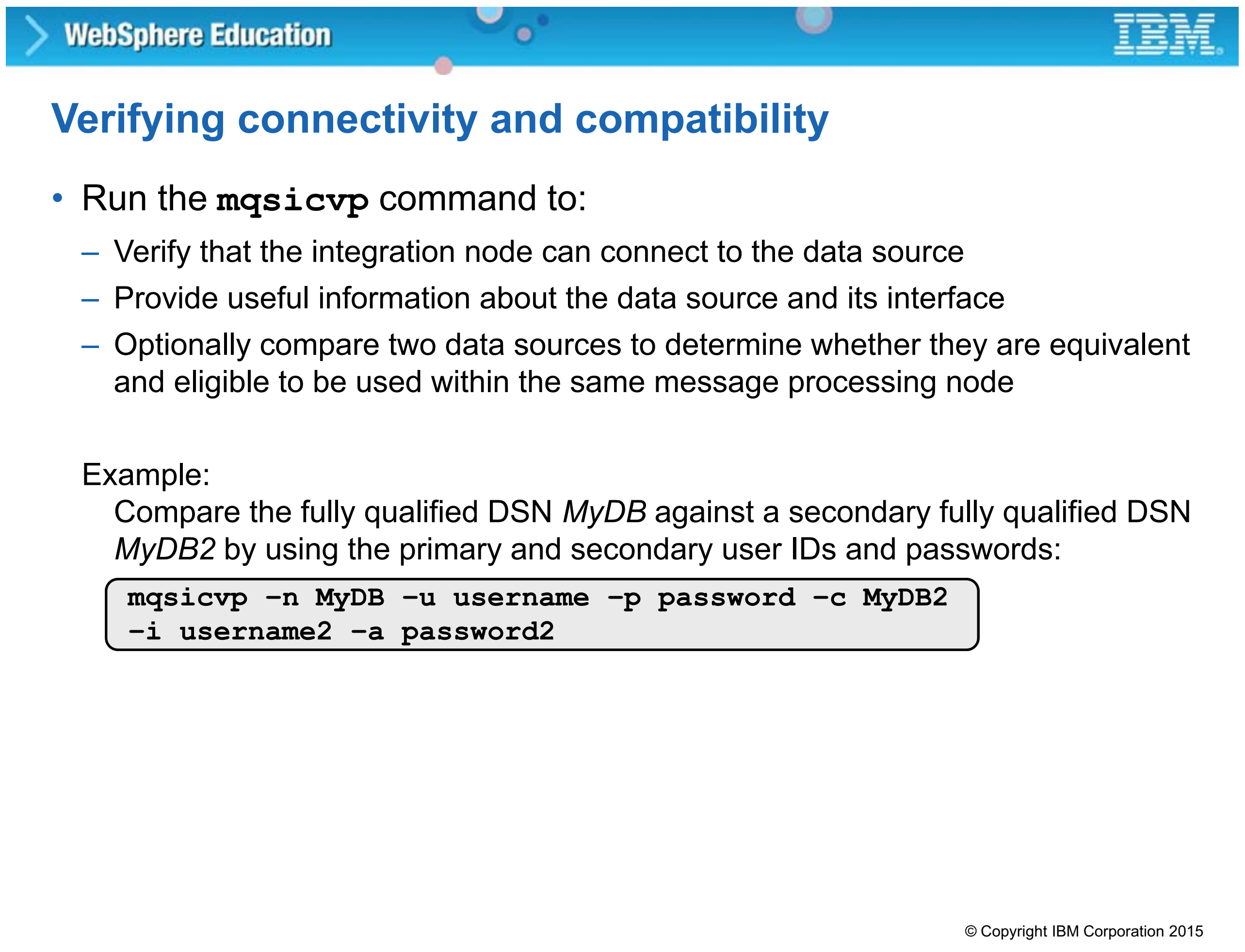
Properties

type4DatasourceClassName	com.ibm.db2.jcc.DB2XADataSource
type4DriverClassName	com.ibm.db2.jcc.DB2Driver
databaseType	DB2 Universal Database
jdbcProviderXASupport	true
portNumber	50000
connectionUrlFormatAttr5	
connectionUrlFormatAttr4	
serverName	default_Database_Server_Name
connectionUrlFormatAttr3	
connectionUrlFormatAttr2	
connectionUrlFormatAttr1	
environmentParms	default_none
maxConnectionPoolSize	0
description	default_Description
jarsURL	C:\Program Files\IBM\SQLLIB\java
databaseName	default_Database_Name
databaseVersion	9.1

Figure 14-6. JDBCProviders configurable services WM6461.0

Notes:

You can configure and manage the JDBCProviders configurable services in the IBM Integration web interface.



WebSphere Education

Verifying connectivity and compatibility

- Run the **`mqsicvp`** command to:
  - Verify that the integration node can connect to the data source
  - Provide useful information about the data source and its interface
  - Optionally compare two data sources to determine whether they are equivalent and eligible to be used within the same message processing node

Example:

Compare the fully qualified DSN *MyDB* against a secondary fully qualified DSN *MyDB2* by using the primary and secondary user IDs and passwords:

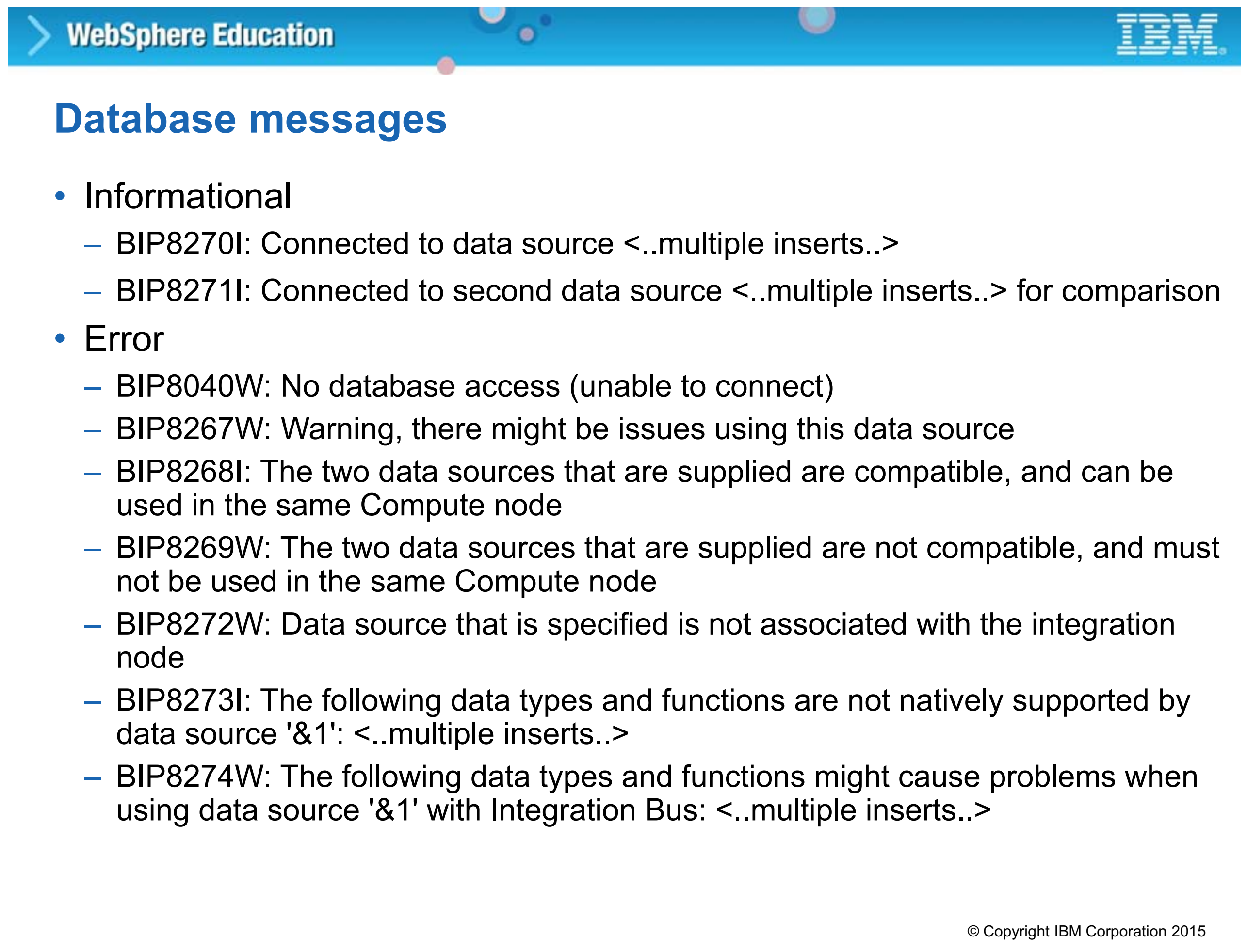
```
mqsicvp -n MyDB -u username -p password -c MyDB2
-i username2 -a password2
```

Figure 14-7. Verifying connectivity and compatibility WM6461.0

Notes:

The `mqsicvp` command provides database information and acts as an ODBC connection test tool.

When you use the `mqsicvp` command as an ODBC test tool, it generates an informational message on successful connection. The informational message provides the name of the data source, database type, and version. If a secondary data source is supplied, the `mqsicvp` command issues a second informational message about the secondary data source.



WebSphere Education

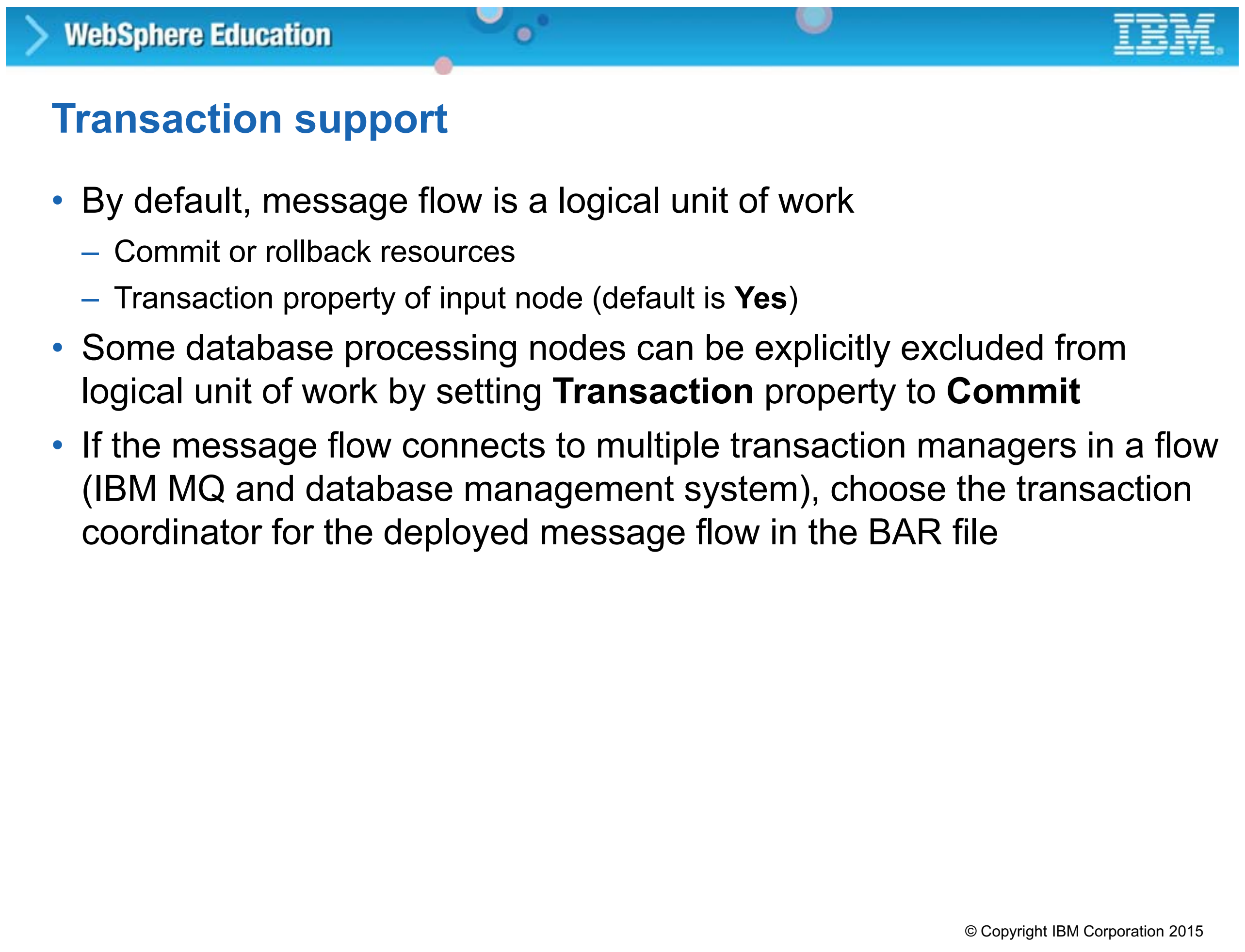
Database messages

- Informational
  - BIP8270I: Connected to data source <..multiple inserts..>
  - BIP8271I: Connected to second data source <..multiple inserts..> for comparison
- Error
  - BIP8040W: No database access (unable to connect)
  - BIP8267W: Warning, there might be issues using this data source
  - BIP8268I: The two data sources that are supplied are compatible, and can be used in the same Compute node
  - BIP8269W: The two data sources that are supplied are not compatible, and must not be used in the same Compute node
  - BIP8272W: Data source that is specified is not associated with the integration node
  - BIP8273I: The following data types and functions are not natively supported by data source '&1': <..multiple inserts..>
  - BIP8274W: The following data types and functions might cause problems when using data source '&1' with Integration Bus: <..multiple inserts..>

Figure 14-8. Database messages WM6461.0

Notes:

The figure lists the database-specific messages that might show in the logs.



WebSphere Education

Transaction support

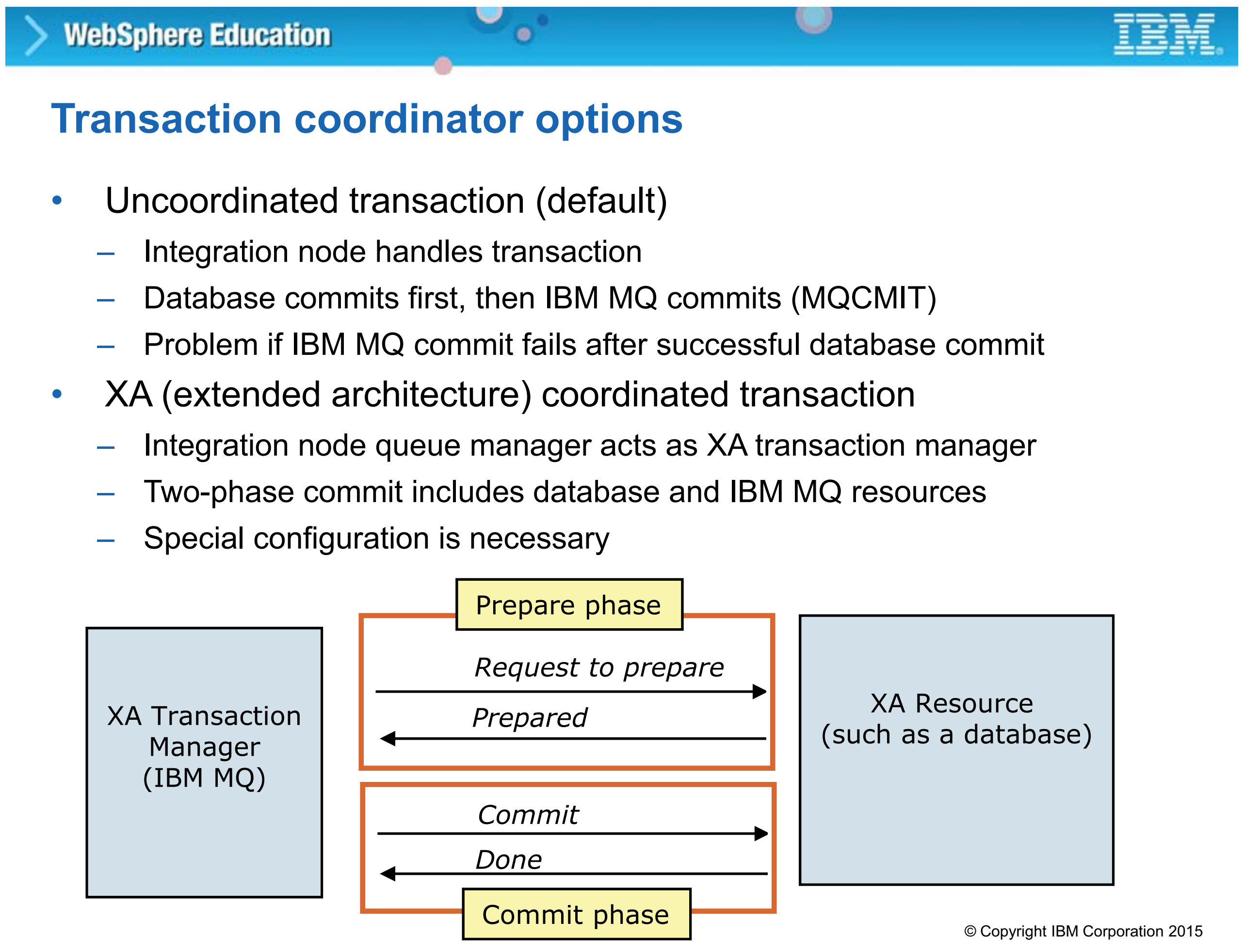
- By default, message flow is a logical unit of work
  - Commit or rollback resources
  - Transaction property of input node (default is **Yes**)
- Some database processing nodes can be explicitly excluded from logical unit of work by setting **Transaction** property to **Commit**
- If the message flow connects to multiple transaction managers in a flow (IBM MQ and database management system), choose the transaction coordinator for the deployed message flow in the BAR file

Figure 14-9. Transaction support WM6461.0

Notes:

A transaction describes a set of updates that are made by an application program, which must be managed together. The updates might be made to one or more systems. The environment in which the program runs controls the updates that the program makes. This property of a transaction is known as consistency: transactions might have other properties of atomicity, isolation, and durability.

Integration Bus supports transactions, and every piece of data that a message flow processes has an associated transaction. The integration node starts a message flow transaction when input data is received at an input node in the flow. The transaction is committed when the flow finishes with that message, or rolls back if an error occurs.



WebSphere Education

Transaction coordinator options

- Uncoordinated transaction (default)
  - Integration node handles transaction
  - Database commits first, then IBM MQ commits (MQCMIT)
  - Problem if IBM MQ commit fails after successful database commit
- XA (extended architecture) coordinated transaction
  - Integration node queue manager acts as XA transaction manager
  - Two-phase commit includes database and IBM MQ resources
  - Special configuration is necessary

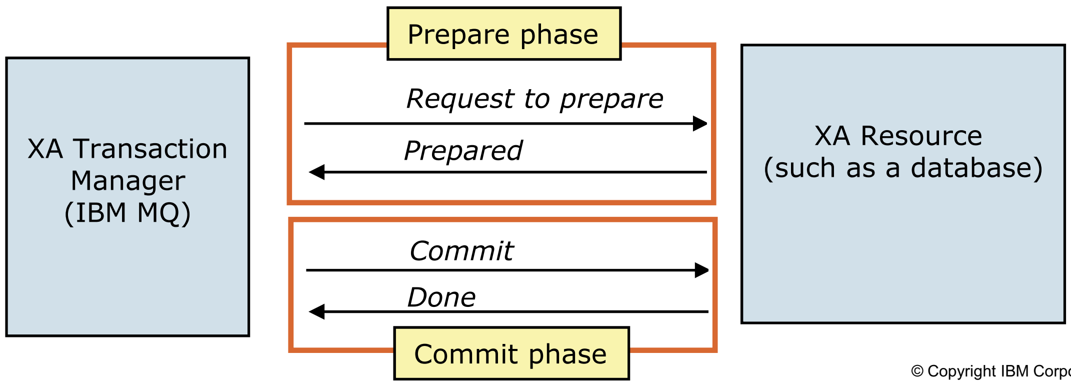

Prepare phase
XA Transaction Manager (IBM MQ)
Request to prepare
Prepared
XA Resource (such as a database)
Commit
Done
Commit phase

Figure 14-10. Transaction coordinator options WM6461.0

Notes:

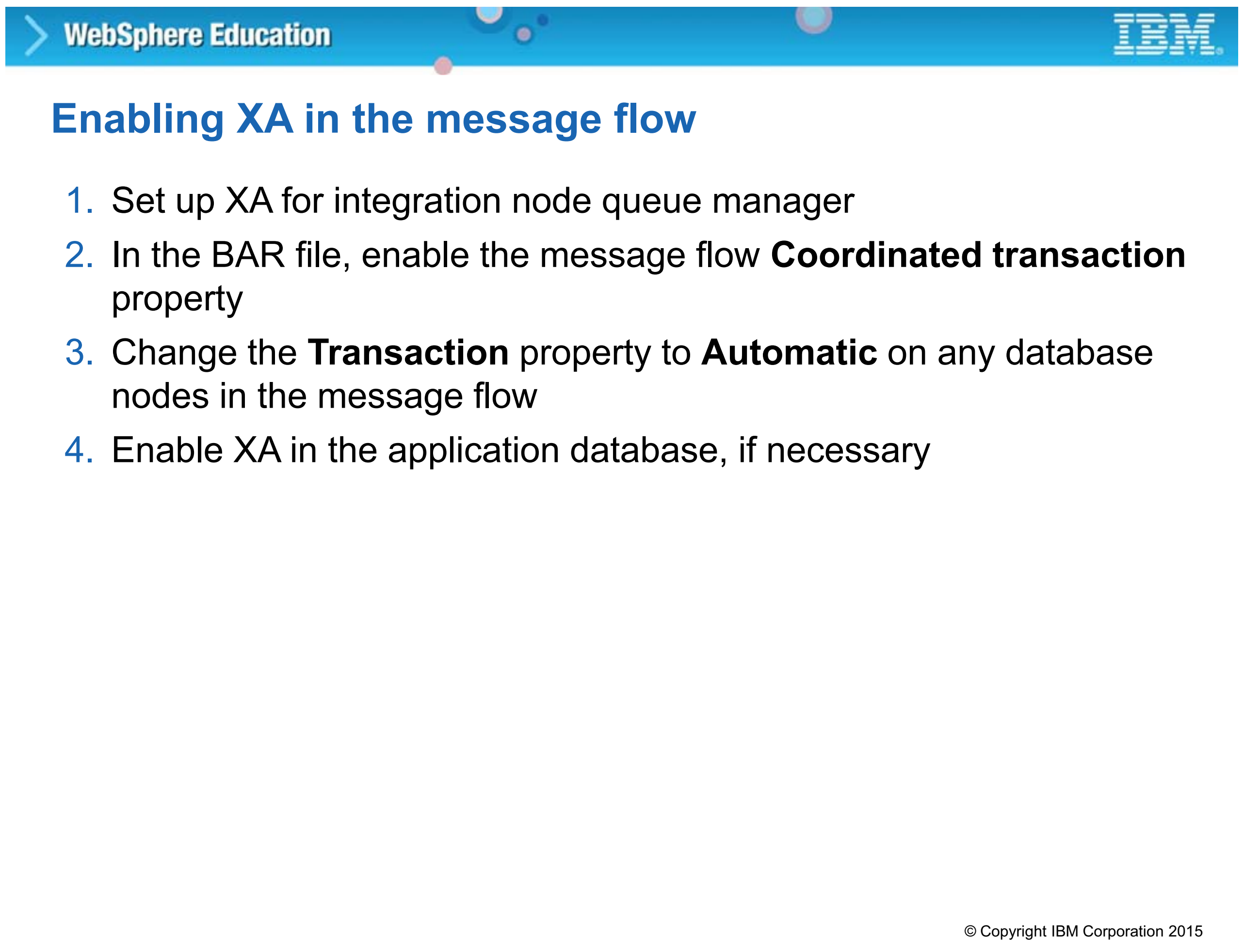
On distributed systems, the integration node manages the message flow transactions by default. These transactions are known as *local transactions* or *locally coordinated transactions*. Coordination is provided by an external transaction manager that uses XA protocols to interact with resource managers.

An XA-coordinated transaction is *not* visible to the message flow developer. It is enabled in the BAR file when it is deployed in special scenarios (for example, banking) where data integrity is critical. XA coordination affects processing and resources.

Uncoordinated flows are flows for which the **Coordinated** property is not set in the BAR file. Separate resource managers manage updates to resources used by an uncoordinated flow. Some resource managers, such as IBM MQ, allow updates to be made nontransactionally, or as part of a resource-specific transaction. Other resource managers, such as database managers, always use a resource-specific transaction. A resource-specific transaction is a transaction whose scope is limited to the resources owned by a single resource manager, such as a database or queue manager.

When multiple separate resource managers are synchronized, transaction coordination by using the XA protocol is provided on distributed environments by IBM MQ and on z/OS systems by RRS.

Message flows are always globally coordinated on z/OS, regardless of whether the **Coordinated** property of the message flow is specified as coordinated or not.



WebSphere Education

Enabling XA in the message flow

1. Set up XA for integration node queue manager
2. In the BAR file, enable the message flow **Coordinated transaction** property
3. Change the **Transaction** property to **Automatic** on any database nodes in the message flow
4. Enable XA in the application database, if necessary

Figure 14-11. Enabling XA in the message flow WM6461.0

Notes:

14.2.JMS provider

WebSphere Education

JMS architecture

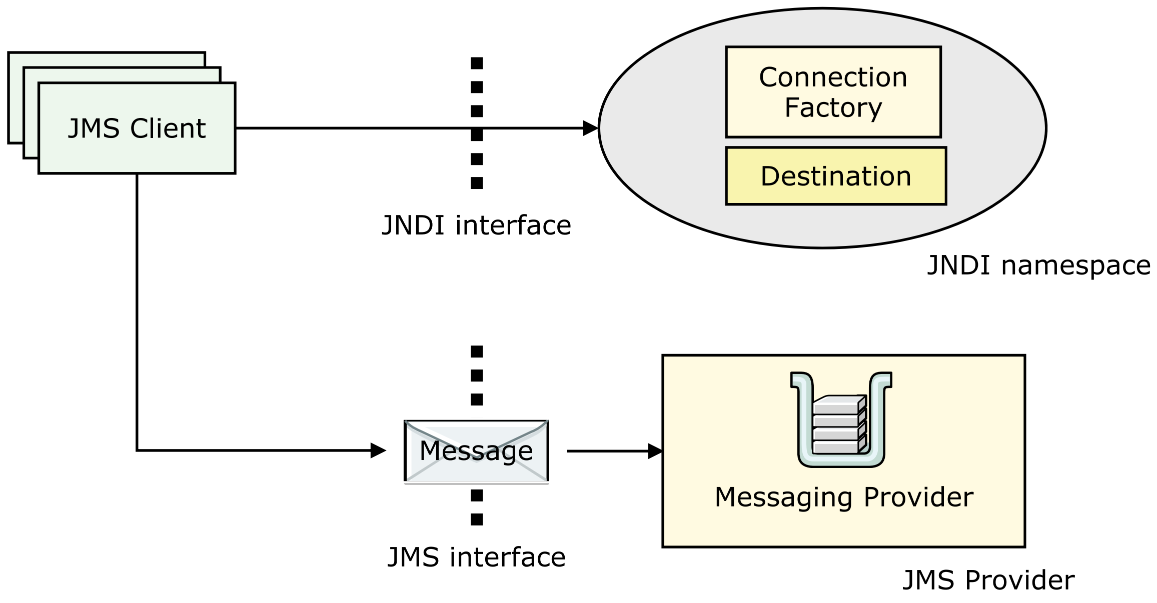


Figure 14-12. JMS architecture WM6461.0

Notes:

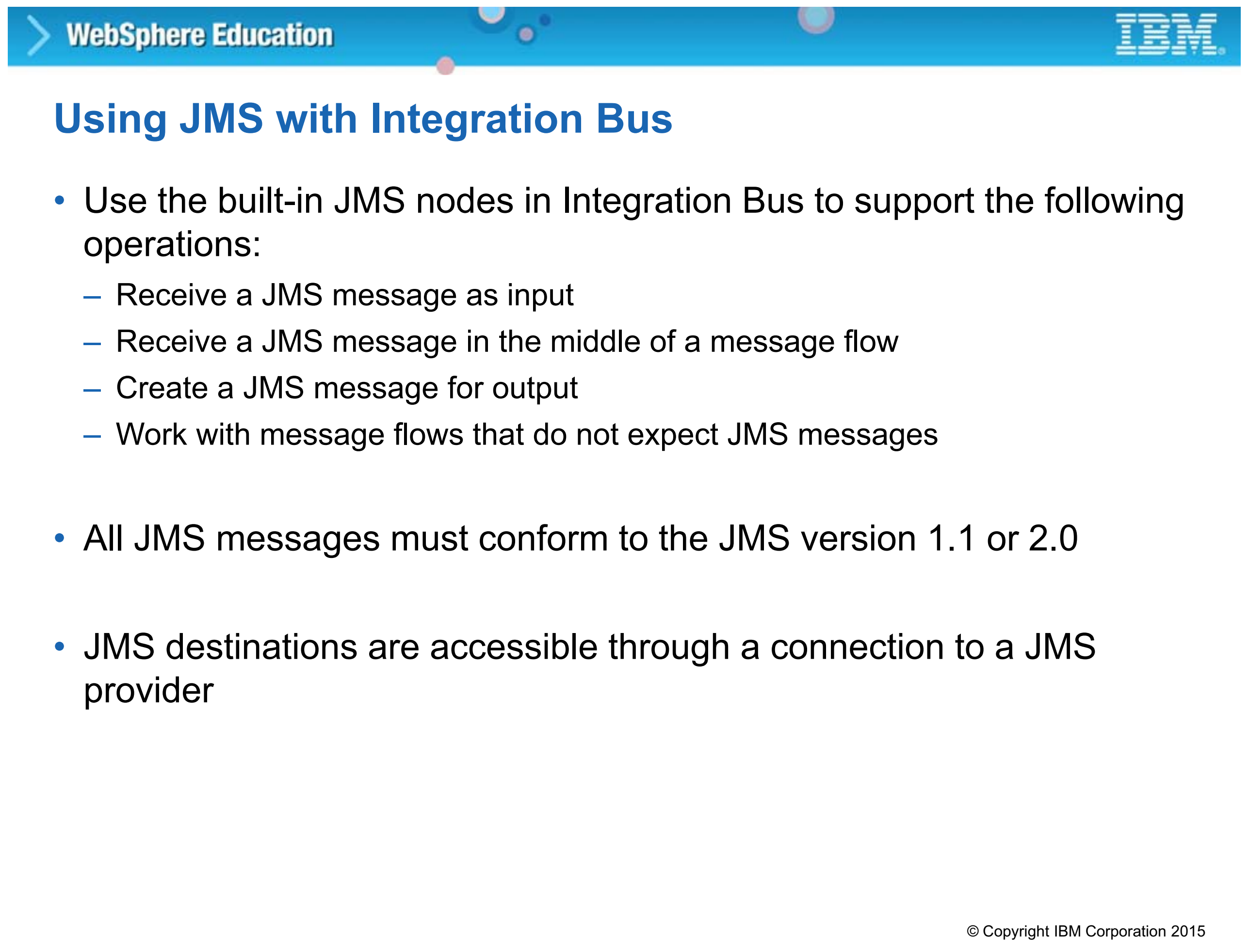
JMS messaging applications typically consist of the following parts:

- JMS messaging provider
- JMS clients: Java programs that produce and use messages by using the JMS. A JMS client specifies a Java Naming and Directory Interface (JNDI) initial context to obtain a JNDI connection to the JMS messaging server.

  JNDI is a standard Java extension that provides a uniform API for accessing various directory and naming services. JMS clients use JNDI to browse a naming service to get references to administered objects. Administered objects are the JMS connection factory and JMS destination topics and queues. The administrator creates and configures administered objects.

  The initial context is the starting point in any JNDI lookup and acts like the root of a file system. The JMS directory service that is used determines the properties for the initial context.
- Messages: The objects that communicate information between clients.
- Administered objects: Preconfigured JMS objects that an administrator creates for the clients to use. JNDI administered objects are stored in the bindings. The bindings can be either file

system or LDAP. A naming service associates names with distributed objects so that the administered objects are located by using names and not complex network addresses.



WebSphere Education

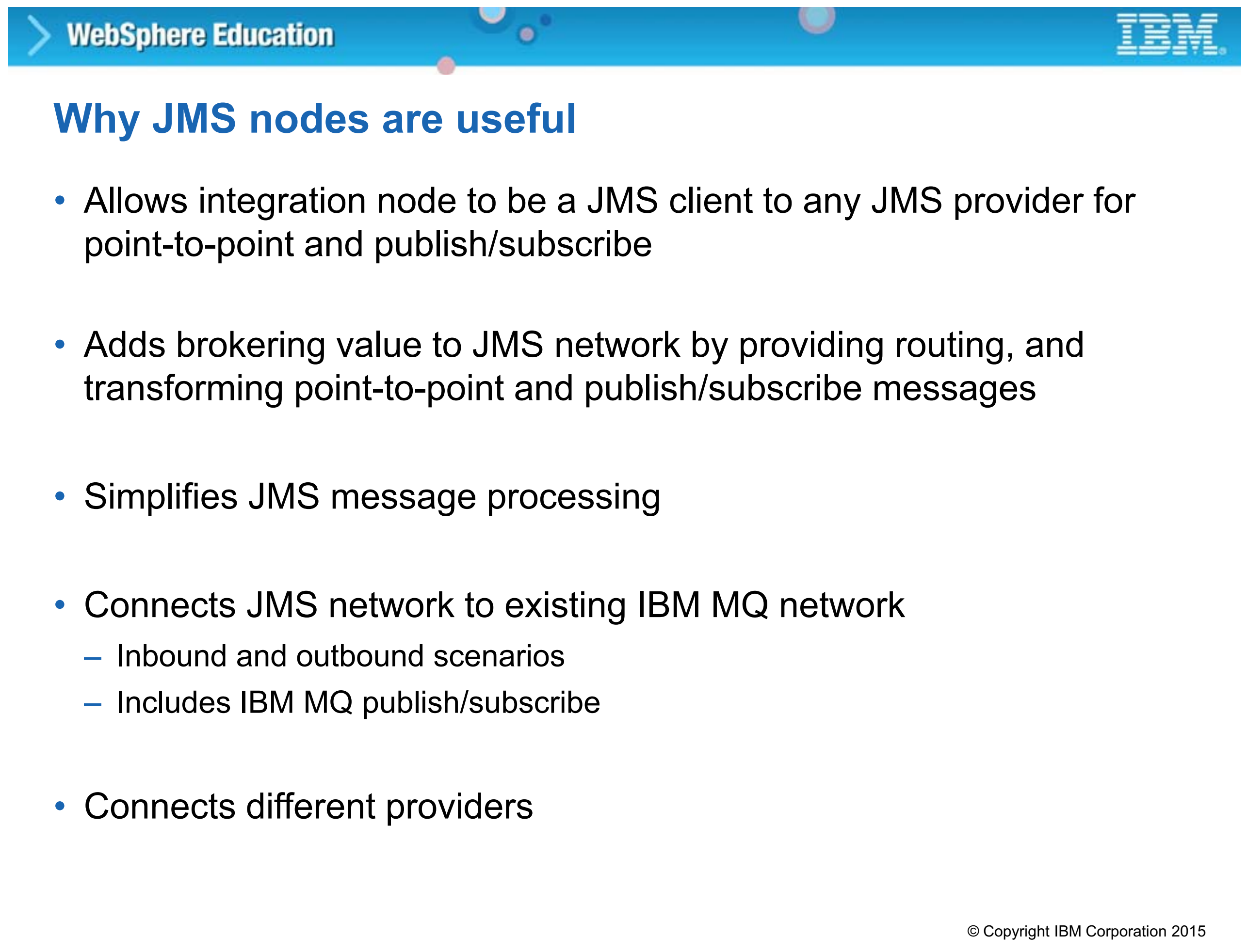
Using JMS with Integration Bus

- Use the built-in JMS nodes in Integration Bus to support the following operations:
  - Receive a JMS message as input
  - Receive a JMS message in the middle of a message flow
  - Create a JMS message for output
  - Work with message flows that do not expect JMS messages
- All JMS messages must conform to the JMS version 1.1 or 2.0
- JMS destinations are accessible through a connection to a JMS provider

Figure 14-13. Using JMS with Integration Bus WM6461.0

Notes:

Integration Bus contains built-in nodes for handling JMS messages. In sending and receiving messages, the JMS nodes behave like JMS clients.



WebSphere Education

Why JMS nodes are useful

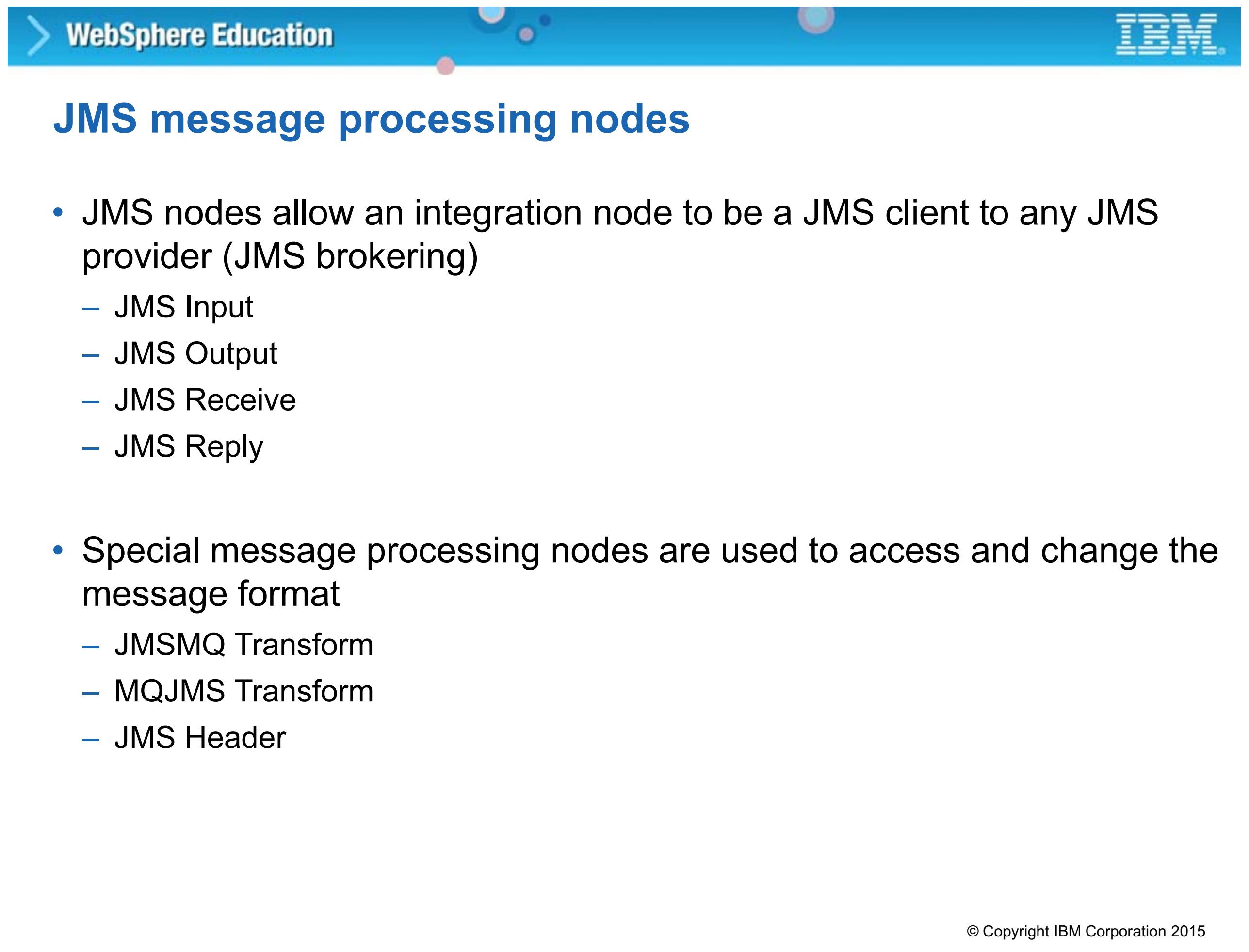
- Allows integration node to be a JMS client to any JMS provider for point-to-point and publish/subscribe
- Adds brokering value to JMS network by providing routing, and transforming point-to-point and publish/subscribe messages
- Simplifies JMS message processing
- Connects JMS network to existing IBM MQ network
  - Inbound and outbound scenarios
  - Includes IBM MQ publish/subscribe
- Connects different providers

Figure 14-14. Why JMS nodes are useful WM6461.0

Notes:

The JMS nodes can be used in applications where messages are produced and used from various JMS destinations. In sending and receiving messages, the JMS nodes behave like JMS clients.

The JMS nodes work with the IBM MQ JMS provider, WebSphere Application Server, and any JMS provider that conforms to JMS 1.1 or JMS 1.2.



WebSphere Education IBM

JMS message processing nodes

- JMS nodes allow an integration node to be a JMS client to any JMS provider (JMS brokering)
  - JMS Input
  - JMS Output
  - JMS Receive
  - JMS Reply
- Special message processing nodes are used to access and change the message format
  - JMSMQ Transform
  - MQJMS Transform
  - JMS Header

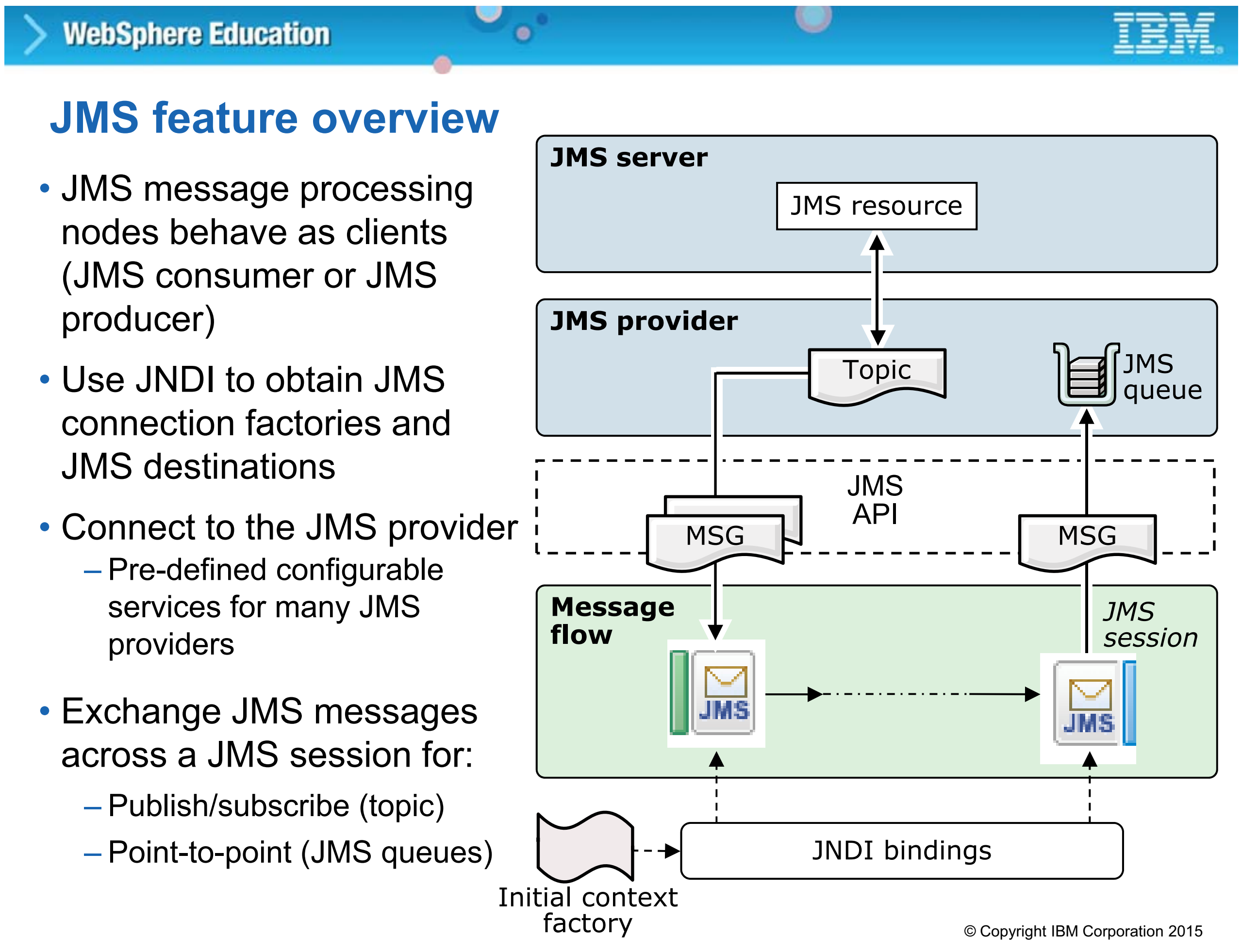
© Copyright IBM Corporation 2015

Figure 14-15. JMS message processing nodes WM6461.0

Notes:

Integration Bus has message processing nodes that offer brokering in a JMS network. These message processing nodes provide support to the integration node so that it can act as a JMS client.

JMS messages can be sent to (JMSOutput node) to and received from (JMSInput node) a JMS provider. In the message flow, a JMS message can be transformed to or from an IBM MQ message by using the JMSMQTransform node or the MQJMSTransform node.



JMS feature overview

- JMS message processing nodes behave as clients (JMS consumer or JMS producer)
- Use JNDI to obtain JMS connection factories and JMS destinations
- Connect to the JMS provider
  - Pre-defined configurable services for many JMS providers
- Exchange JMS messages across a JMS session for:
  - Publish/subscribe (topic)
  - Point-to-point (JMS queues)

Figure 14-16. JMS feature overview WM6461.0

Notes:

JMS nodes behave like JMS clients. They use JNDI to obtain JMS connection factories and JMS destinations.

JMS clients operate with both publish/subscribe and point-to-point messages. Both of these communication models use virtual channels that are called *destinations*. In the publish/subscribe model, the destinations are topics. For the point-to-point model, the destinations are known as queues.

The following application communication model properties can be configured for JMSInput and JMSOutput nodes:

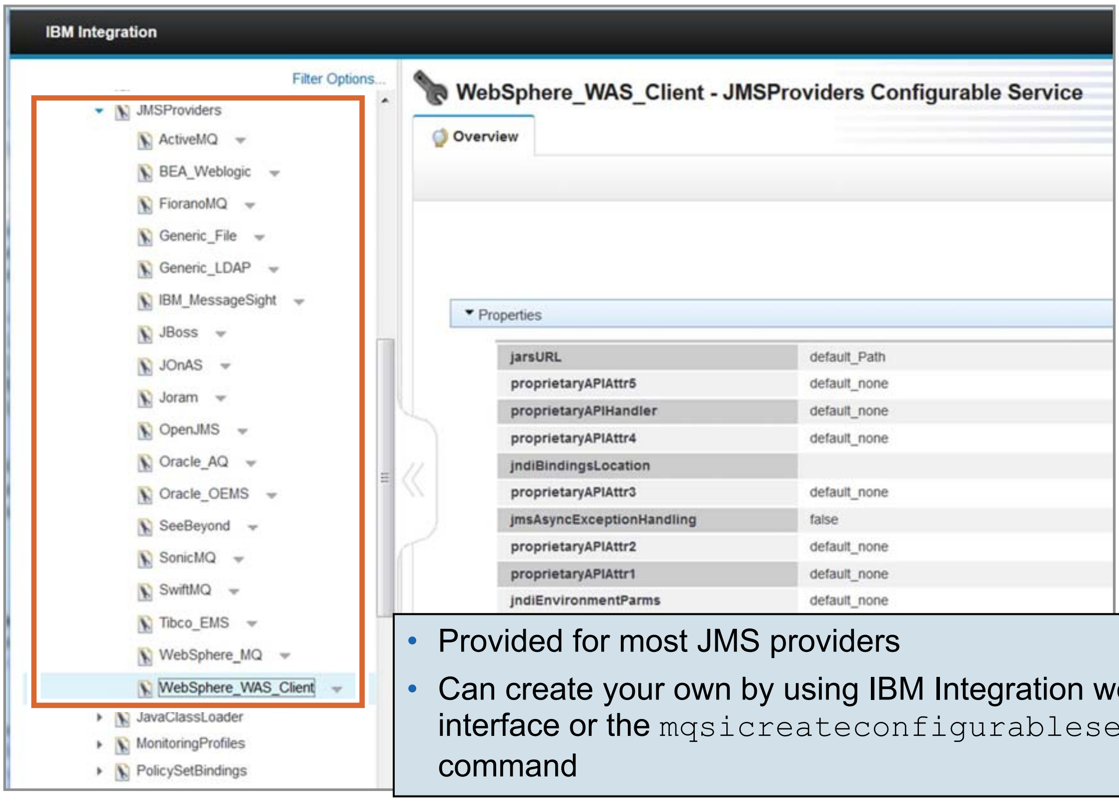
- **Connection factory name:** A string name that is passed to JNDI to look up the administered connection factory object. The connection factory object is used to create a connection to the JMS destination. For a client operating as a publish/subscribe client, the connection factory name is for a TopicConnectionFactory. For a client operating as a point-to-point client, the connection factory name is for a QueueConnectionFactory.

- **Subscription topic:** The string name that is passed to JNDI to look up the JMS topic destination. The topic is used to create a JMS session when the node is being used to process publish/subscribe messages.
- **Durable subscription ID:** A durable subscription is one that outlasts the client connection to a message server. When a durable subscriber is disconnected from the server, it is the responsibility of the server to store messages that are published. Therefore, when the durable subscriber reconnects, the message server sends all the unexpired messages. Durable subscriptions cannot be unsubscribed from a message flow. It requires a separate administration task to unsubscribe a previously registered durable subscription. Some JMS providers supply an administration tool to do this task.
- **Source queue:** The string name that is passed to JNDI to look up the JMS queue destination. The queue is used to create a JMS session when the node is being used to process point-to-point messages.

You can use configurable services to configure Integration Bus for JMS.

WebSphere Education IBM

JMSProviders configurable services



IBM Integration

Filter Options...

- JMSProviders
  - ActiveMQ
  - BEA_Weblogic
  - FioranoMQ
  - Generic_File
  - Generic_LDAP
  - IBM_MessageSight
  - JBoss
  - JOnAS
  - Joram
  - OpenJMS
  - Oracle_AQ
  - Oracle_OEMS
  - SeeBeyond
  - SonicMQ
  - SwiftMQ
  - Tibco_EMS
  - WebSphere_MQ
  - WebSphere_WAS_Client
- JavaClassLoader
- MonitoringProfiles
- PolicySetBindings

WebSphere_WAS_Client - JMSProviders Configurable Service

Overview

Properties

jarsURL	default_Path
proprietaryAPIAttr5	default_none
proprietaryAPIHandler	default_none
proprietaryAPIAttr4	default_none
jndiBindingsLocation	
proprietaryAPIAttr3	default_none
jmsAsyncExceptionHandling	false
proprietaryAPIAttr2	default_none
proprietaryAPIAttr1	default_none
jndiEnvironmentParms	default_none

- Provided for most JMS providers
- Can create your own by using IBM Integration web user interface or the `mqsicreateconfigurableservice` command

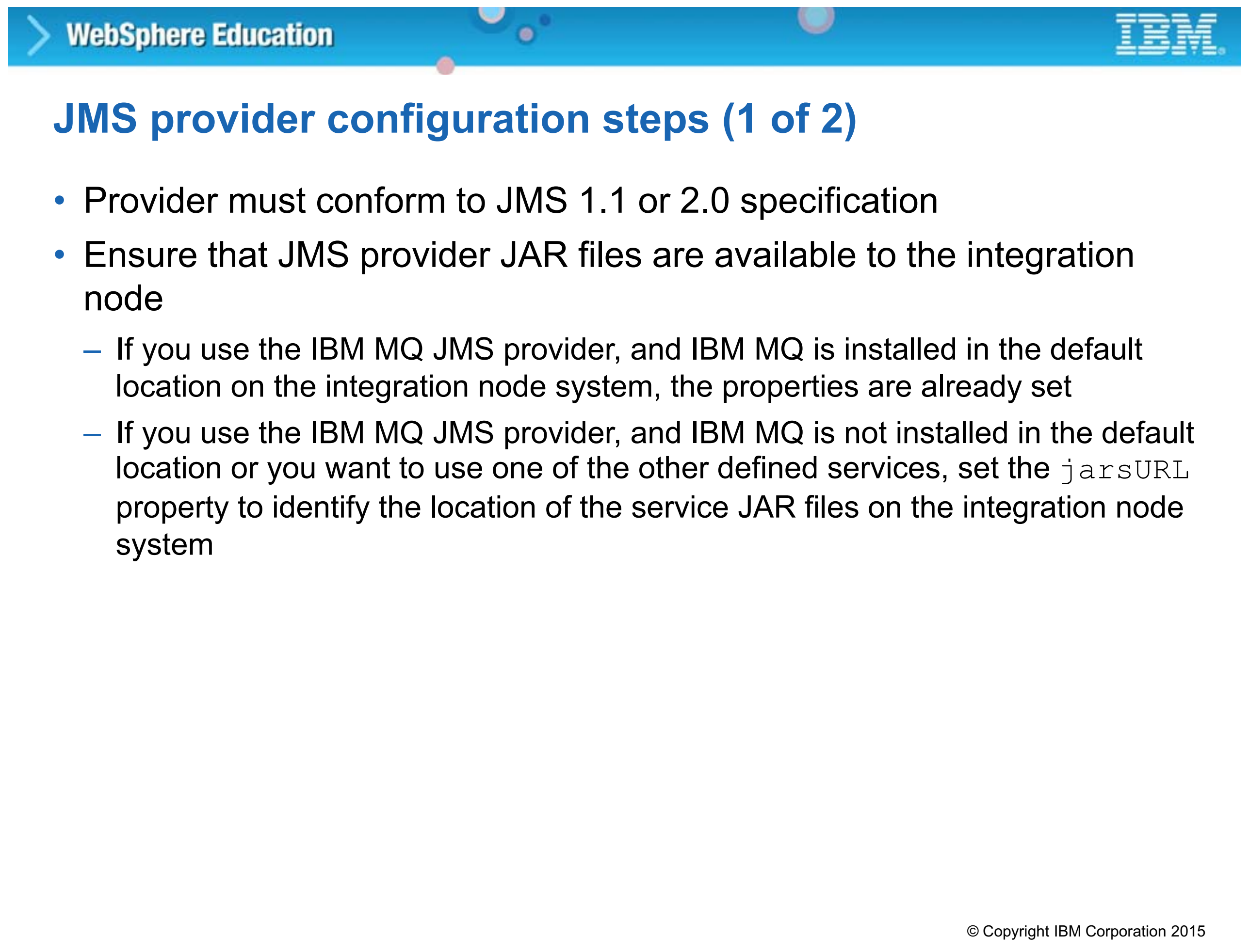
© Copyright IBM Corporation 2015

Figure 14-17. JMSProviders configurable services WM6461.0

Notes:

Configurable services are defined for many JMS providers. You can choose one of the predefined configurable services, or you can create a service for a new provider, or for one of the existing providers.

To create a configurable service for a JMS provider, use IBM Integration web user interface or the `mqsicreateconfigurableservice` command.



WebSphere Education

JMS provider configuration steps (1 of 2)

- Provider must conform to JMS 1.1 or 2.0 specification
- Ensure that JMS provider JAR files are available to the integration node
  - If you use the IBM MQ JMS provider, and IBM MQ is installed in the default location on the integration node system, the properties are already set
  - If you use the IBM MQ JMS provider, and IBM MQ is not installed in the default location or you want to use one of the other defined services, set the `jarsURL` property to identify the location of the service JAR files on the integration node system

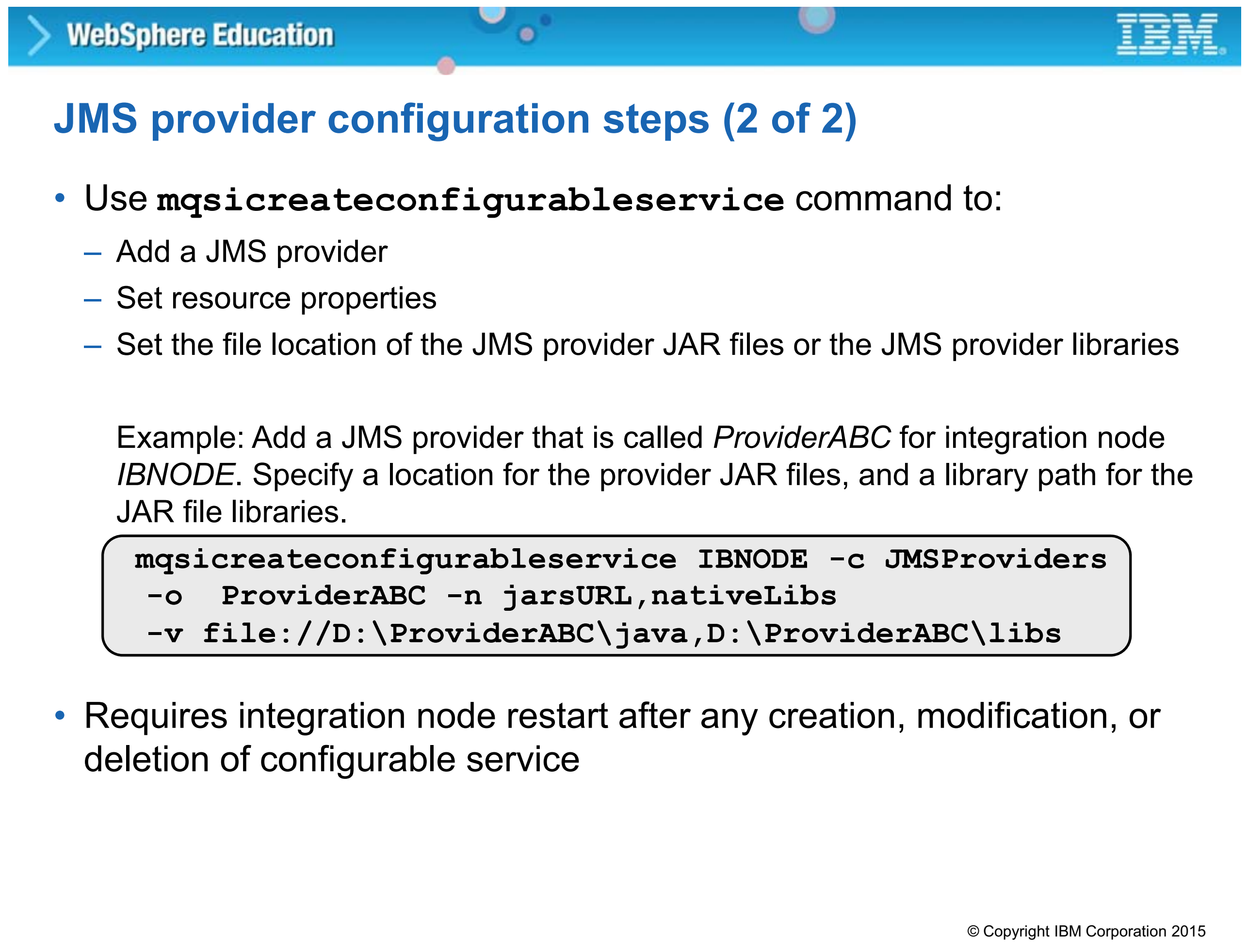
© Copyright IBM Corporation 2015

Figure 14-18. JMS provider configuration steps (1 of 2) WM6461.0

Notes:

The figure shows how to configure the environment for a JMS provider. This description is based on using IBM MQ as the JMS provider.

You must copy the Java `.jar` files and any libraries for the JMS provider client into the integration node shared-classes directory. For example, on Windows, this directory is `C:\Documents and Settings\All Users\Application Data\IBM\MQSI\shared-classes`. This action ensures that the Java class path for the JMS nodes is set correctly.



WebSphere Education

JMS provider configuration steps (2 of 2)

- Use **`mqsicreateconfigurableservice`** command to:
  - Add a JMS provider
  - Set resource properties
  - Set the file location of the JMS provider JAR files or the JMS provider libraries

  Example: Add a JMS provider that is called *ProviderABC* for integration node *IBNODE*. Specify a location for the provider JAR files, and a library path for the JAR file libraries.

```
mqsicreateconfigurableservice IBNODE -c JMSProviders
 -o  ProviderABC -n jarsURL,nativeLibs
 -v file://D:\ProviderABC\java,D:\ProviderABC\libs
```

- Requires integration node restart after any creation, modification, or deletion of configurable service

Figure 14-19. JMS provider configuration steps (2 of 2) WM6461.0

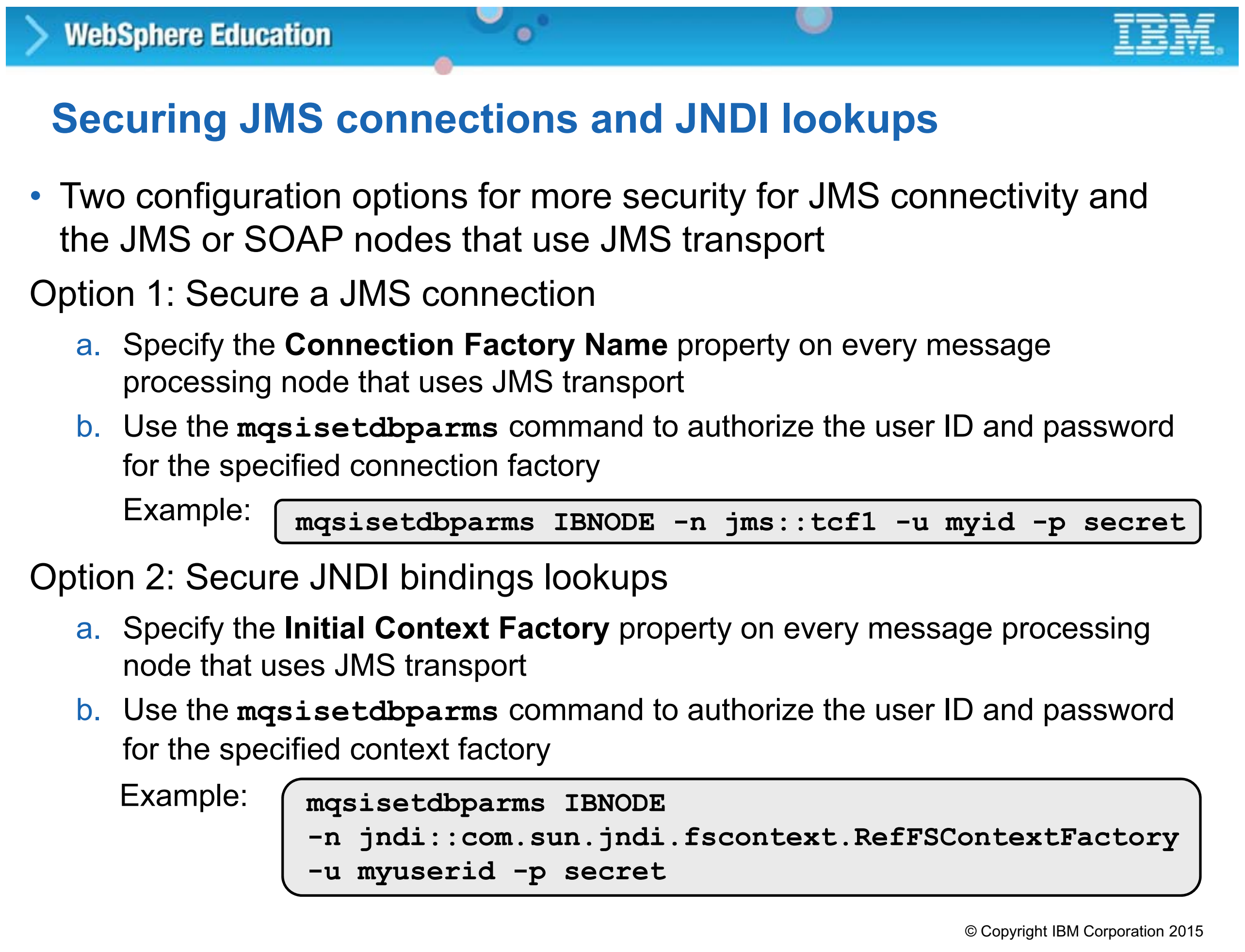
Notes:

You can use the `mqsicreateconfigurableservice` command to add a JMS provider, configure its properties, and set the location for the associated library files.

Related commands can be used to report, modify, or delete configurable services:

- Use `mqsichangeproperties` command to modify configurable services.
- Use `mqsideleteconfigurableservice` command to delete configurable services.
- Use `mqsireportproperties` command to view configurable services.

These commands can also be used to configure JDBC properties and FTP services.



WebSphere Education

Securing JMS connections and JNDI lookups

- Two configuration options for more security for JMS connectivity and the JMS or SOAP nodes that use JMS transport

Option 1: Secure a JMS connection

a. Specify the **Connection Factory Name** property on every message processing node that uses JMS transport

b. Use the **`mqsisetdbparms`** command to authorize the user ID and password for the specified connection factory

Example:

```
mqsisetdbparms IBNODE -n jms::tcf1 -u myid -p secret
```

Option 2: Secure JNDI bindings lookups

a. Specify the **Initial Context Factory** property on every message processing node that uses JMS transport

b. Use the **`mqsisetdbparms`** command to authorize the user ID and password for the specified context factory

Example:

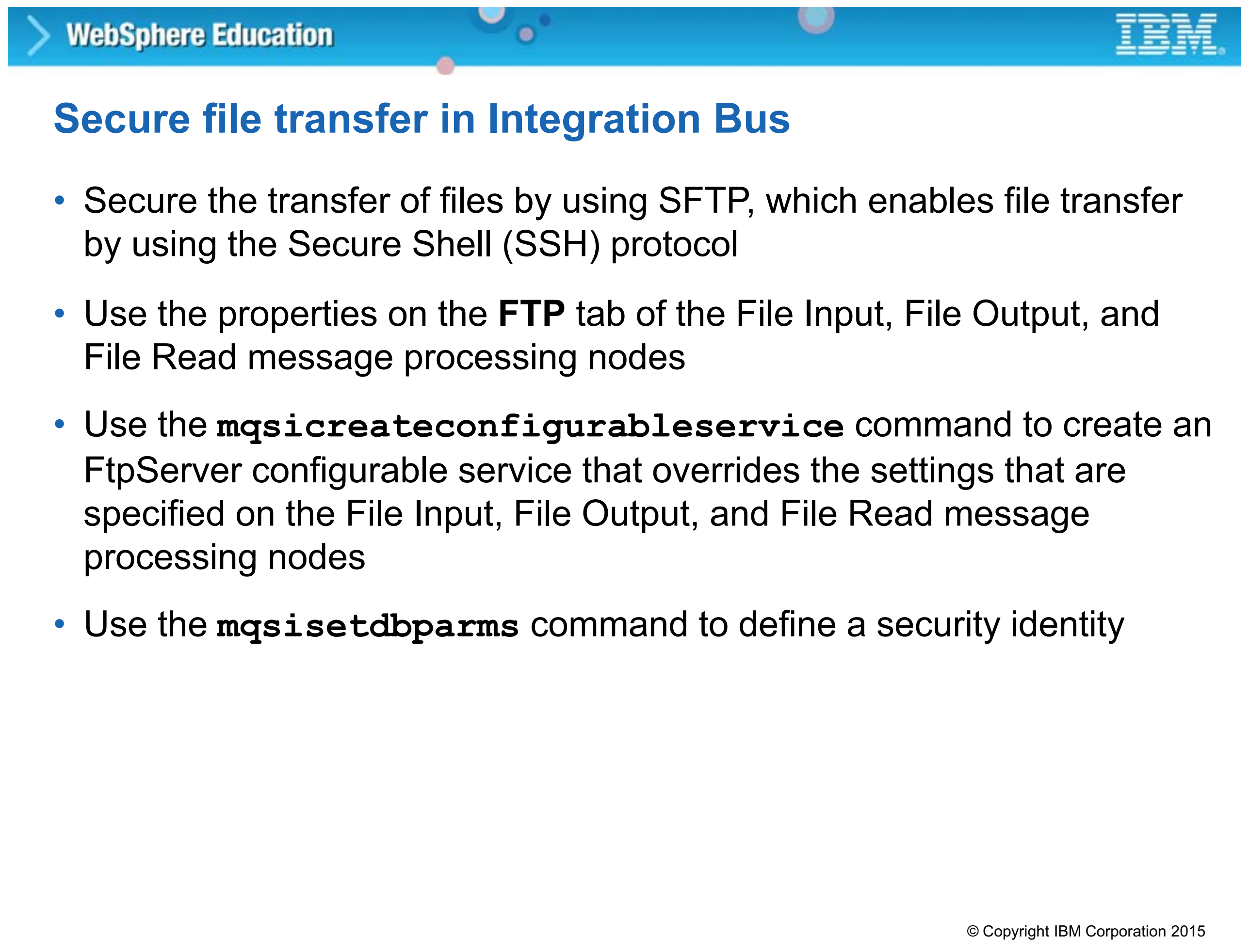
```
mqsisetdbparms IBNODE
-n jndi::com.sun.jndi.fscontext.RefFSContextFactory
-u myuserid -p secret
```

Figure 14-20. Securing JMS connections and JNDI lookups WM6461.0

Notes:

There are two configuration options for adding security for JMS connectivity. You can secure a JMS connection and you can secure JNDI bindings lookups. The figure shows the steps for securing JMS connections and JNDI lookups.

14.3.Configuring for SFTP



WebSphere Education IBM

Secure file transfer in Integration Bus

- Secure the transfer of files by using SFTP, which enables file transfer by using the Secure Shell (SSH) protocol
- Use the properties on the **FTP** tab of the File Input, File Output, and File Read message processing nodes
- Use the **`mqsicreateconfigurableservice`** command to create an FtpServer configurable service that overrides the settings that are specified on the File Input, File Output, and File Read message processing nodes
- Use the **`mqsisetdbparms`** command to define a security identity

© Copyright IBM Corporation 2015

Figure 14-21. Secure file transfer in Integration Bus WM6461.0

Notes:

A message flow that contains Integration Bus File nodes can transfer files securely by using SFTP.

The SFTP properties are specified on the File node in the message flow. An administrator can override these properties by creating an FtpServer configurable service.

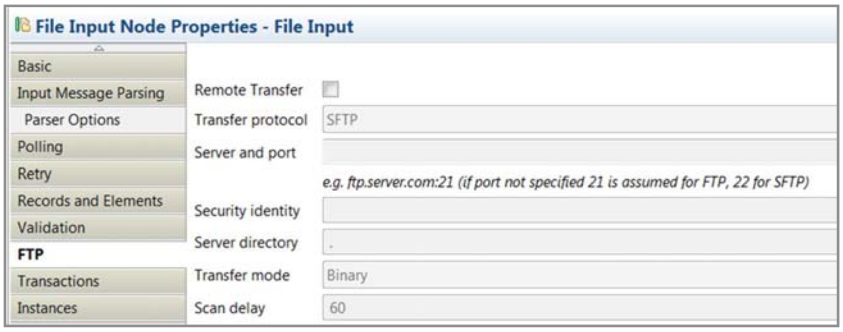
The settings that you specify by using an FtpServer configurable service are read and validated when the message flow starts. The configurable service defines the SFTP connections that are made for the integration node.

The configurable service can override any or all of the remote transfer properties on the **FTP** tab of the FileInput, FileOutput, and FileRead nodes.

WebSphere Education IBM

Steps for implementing SFTP

1. Use the **`mqsicreateconfigurableservice`** command to create an FtpServer configurable service
2. In the message flow File Input, File Output, and File Read nodes **FTP** properties:
   - Select **Remote Transfer**
   - Set the **Transfer protocol** to **SFTP**
   - Specify the name of the FtpServer configurable service in **Server and port**


File Input Node Properties - File Input

Remote Transfer ☐
Transfer protocol SFTP
Server and port
e.g. ftp.server.com:21 (if port not specified 21 is assumed for FTP, 22 for SFTP)
Security identity
Server directory .
Transfer mode Binary
Scan delay 60

3. Stop and start the integration server

© Copyright IBM Corporation 2015

Figure 14-22. Steps for implementing SFTP WM6461.0

Notes:

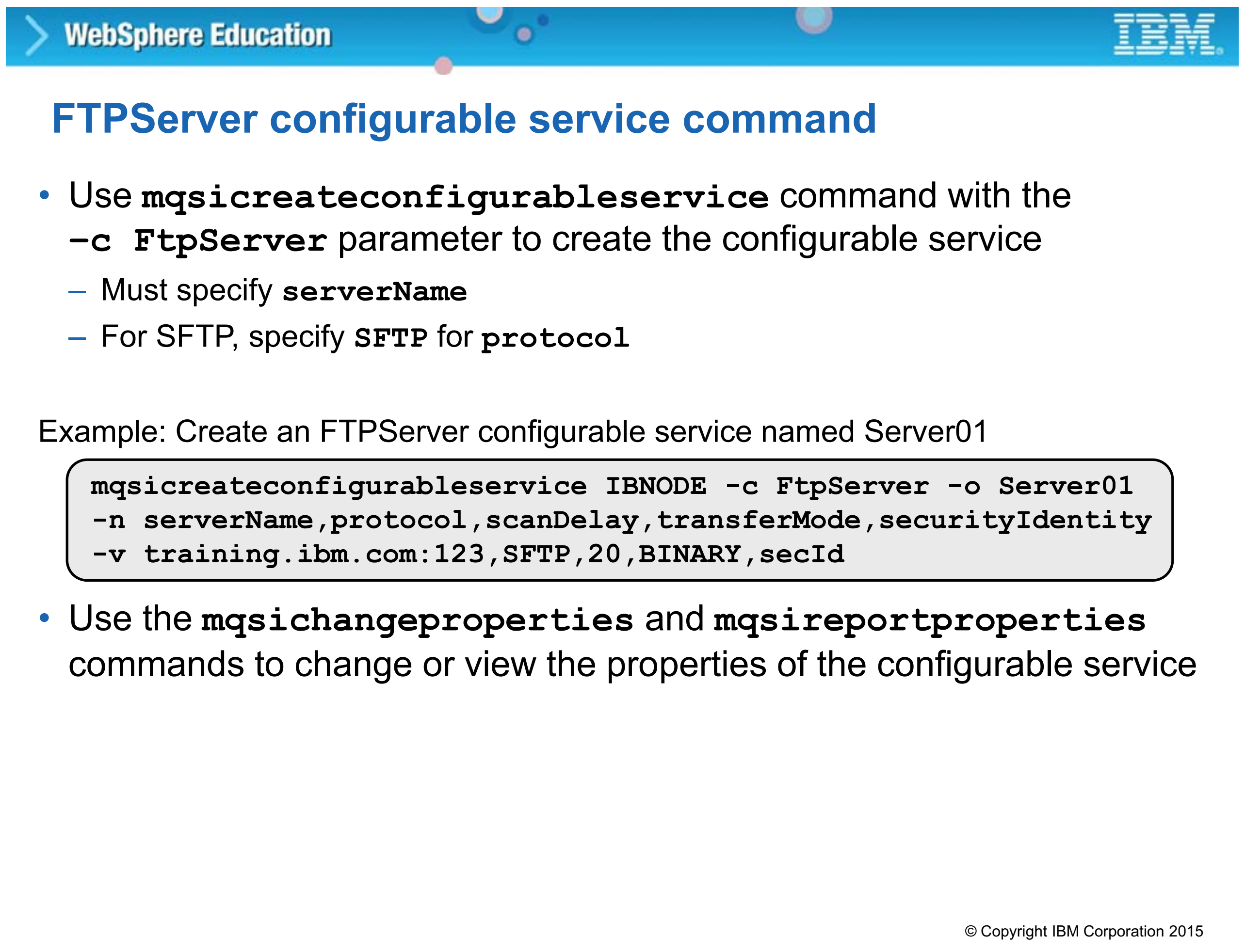
The figure lists the steps for implementing SFTP for File nodes in a message flow. The figure shows the FTP properties available for a FileInput node.

If a configurable service name is specified in the **Server and port** field, its properties override the remote transfer properties on the **FTP** tab.

The following FTP properties can be overridden in the BAR file or with the `mqsiapplybaroverride` command:

- **Remote Transfer** (fileFtp)
- **Transfer protocol** (remoteTransferType)
- **Server and port** (fileFtpServer)
- **Security identify** (fileFtpUser)
- **Server directory** (fileFtpDirectory)

You stop and start the integration node to ensure that the configurable service is available to all resources that are running on the integration node.



WebSphere Education IBM.

FTPServer configurable service command

- Use **`mqsicreateconfigurableservice`** command with the **`-c FtpServer`** parameter to create the configurable service
  - Must specify **`serverName`**
  - For SFTP, specify **`SFTP`** for **`protocol`**

Example: Create an FTPServer configurable service named Server01

```
mqsicreateconfigurableservice IBNODE -c FtpServer -o Server01
-n serverName,protocol,scanDelay,transferMode,securityIdentity
-v training.ibm.com:123,SFTP,20,BINARY,secId
```

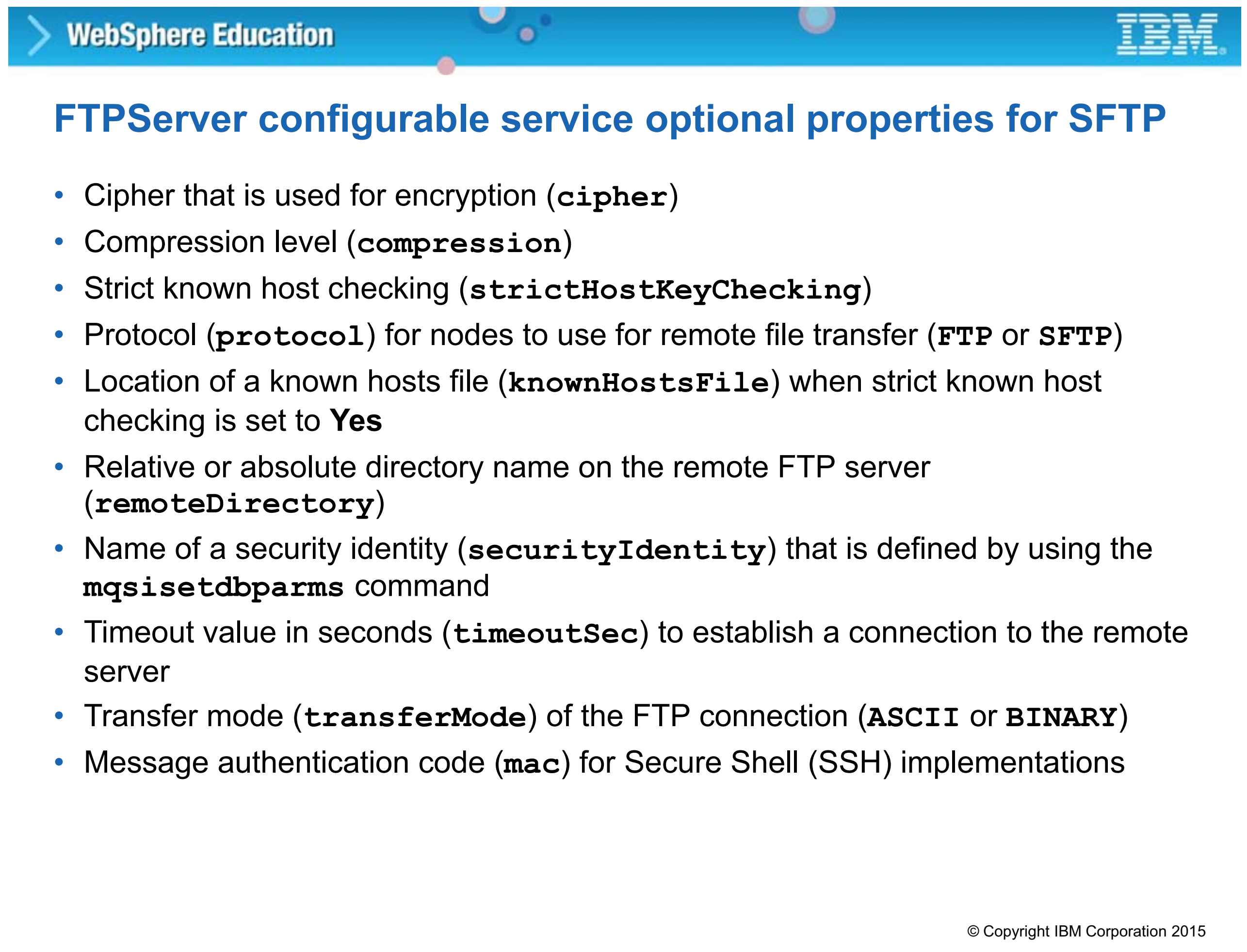
- Use the **`mqsichangeproperties`** and **`mqsireportproperties`** commands to change or view the properties of the configurable service

Figure 14-23. FTPServer configurable service command WM6461.0

Notes:

You can use the `mqsicreateconfigurableservice` command to create the FTPserver configurable service.

You can either enter each name-value pair in a separate command or in a single command with each name and value that is separated by a comma. The example in the figure shows an example of a single command.



WebSphere Education
IBM

FTPServer configurable service optional properties for SFTP

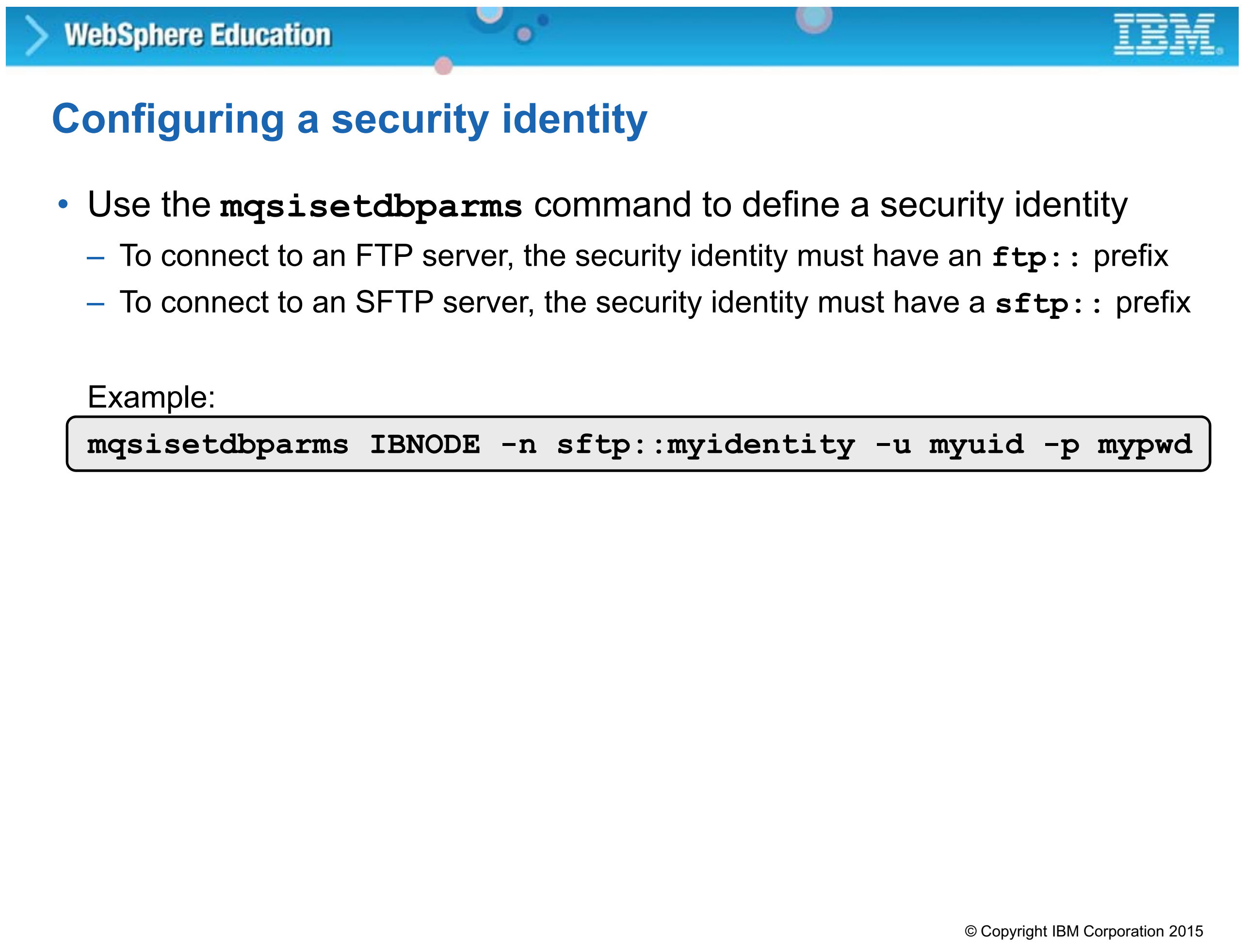
- Cipher that is used for encryption (**`cipher`**)
- Compression level (**`compression`**)
- Strict known host checking (**`strictHostKeyChecking`**)
- Protocol (**`protocol`**) for nodes to use for remote file transfer (**`FTP`** or **`SFTP`**)
- Location of a known hosts file (**`knownHostsFile`**) when strict known host checking is set to **Yes**
- Relative or absolute directory name on the remote FTP server (**`remoteDirectory`**)
- Name of a security identity (**`securityIdentity`**) that is defined by using the **`mqsisetdbparms`** command
- Timeout value in seconds (**`timeoutSec`**) to establish a connection to the remote server
- Transfer mode (**`transferMode`**) of the FTP connection (**`ASCII`** or **`BINARY`**)
- Message authentication code (**`mac`**) for Secure Shell (SSH) implementations

Figure 14-24. FTPServer configurable service optional properties for SFTP WM6461.0

Notes:

The figure lists some of the SFTP properties in the FTPserver configurable service.

The `cipher` and `compression` properties are valid only when SFTP is specified as the protocol. If FTP is used, this property is ignored.



Configuring a security identity

- Use the **`mqsisetdbparms`** command to define a security identity
  - To connect to an FTP server, the security identity must have an **`ftp::`** prefix
  - To connect to an SFTP server, the security identity must have a **`sftp::`** prefix

Example:

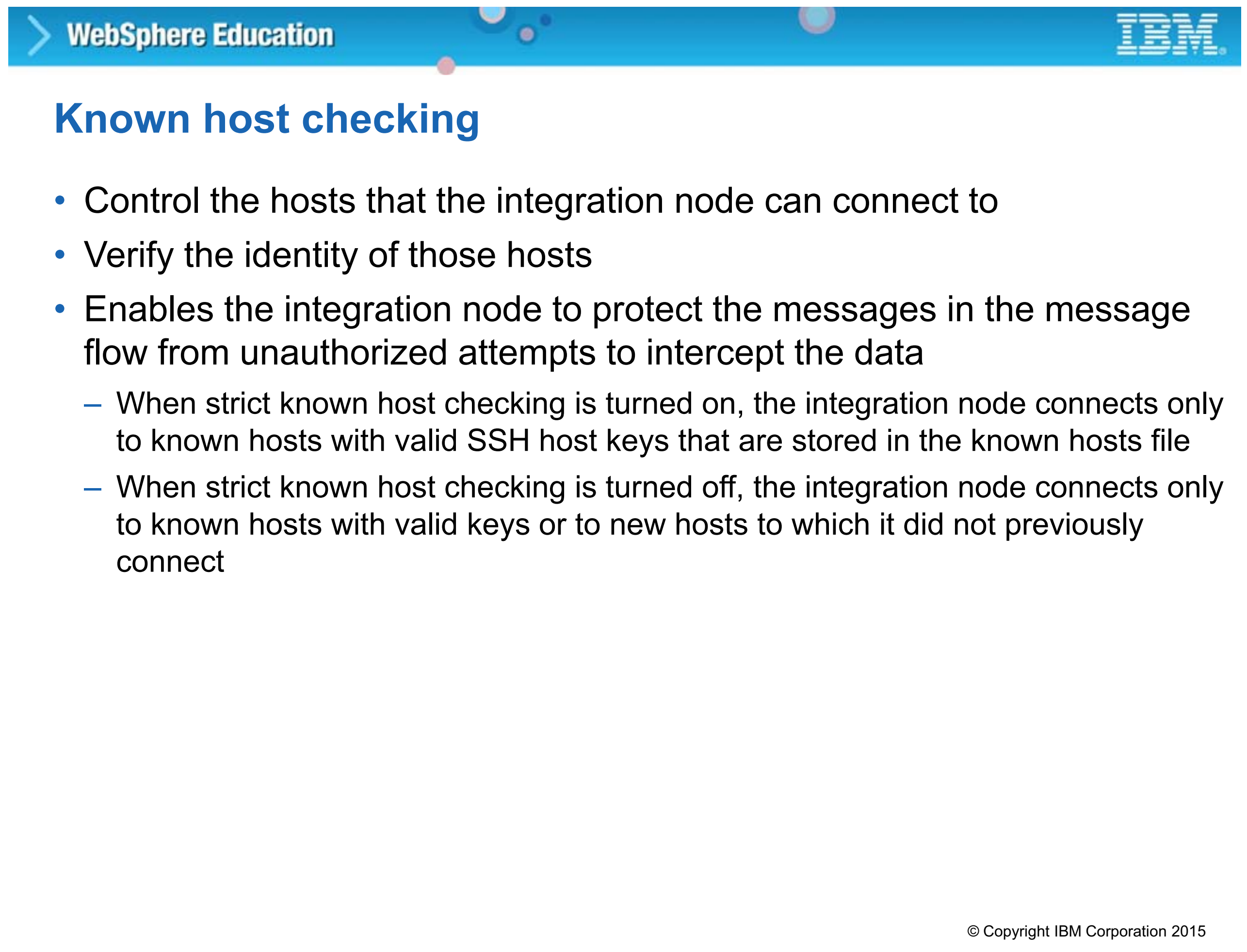
```
mqsisetdbparms IBNODE -n sftp::myidentity -u myuid -p mypwd
```

Figure 14-25. Configuring a security identity WM6461.0

Notes:

Secure file transfer requires an integration node security identity when connecting to the FTP or SFTP server. You use the `mqsisetdbparms` command to define the integration node security identity.

The example command defines a security identity to connect to an SFTP server.



WebSphere Education

Known host checking

- Control the hosts that the integration node can connect to
- Verify the identity of those hosts
- Enables the integration node to protect the messages in the message flow from unauthorized attempts to intercept the data
  - When strict known host checking is turned on, the integration node connects only to known hosts with valid SSH host keys that are stored in the known hosts file
  - When strict known host checking is turned off, the integration node connects only to known hosts with valid keys or to new hosts to which it did not previously connect

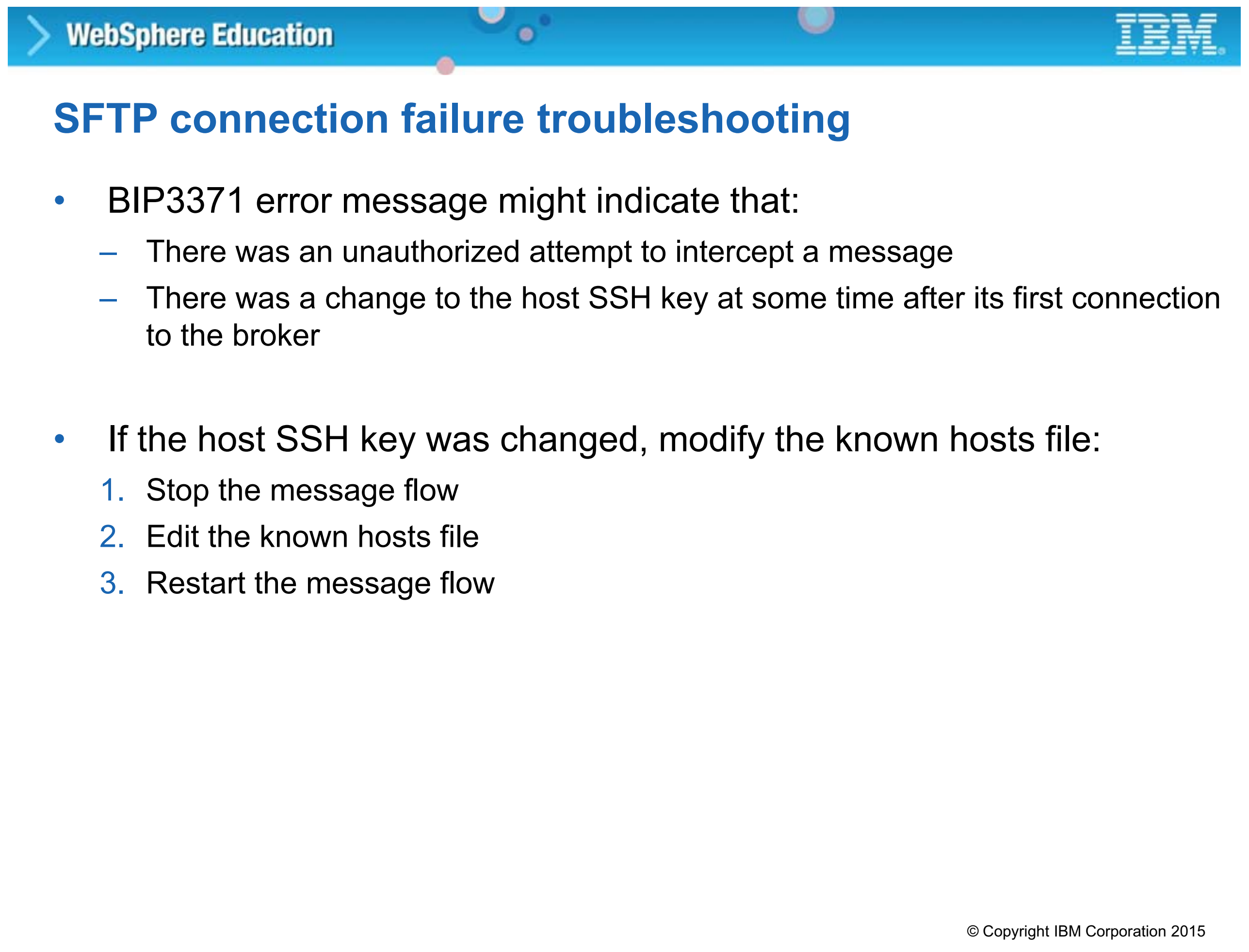
Figure 14-26. Known host checking WM6461.0

Notes:

Before it connects to a host to receive or transfer a file, the integration node checks the host key against the keys that are stored in the known hosts file. If the host key does not match an existing entry for the host in the known hosts file, the connection fails. If the host is new (it has no entry in the known hosts file), the result depends on whether strict known host checking is turned on or off.

When strict host key checking is enabled, the integration node connects only to known hosts with valid Secure Shell (SSH) host keys in the known hosts file. To enable strict host checking, you set the `strictHostKeyChecking` parameter in the FTPserver configurable service to Yes.

When strict known host checking is turned off, the integration node connects only to known hosts that have valid keys or to new hosts.



WebSphere Education

SFTP connection failure troubleshooting

- BIP3371 error message might indicate that:
  - There was an unauthorized attempt to intercept a message
  - There was a change to the host SSH key at some time after its first connection to the broker

- If the host SSH key was changed, modify the known hosts file:
  1. Stop the message flow
  2. Edit the known hosts file
  3. Restart the message flow

Figure 14-27. SFTP connection failure troubleshooting WM6461.0

Notes:

You might see a BIP3371 error message when implementing and SFTP connection.

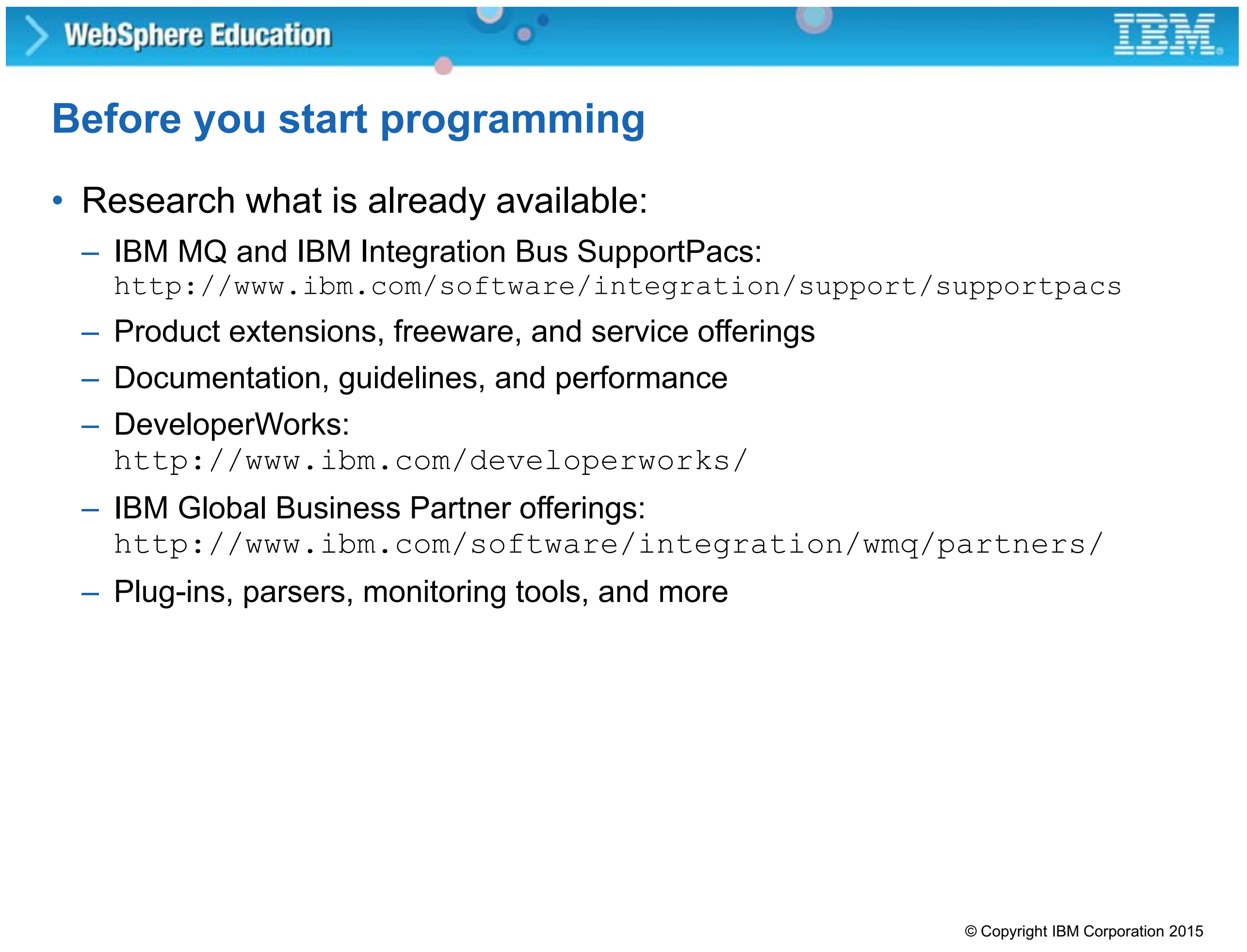
If the SSH key changed and you enabled strict known host checking, correct the entry for the host in the known hosts file. The known hosts file is specified in the `knownHostsFile` parameter of the FTPserver configurable service.

If you have strict known host checking turned off, remove the host entry from the integration node managed known hosts file. The known hosts file is in the `\components\INODENAME\IS-UID\config\known_hosts` subdirectory of the directory in which Integration Bus is installed.

- On Windows, the default directory is `C:\ProgramData\IBM\MQSI\components\INODENAME\IS-UID\config\known_hosts`
- On UNIX systems, the default directory is `/var/mqsi/components/INODENAME/IS-UID/config/known_hosts`

When the integration node attempts to reconnect, it adds the new host key to the known hosts file.

14.4.IBM Integration Bus SupportPacs



WebSphere Education

Before you start programming

- Research what is already available:
  - IBM MQ and IBM Integration Bus SupportPacs: `http://www.ibm.com/software/integration/support/supportpacs`
  - Product extensions, freeware, and service offerings
  - Documentation, guidelines, and performance
  - DeveloperWorks: `http://www.ibm.com/developerworks/`
  - IBM Global Business Partner offerings: `http://www.ibm.com/software/integration/wmq/partners/`
  - Plug-ins, parsers, monitoring tools, and more

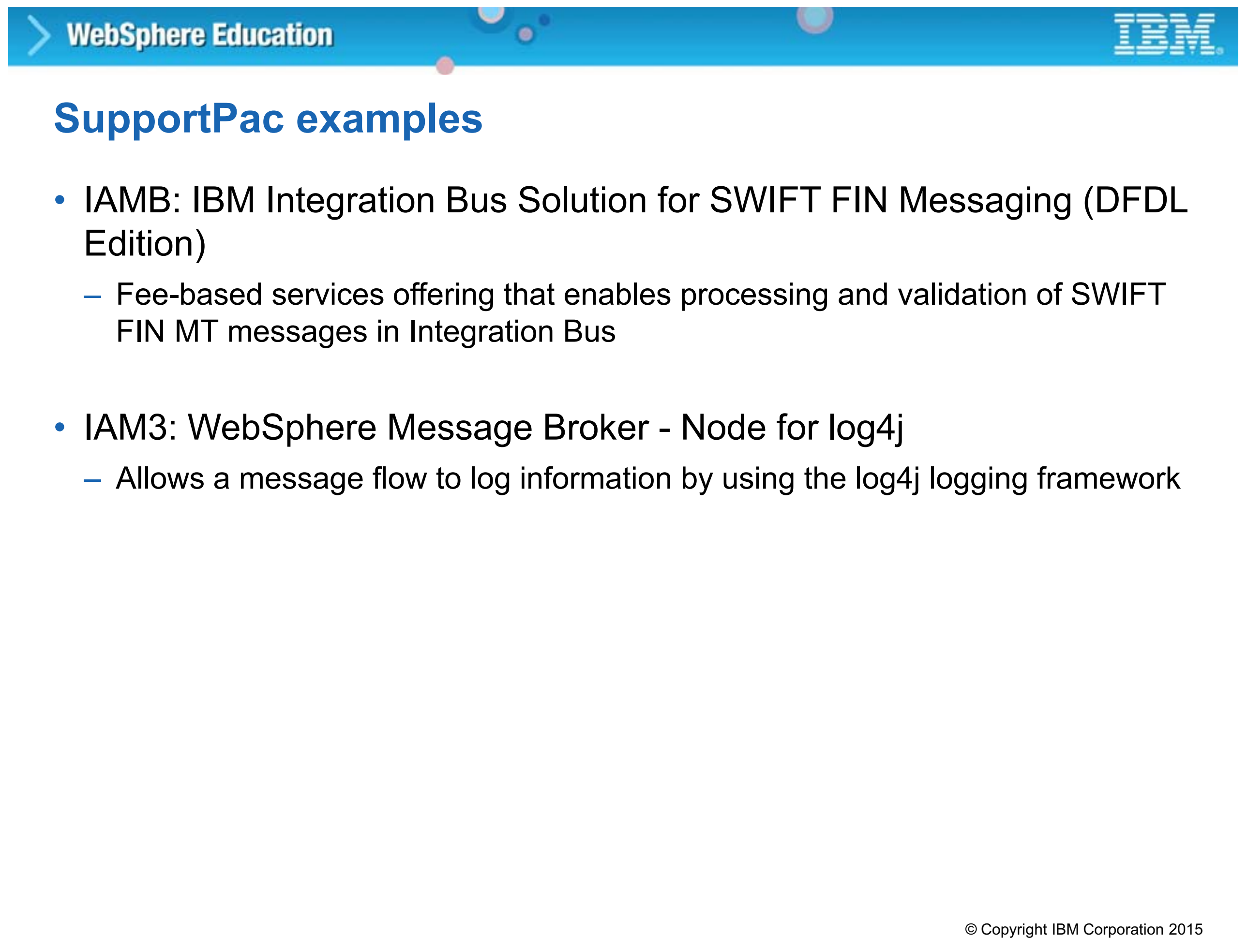
Figure 14-28. Before you start programming WM6461.0

Notes:

If you become involved in a plan to implement user-defined extensions, first suggest a thorough search of already-written extensions. You might find just what you are looking for.

Business Partner offerings usually involve a fee; SupportPac material is normally available at no charge.

As an administrator, you must be involved in the implementation of the finished product, whether purchased, downloaded, or written locally.



WebSphere Education IBM

SupportPac examples

- IAMB: IBM Integration Bus Solution for SWIFT FIN Messaging (DFDL Edition)
  - Fee-based services offering that enables processing and validation of SWIFT FIN MT messages in Integration Bus
- IAM3: WebSphere Message Broker - Node for log4j
  - Allows a message flow to log information by using the log4j logging framework

© Copyright IBM Corporation 2015

Figure 14-29. SupportPac examples WM6461.0

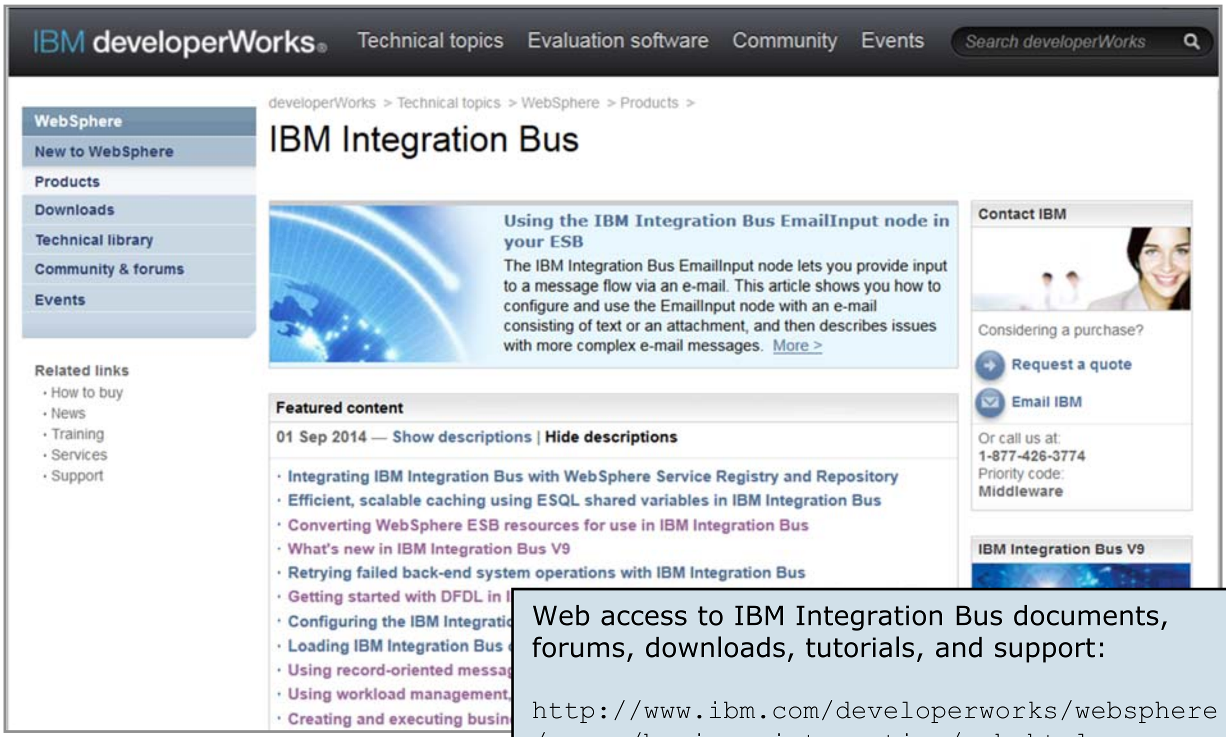
Notes:

Each SupportPac includes a description and installation instructions.

In some cases, older versions of a SupportPac are updated for the current product version. For example, SupportPac IAM3 is a WebSphere Message Broker SupportPac that is updated to support IBM Integration Bus V10.

WebSphere Education

IBM Integration Bus in developerWorks



IBM developerWorks. Technical topics Evaluation software Community Events Search developerWorks

WebSphere
New to WebSphere
Products
Downloads
Technical library
Community & forums
Events

Related links
- How to buy
- News
- Training
- Services
- Support

developerWorks > Technical topics > WebSphere > Products >

IBM Integration Bus

Using the IBM Integration Bus EmailInput node in your ESB

The IBM Integration Bus EmailInput node lets you provide input to a message flow via an e-mail. This article shows you how to configure and use the EmailInput node with an e-mail consisting of text or an attachment, and then describes issues with more complex e-mail messages. More >

Contact IBM

Considering a purchase?

Request a quote

Email IBM

Or call us at:
1-877-426-3774
Priority code:
Middleware

IBM Integration Bus V9

Featured content

01 Sep 2014 — Show descriptions | **Hide descriptions**

- Integrating IBM Integration Bus with WebSphere Service Registry and Repository
- Efficient, scalable caching using ESQL shared variables in IBM Integration Bus
- Converting WebSphere ESB resources for use in IBM Integration Bus
- What's new in IBM Integration Bus V9
- Retrying failed back-end system operations with IBM Integration Bus
- Getting started with DFDL in I
- Configuring the IBM Integratio
- Loading IBM Integration Bus
- Using record-oriented messag
- Using workload management,
- Creating and executing busin

Web access to IBM Integration Bus documents, forums, downloads, tutorials, and support:

`http://www.ibm.com/developerworks/websphere/zones/businessintegration/wmb.html`

Figure 14-30. IBM Integration Bus in developerWorks WM6461.0

Notes:

This figure shows the IBM Integration Bus page in IBM DeveloperWorks. It provides access to documents, forums, downloads, and support.

The link for the IBM Integration Bus DeveloperWorks page is `http://www.ibm.com/developerworks/websphere/zones/businessintegration/wmb.html`

WebSphere Education IBM

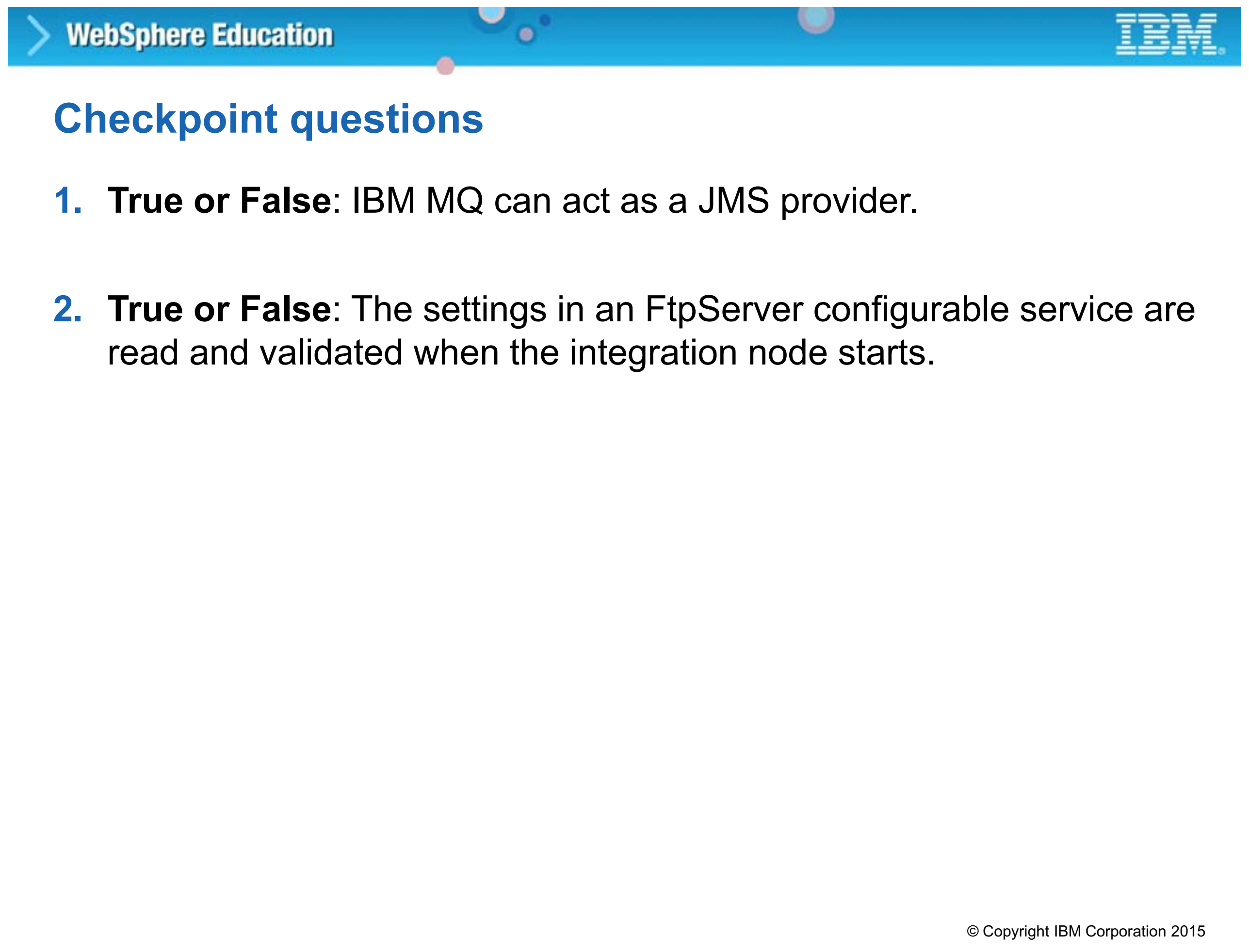
Unit summary

Having completed this unit, you should be able to:

- Enable an integration node to connect to a database with ODBC and JDBC
- Configure a JMS provider for use with the JMS nodes
- Configure IBM Integration Bus for the Secure File Transfer Protocol (SFTP)
- Find and install IBM Integration Bus SupportPac components

Figure 14-31. Unit summary WM6461.0

Notes:



WebSphere Education IBM

Checkpoint questions

1. **True or False**: IBM MQ can act as a JMS provider.

2. **True or False**: The settings in an FtpServer configurable service are read and validated when the integration node starts.

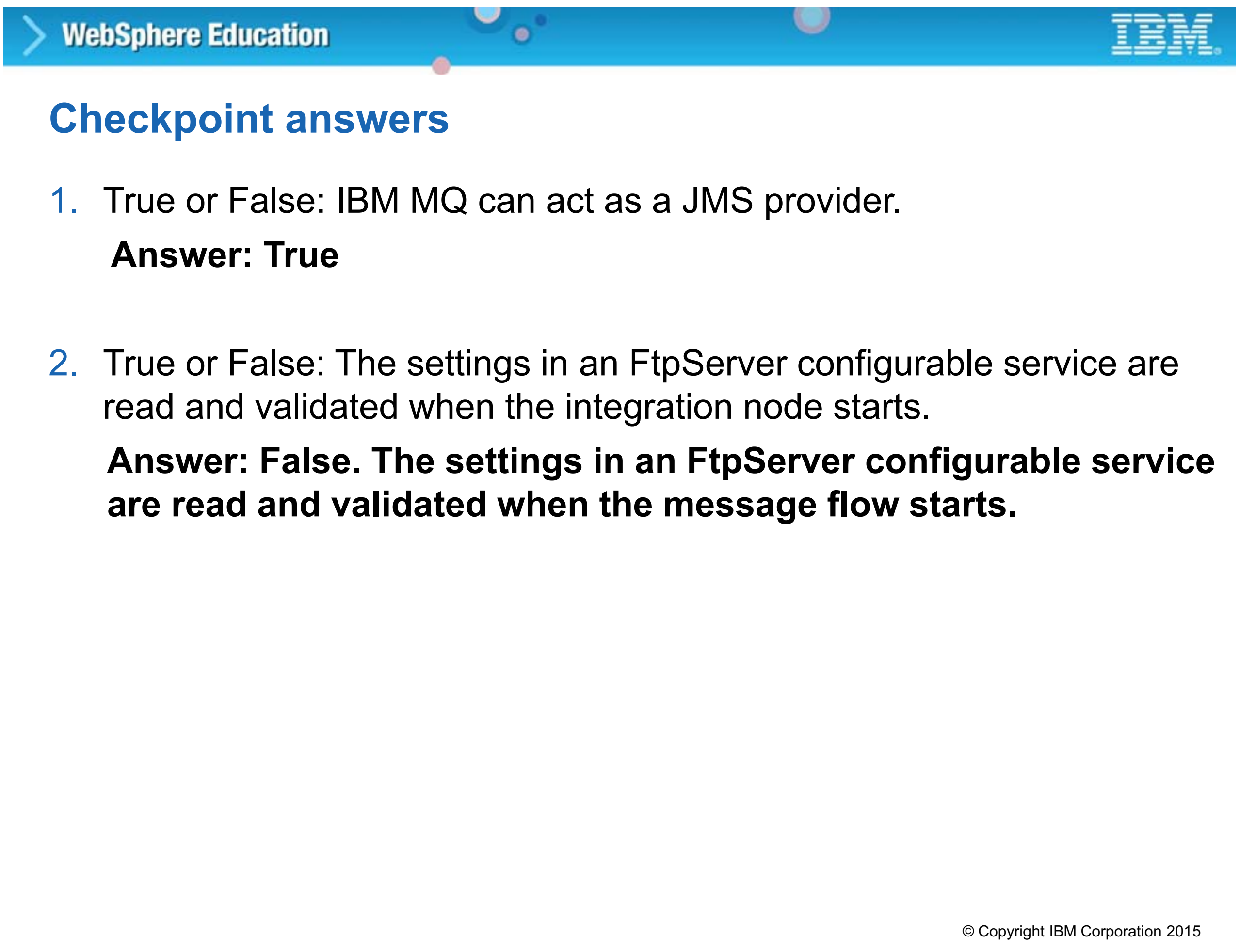
Figure 14-32. Checkpoint questions WM6461.0

Notes:

Write your answers here:

1.

2.



WebSphere Education

Checkpoint answers

1. True or False: IBM MQ can act as a JMS provider.

   Answer: True

2. True or False: The settings in an FtpServer configurable service are read and validated when the integration node starts.

   Answer: False. The settings in an FtpServer configurable service are read and validated when the message flow starts.

Figure 14-33. Checkpoint answers WM6461.0

Notes:

Unit 15. Course summary

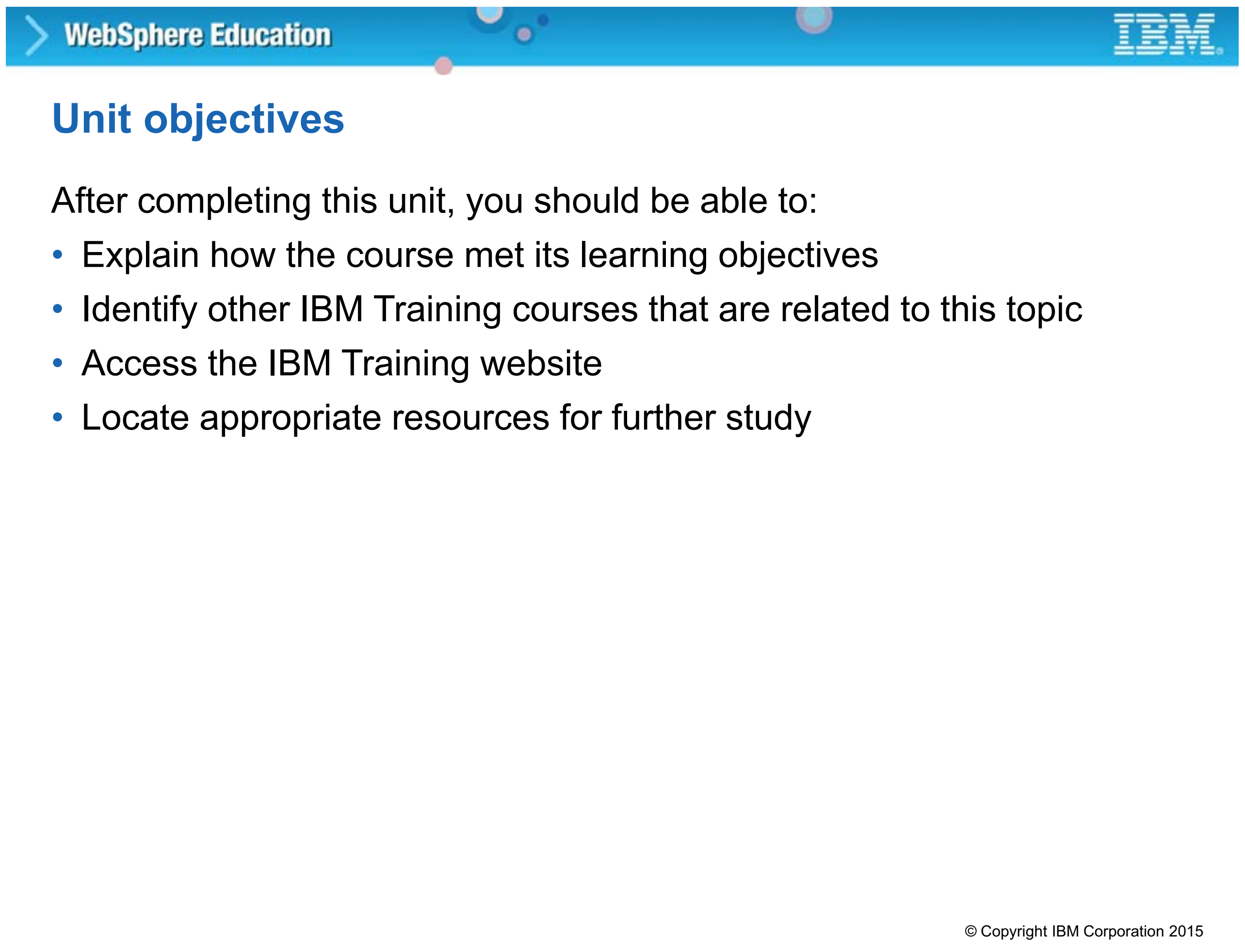
What this unit is about

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

What you should be able to do

After completing this unit, you should be able to:

- Explain how the course met its learning objectives
- Identify other IBM Training courses that are related to this topic
- Access the IBM Training website
- Locate appropriate resources for further study



WebSphere Education

Unit objectives

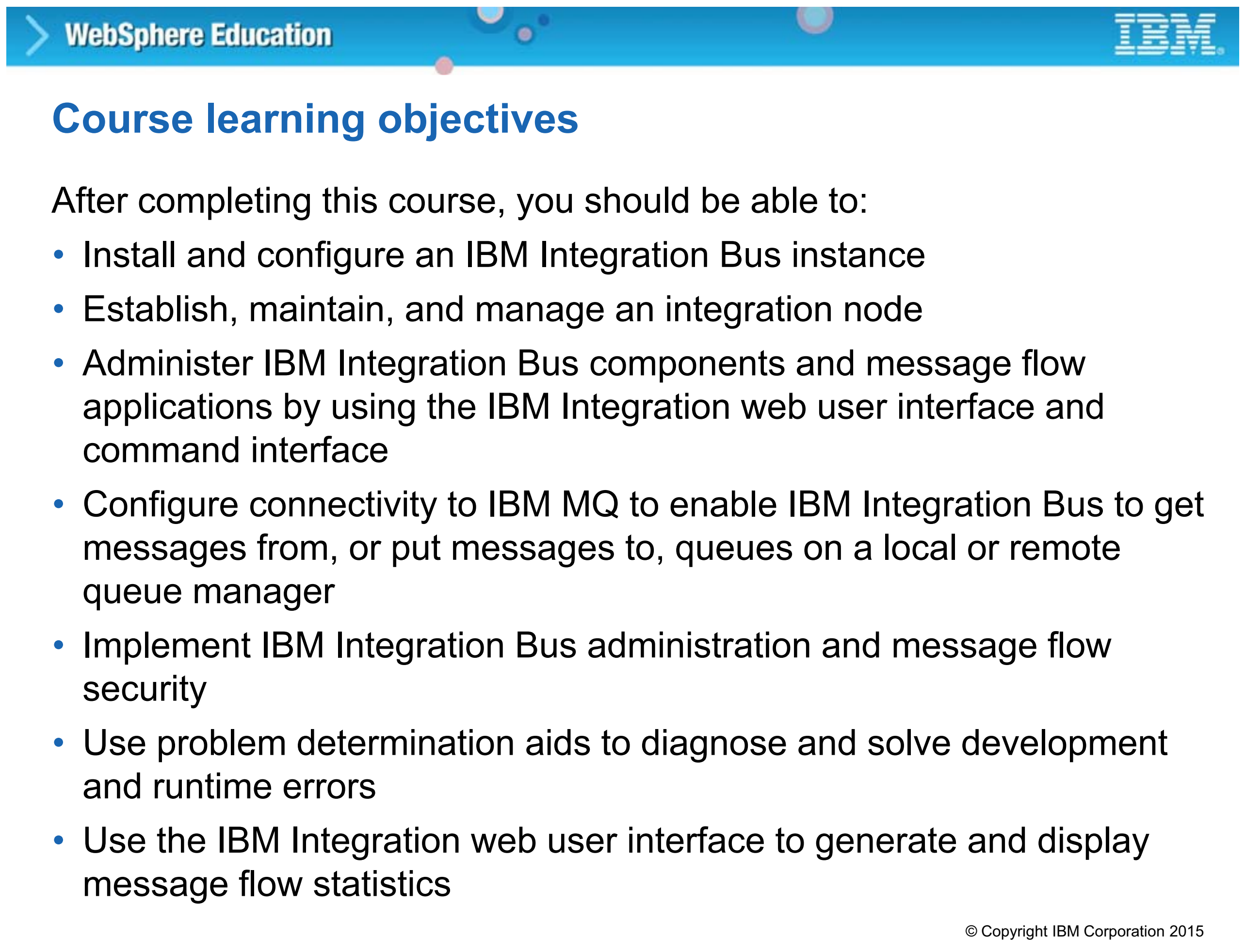
After completing this unit, you should be able to:

- Explain how the course met its learning objectives
- Identify other IBM Training courses that are related to this topic
- Access the IBM Training website
- Locate appropriate resources for further study

© Copyright IBM Corporation 2015

Figure 15-1. Unit objectives WM6461.0

Notes:



WebSphere Education IBM

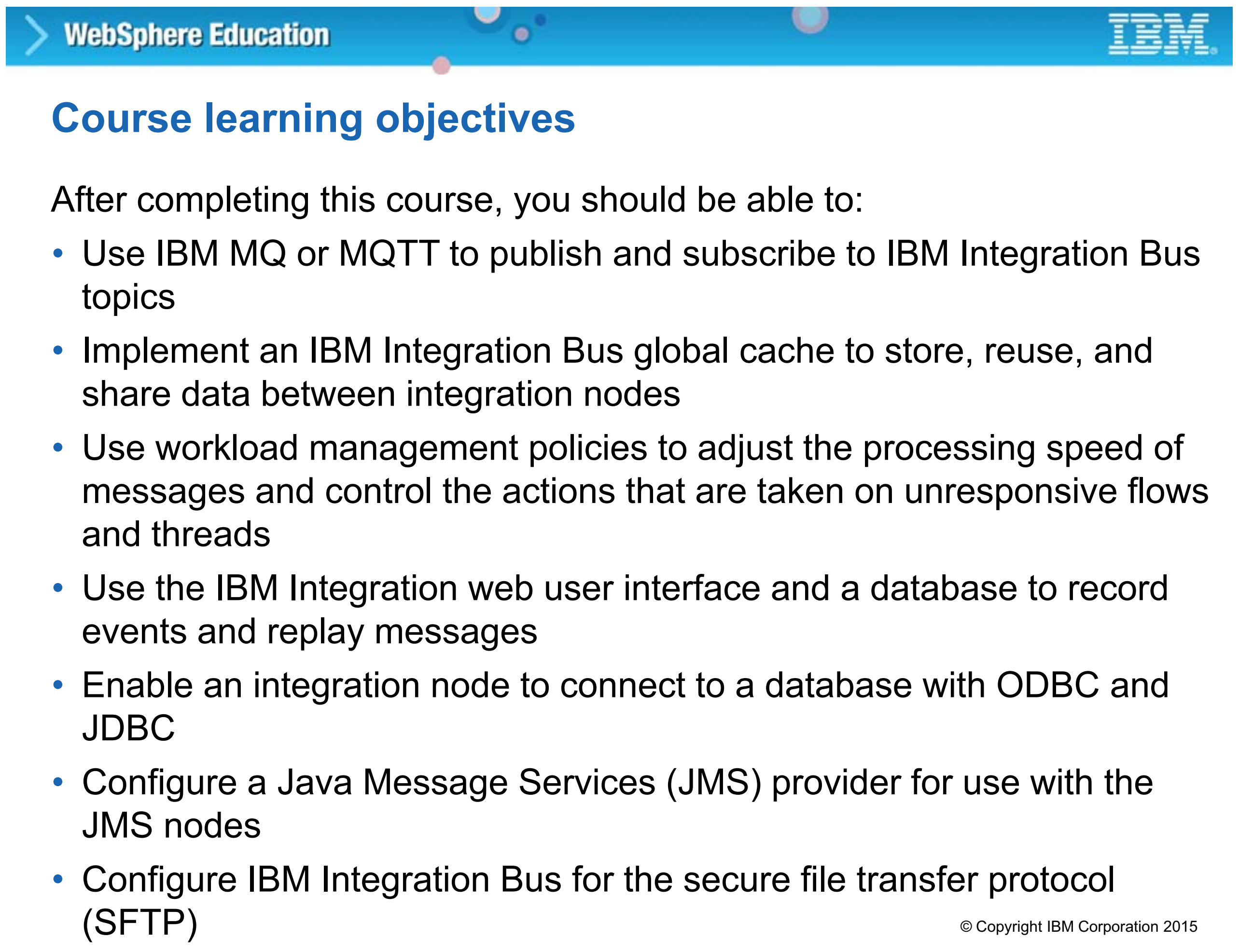
Course learning objectives

After completing this course, you should be able to:

- Install and configure an IBM Integration Bus instance
- Establish, maintain, and manage an integration node
- Administer IBM Integration Bus components and message flow applications by using the IBM Integration web user interface and command interface
- Configure connectivity to IBM MQ to enable IBM Integration Bus to get messages from, or put messages to, queues on a local or remote queue manager
- Implement IBM Integration Bus administration and message flow security
- Use problem determination aids to diagnose and solve development and runtime errors
- Use the IBM Integration web user interface to generate and display message flow statistics

Figure 15-2. Course learning objectives WM6461.0

Notes:



WebSphere Education

Course learning objectives

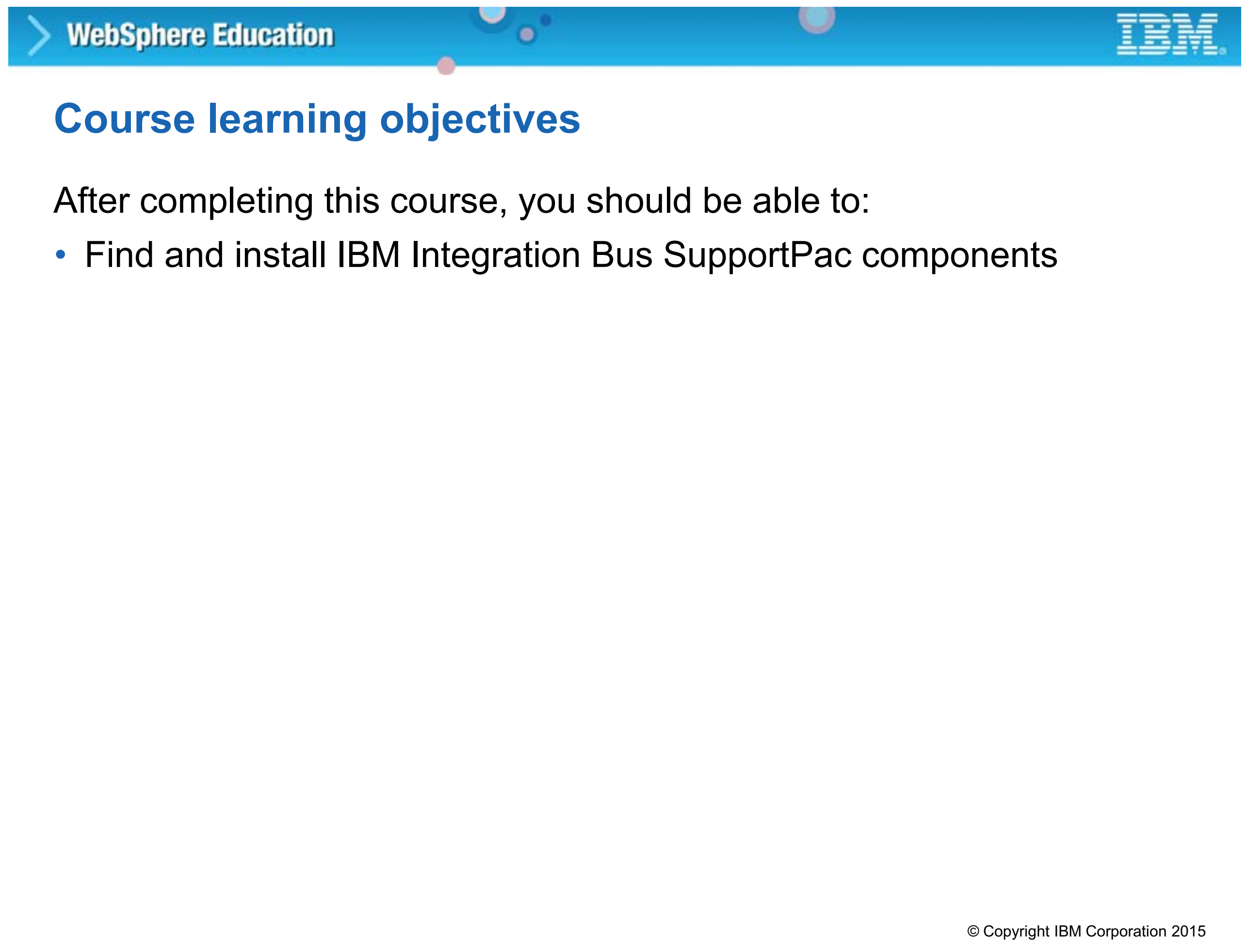
After completing this course, you should be able to:

- Use IBM MQ or MQTT to publish and subscribe to IBM Integration Bus topics
- Implement an IBM Integration Bus global cache to store, reuse, and share data between integration nodes
- Use workload management policies to adjust the processing speed of messages and control the actions that are taken on unresponsive flows and threads
- Use the IBM Integration web user interface and a database to record events and replay messages
- Enable an integration node to connect to a database with ODBC and JDBC
- Configure a Java Message Services (JMS) provider for use with the JMS nodes
- Configure IBM Integration Bus for the secure file transfer protocol (SFTP)

© Copyright IBM Corporation 2015

Figure 15-3. Course learning objectives WM6461.0

Notes:



WebSphere Education

Course learning objectives

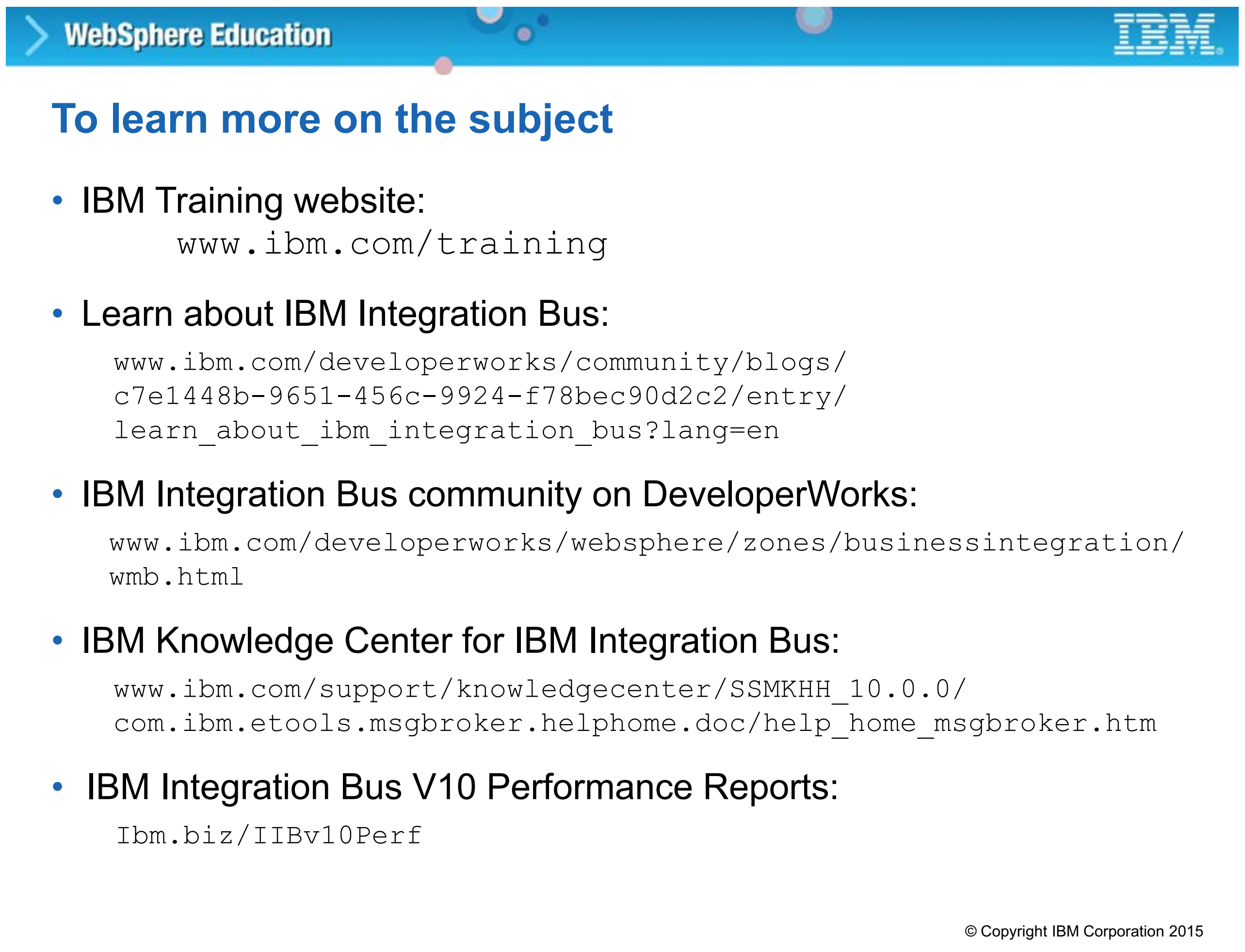
After completing this course, you should be able to:

- Find and install IBM Integration Bus SupportPac components

© Copyright IBM Corporation 2015

Figure 15-4. Course learning objectives WM6461.0

Notes:



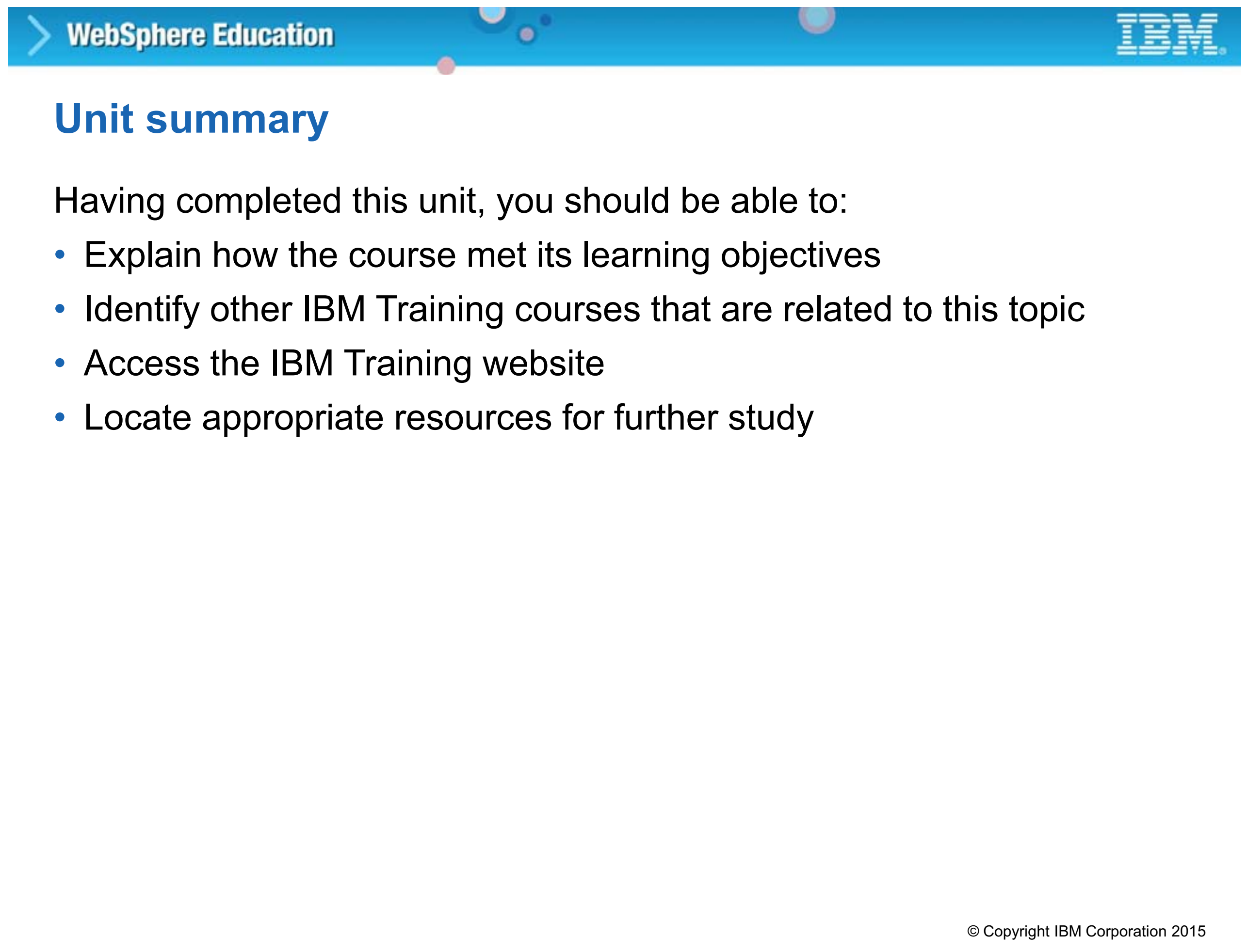
WebSphere Education

To learn more on the subject

- IBM Training website:
  `www.ibm.com/training`
- Learn about IBM Integration Bus:
  `www.ibm.com/developerworks/community/blogs/c7e1448b-9651-456c-9924-f78bec90d2c2/entry/learn_about_ibm_integration_bus?lang=en`
- IBM Integration Bus community on DeveloperWorks:
  `www.ibm.com/developerworks/websphere/zones/businessintegration/wmb.html`
- IBM Knowledge Center for IBM Integration Bus:
  `www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm`
- IBM Integration Bus V10 Performance Reports:
  `Ibm.biz/IIBv10Perf`

Figure 15-5. To learn more on the subject WM6461.0

Notes:



WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Explain how the course met its learning objectives
- Identify other IBM Training courses that are related to this topic
- Access the IBM Training website
- Locate appropriate resources for further study

Figure 15-6. Unit summary WM6461.0

Notes:

Appendix A. List of abbreviations

API	Application programming interface
BAR	broker archive
BPM	Business process monitoring
BPM	business process management
CCDT	Client channel definition table
CDA	IBM Common Debug Architecture
CICS	Customer Information Control System
CMP	Configuration proxy manager
COBOL	Common Business Oriented Language
CORBA	Common Object Request Broker Architecture
CPU	central processing unit
CRL	certificate revocation list
CSV	Comma-separated value
CVS	Concurrent Version System
DFDL	Data Format Description Language
DLQ	dead-letter queue
DSN	Data source name
EDI	electronic data interchange
EIS	enterprise information system
ENF	event notification flag
ERP	Enterprise resource planning
ESQL	Extended Structured Query Language
FTP	File transfer protocol
HA	High availability
HL7	Health Level 7
HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
IBM	International Business Machines Corporation
ID	identifier
JAR	Java archive
JCA	Java EE Connector Architecture

JDBC	Java Database Connectivity
JDWP	Java Debug Wire Protocol
JMS	Java message service
JNDI	Java Naming and Directory Interface
JSON	JavaScript Object Notation
JVM	Java virtual machine
KDC	Key Distribution Center
LDAP	Lightweight Directory Access Protocol
LTPA	Lightweight Third-Party Authentication
MAP	Message Addressing Properties
MQMD	MQ message descriptor
MQTT	MQ Telemetry Transport
NAS	Network attached storage
NFS	network file system
ODBC	Open Database Connectivity
OMVS	Open Multiple Virtual Storage
PAR	Package archive
PKI	Public key infrastructure
POP3	Post Office Protocol Version 3
RACF	Resource Access Control Facility
REST	Representational State Transfer
RPC	Remote procedure call
RRS	Resource Recovery Services
SAML	Security Assertion Markup Language
SCA	Service component architecture
SCM	Software configuration management
SDSF	System Display and Search Facility
SFTP	Secure file transfer protocol
SMP/E	SMP/E for z/OS
SOA	service-oriented architecture
SOAP	A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP can be used to query and return information and invoke services across the Internet.
SSH	Secure shell

SSL	Secure socket layer
STS	Security Token Service
TCP/IP	Transmission Control Protocol/Internet Protocol
TFIM	Tivoli Federated Identity Manager
TLS	Transport Layer Security
UDB	Universal database
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
UUID	Universally unique identifier
WS	Web Services
WSDL	Web services description language
XA	extended architecture
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSTL	XSL transformations

Appendix A. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: `http://www.ibm.com/training`
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: `http://bit.ly/IBMTrainEN`
- **IBM Certification**
  - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
  - For more information, see: `http://www.ibm.com/certify`
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
  - For more information, see: `http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0003096`

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the latest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
  - Receive concise updates from Middleware Education a few times each week.
  - Follow Middleware Education at: `twitter.com/websphere_edu`
- **Facebook**:
  - Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.

  - Find IBM Training at: `facebook.com/ibmtraining`
- **YouTube**:
  - See the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: `youtube.com/IBMTraining`

Support

- **Middleware Support portal**
  - The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARS). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
  - For more information, see: `http://www.ibm.com/software/websphere/support`
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: `http://www.ibm.com/software/support/isa`
- **IBM Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.
  - For more information, see:
    `http://www.ibm.com/software/info/education/assistant/`

Middleware documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.

    - For more information, see: `http://www.ibm.com/redbooks`

- **IBM documentation and libraries**
    - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
    - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
    - For more information, see: `http://www.ibm.com/developerworks`

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see:
  `http://www.ibm.com/services/us/en/it-services/systems/middleware-services/`