



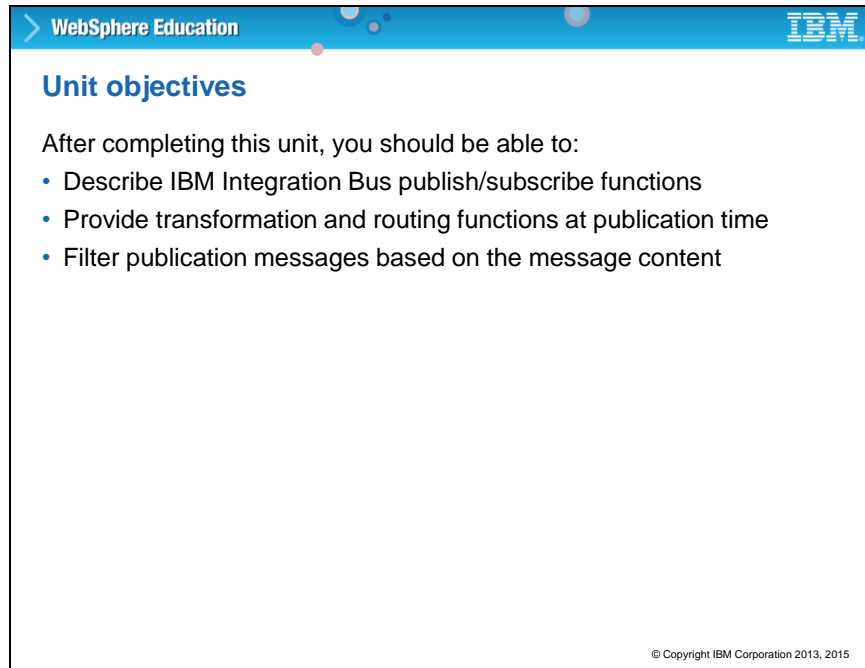
Slide 1


> WebSphere Education 

Implementing publish/subscribe



© Copyright IBM Corporation 2013, 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

A presentation slide with a blue header bar containing the text 'WebSphere Education' and the IBM logo. The main content area is white and contains the title 'Unit objectives' in blue. Below the title, it states 'After completing this unit, you should be able to:' followed by a bulleted list of three objectives. The footer contains a small copyright notice.

> WebSphere Education 

Unit objectives

After completing this unit, you should be able to:

- Describe IBM Integration Bus publish/subscribe functions
- Provide transformation and routing functions at publication time
- Filter publication messages based on the message content

© Copyright IBM Corporation 2013, 2015

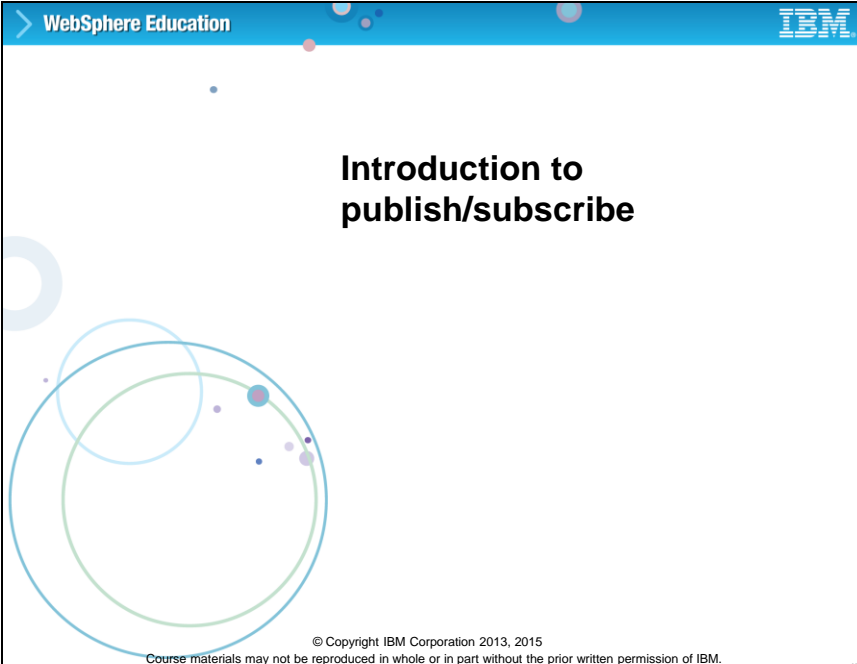
Unit objectives

IBM Integration Bus uses publish/subscribe to notify applications of significant events that occur in integration nodes. IBM Integration Bus can also act as a content filtering provider for IBM MQ to allow subscribers to refer to elements in the body of publications. This unit describes how IBM Integration Bus supports these publish/subscribe applications.

After completing this unit, you should be able to:

- Describe IBM Integration Bus publish/subscribe functions
- Provide transformation and routing functions at publication time
- Filter publication messages based on the message content

Slide 3



The slide features a blue header bar with the text "WebSphere Education" on the left and the IBM logo on the right. The main title "Introduction to publish/subscribe" is centered in a bold, black font. To the left of the title is a decorative graphic consisting of several overlapping circles in light blue and green, with small colored dots scattered around them. At the bottom of the slide, there is a small copyright notice: "© Copyright IBM Corporation 2013, 2015. Course materials may not be reproduced in whole or in part without the prior written permission of IBM."

WebSphere Education

IBM

Introduction to publish/subscribe

© Copyright IBM Corporation 2013, 2015
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Topic 1: Introduction to publish/subscribe

This topic describes IBM Integration Bus publish/subscribe functions.

WebSphere Education

IBM

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

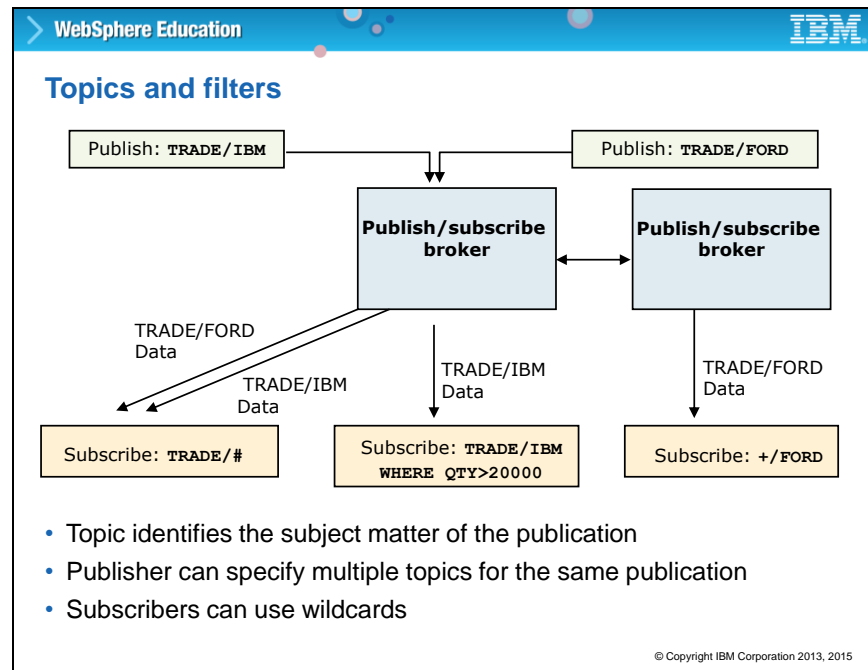
```
graph LR; Publisher((Publisher)) -- Publication --> Broker[Publish/subscribe broker]; Subscriber((Subscriber)) -- Subscription --> Broker; Broker -- Publication --> Subscriber;
```

© Copyright IBM Corporation 2013, 2015

What is publish/subscribe?

A publish/subscribe topology decouples the sender and the receiver of a message. Message senders, or publishers, create topics. Applications or users can subscribe and unsubscribe to a topic. Facebook and Twitter are examples of publish/subscribe applications.

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.



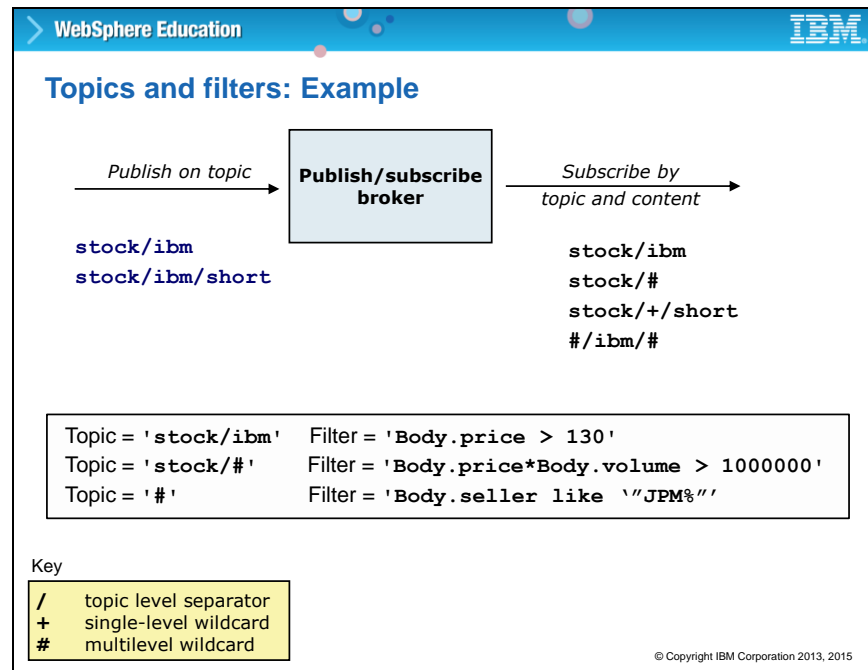
Topics and filters

A publisher specifies a topic identifier that represents the subject matter of the publication.

The topic string can be any length. A topic tree structure can be built by qualifying the topic with the forward slash character. Publishers can specify multiple topics for the same publication.

Subscribers subscribe to topics and can use wildcards. A plus sign character matches exactly one topic level. The hashtag character 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

A subscription must include a topic. A hashtag character denotes a generic topic. Filters are optional.

The slide shows examples of valid topic and filter strings by using topic level separators, single-level wildcards, and multilevel wildcards.

The example shows that the subscriber can subscribe to a specific topic by specifying `stock/ibm`. A subscriber might subscribe to all stocks by specifying `stock/#`. A subscriber might get information about all shorted stocks by specifying `stock/+/short`. Finally, a subscriber can subscribe to all information about IBM by specifying `#/ibm/#`.

The example also shows how filter strings can supplement the topic strings to get information based on conditions. For example, the first string subscribes to IBM stock information when the price of the stock is greater than 130. The second string subscribes to any stocks with volumes over 1 million shares. The third string subscribes to any topic where the seller contains the characters "JPM".



Topic specifications



- Topic names are case-sensitive
For example, `ACCOUNTS` and `Accounts` are two different topics
- Topic names can include the space character
- A topic can contain one or more `/` characters; this character separates **levels** in the topic. Levels are used to define hierarchies within a topic
- A leading `/` creates a distinct topic
For example, `/Country` is different from `Country`
- You can reference multiple topics and topic levels by using the wildcard characters `+` and `#`
 - `+` is the single-level wildcard character; matches exactly one topic level
 - `#` is the multi-level wildcard character; matches any number of topic levels

© Copyright IBM Corporation 2013, 2015

Topic specifications

You must follow some rules when defining topics:

- Topic names are case-sensitive.
- Topic name can include a space character.
- You can create a hierarchy to a topic to define a topic “tree” by using a forward slash character. For example, you can create a topic *InventoryMessage*, and then create subtopics such as *InventoryMessage/Sold*, *InventoryMessage/Restock*, and *InventoryMessage/ReturnToInventory*. You can create more subtopics for any of these topics. There is no limit to the number of levels in a topic tree, nor is there any limit to the length of the name of a level in a topic tree.
- When you reference topics, you can specify wildcard characters to match one or more levels of a topic string.

 WebSphere Education 

Integration Bus and publish/subscribe support (1 of 2)



- Integration Bus 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
- MQTT or IBM MQ publish/subscribe brokers notify external applications of significant events that occur in integration nodes
 - Operational events (message flow statistics, resource statistics, and workload management)
 - Administrative events (integration node status, integration server status, and integration server configuration)
 - Business events (business monitoring)
- Choose the publish/subscribe broker based on processing requirements and your existing architecture

© Copyright IBM Corporation 2013, 2015

Integration Bus and publish/subscribe support (1 of 2)

Integration Bus enhances publish/subscribe applications in two ways. First, Integration Bus can provide transformation and routing functions at publication time. Second, Integration Bus can filter messages based on the content of the message.

A publish/subscribe broker ensures that messages are delivered to the correct subscribers. Integration Bus supports two types of publish/subscribe broker, MQTT, and MQ. You can choose which type to use based on your processing requirements and your existing architecture.

 WebSphere Education 

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 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 of the publish/subscribe broker by using the `mqsischangeproperties` command

Example:

```
mqsischangeproperties IBNODE -b pubsub  
-o BusinessEvents/MQTT -n enabled -v true
```

- Integration Bus built-in nodes for publish/subscribe
 - MQTT Subscribe
 - MQTT Publish
 - Publication

© Copyright IBM Corporation 2013, 2015



Integration Bus and publish/subscribe support (2 of 2)

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. If MQ is installed and a queue manager is specified on the integration node, you can use the MQ broker instead of or in addition to the MQTT broker.

Besides enhancing an existing publish/subscribe network, Integration Bus uses a publish/subscribe broker to publish administration events to well-known topics. As you learn in Unit 16, you can also configure a message flow to emit event messages that can be used to support transaction monitoring and auditing, and business process monitoring.

You can modify the configuration of the MQTT broker by using the `mqsischangeproperties` command.

Integration Bus also includes built-in message processing nodes for publish/subscribe. You learn more about configuring the MQTT and MQ brokers and the publish/subscribe nodes later in this unit.



Viewing publish/subscribe properties

- Use the `mqsireportproperties` command with `-b pubsub` to examine the values of the publish/subscribe properties on an integration node

Example:

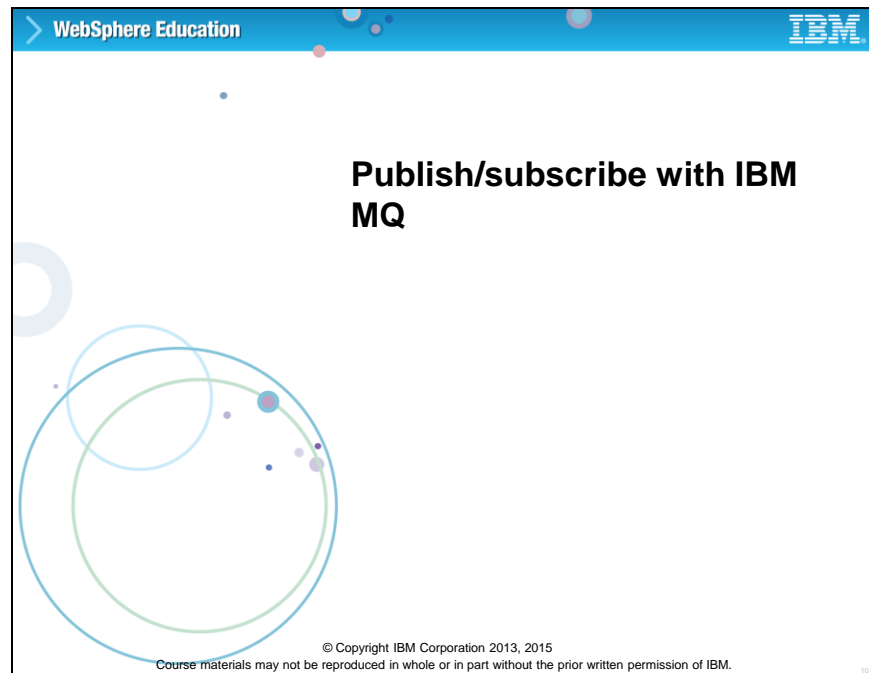
```
mqsireportproperties IBNODE -b pubsub -o AllReportableEntityNames -r
```

```
OperationalEvents
MQ
  policyUrl=''
  enabled='true'
  format=''
MQTT
  policyUrl=''
  enabled='true'
AdminEvents
MQ
  policyUrl=''
  enabled='true'
  format=''
MQTT
  policyUrl=''
```

© Copyright IBM Corporation 2013, 2015

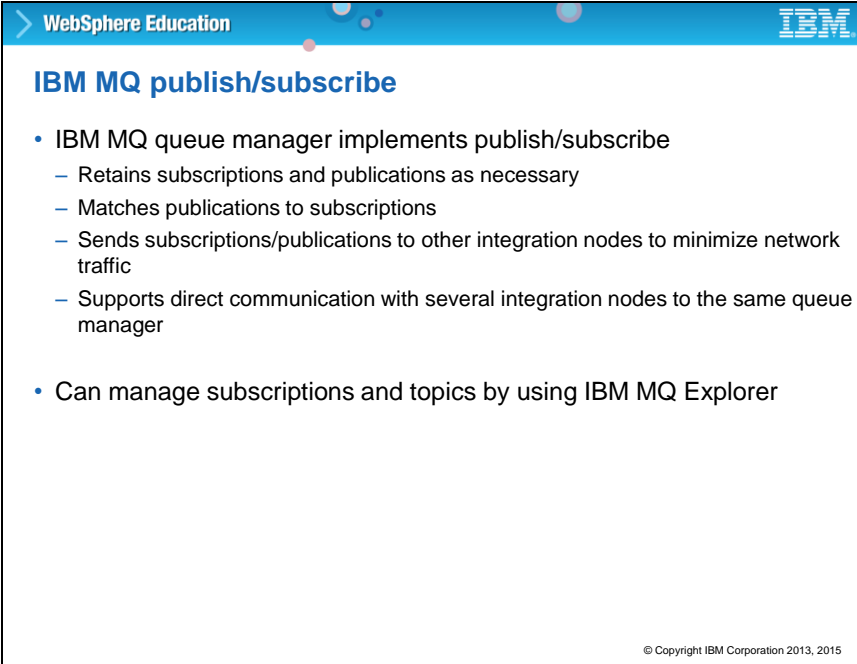
Viewing publish/subscribe properties

Integration Bus publishes operational events, administration events, and optionally business events. You can use the `mqsireportproperties` command with the `-b pubsub` option to determine whether the MQ or the built-in MQTT publish/subscribe broker is enabled for an integration node event type.




Topic 2: Publish/subscribe with IBM MQ

If you installed MQ and the integration node is associated with a queue manager, you can use the MQ publish/subscribe broker with Integration Bus. This topic describes how to integrate Integration Bus with the MQ publish/subscribe broker.



The slide features a blue header bar with the text 'WebSphere Education' on the left and the IBM logo on the right. The main content area is white with a blue title 'IBM MQ publish/subscribe'. Below the title is a bulleted list of features. At the bottom right, there is a small copyright notice.

> 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
- Can manage subscriptions and topics by using IBM MQ Explorer

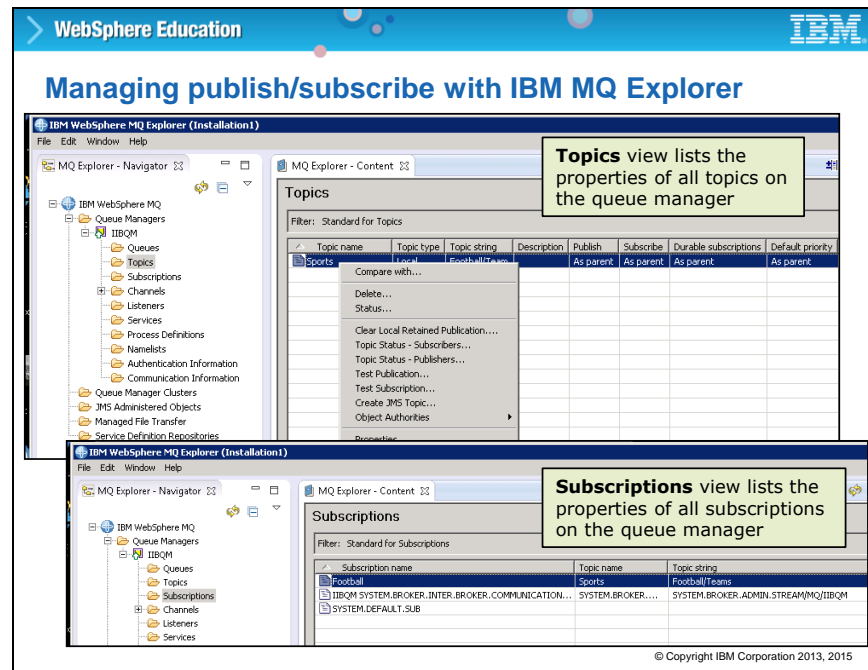
© Copyright IBM Corporation 2013, 2015

IBM MQ publish/subscribe

You can use MQ to support publish/subscribe applications.

MQ retains subscriptions and publications as necessary. It also matches publications to subscriptions.

When integrated with Integration Bus, MQ can send subscriptions and publications to other integration nodes to minimize network traffic. MQ also supports direct communication with several integration nodes to the same queue manager.



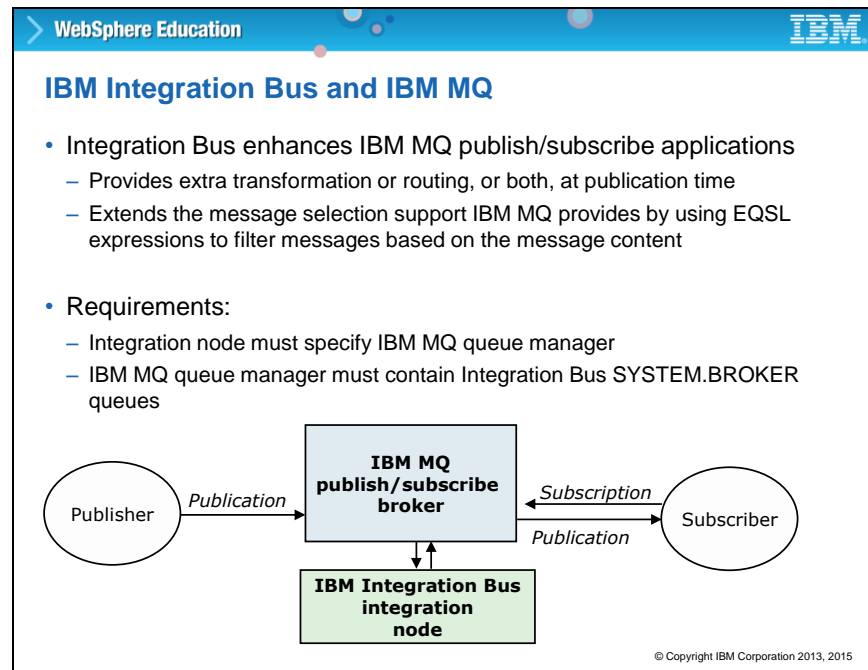
Managing publish/subscribe with IBM MQ Explorer

You can use MQ Explorer or MQ commands to manage publication topics and subscriptions.

To access the topics on a queue manager, click the **Topics** folder under the queue manager name in the MQ Explorer Navigator view. From the **Topics** view, you can create new topics and check topic status. You can also test publications.

To access the subscriptions on a queue manager, click the **Subscriptions** folder under the queue manager name in the **MQ Explorer Navigator** view. From the **Subscriptions** view, you can create new subscriptions and check subscription status. You can also test subscriptions.

IBM MQ publish/subscribe is taught in course ZM212, *IBM MQ V8 Advanced System Administration (Distributed)*.

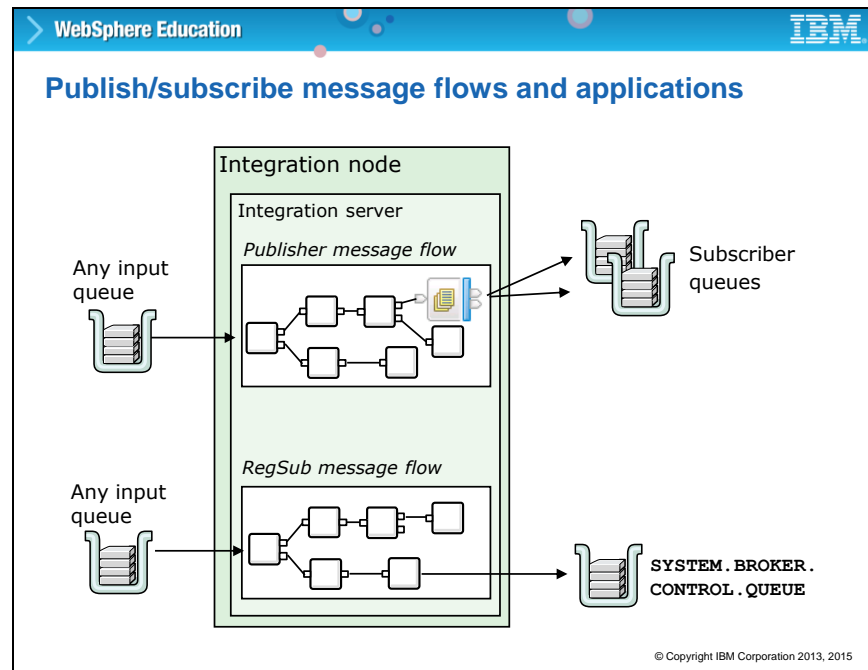


IBM Integration Bus and IBM MQ

MQ can specify a filter when a subscription is made, but the filter can refer to items in the MQ headers only. Integration Bus enhances MQ publish/subscribe applications by acting as a content filter provider for MQ. Integration Bus allows subscriptions to specify extended filters 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 MQ content filtering require that the integration node is associated with a queue manager and that the queue manager contains the Integration Bus SYSTEM.BROKER queues.



Publish/subscribe message flows and applications

Message flows can complete publish/subscribe tasks for applications. This approach allows existing applications to publish messages or register as a subscriber without being modified with publish/subscribe commands.

In the first example, a publishing application publishes messages by using a Publication node. Those messages are sent to the subscriber queues by Integration Bus and MQ.

In the second example, a Register Subscription (RegSub) message flow registers subscriptions by reading from an input queue and then writing a Register Subscription message to the SYSTEM.BROKER.CONTROL.QUEUE. A more detailed example of this technique is shown next.

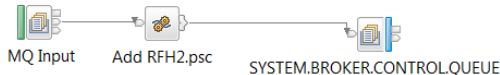
WebSphere Education

IBM

Registering subscriptions in IBM Integration Bus (1 of 2)

You can register subscriptions in a number of different ways:

- Use a message flow to send application message that contains RegSub command to SYSTEM.BROKER.CONTROL.QUEUE



```
graph LR; MQInput[MQ Input] --> AddRFH2[Add RFH2.psc]; AddRFH2 --> Output[SYSTEM.BROKER.CONTROL.QUEUE];
```

- Use IBM MQ SupportPac IH03, RfhUtil, to register static subscriptions manually


© Copyright IBM Corporation 2013, 2015

Registering subscriptions in IBM Integration Bus (1 of 2)

Integration Bus does not contain a specific “Subscribe” message processing node for the MQ broker. You can build a message flow that receives messages on any input queue, build an MQRFH2 header that contains the register subscriber command, and send this message to SYSTEM.BROKER.CONTROL.QUEUE.

The alternative is to use a tool such as RfhUtil to build a subscription statically. You should be aware that static registration means that the Integration Bus administrators must take responsibility for maintaining and deleting subscriptions for subscriber applications. Managing subscriptions in this way does not take advantage of the dynamic nature of subscriptions.

Details on how to format the MQRFH2 header to register a subscription are next.

WebSphere Education


Registering subscription in IBM Integration Bus (2 of 2)

- Use Compute node to set the MQRFH2 header

```

SET OutputRoot.MQRFH2.psc.Command = 'RegSub';
SET OutputRoot.MQRFH2.psc.Topic = 'Complaint/Order/#';
SET OutputRoot.MQRFH2.psc.Filter = 'Body.Complaint.Text like
''%late%''';
SET OutputRoot.MQRFH2.psc.QName = 'MYQUEUE';
SET OutputRoot.MQRFH2.psc.QMgrName = 'MYQM';

```
- Integration Bus replies are sent to `MQMD.ReplyToQueue` if the subscription was a request message
- Queue manager automatically deletes subscriptions for the following cases:
 - Subscription expires (`MQMD.Expiry` in `RegisterSubscriber` message)
 - Temporary (local) delivery queue deleted
 - Subscriber application ends

© Copyright IBM Corporation 2013, 2015

Registering subscription in IBM Integration Bus (2 of 2)

The MQRFH2 folder contains a number of folders that applications that communicate with MQ use, including:

- Message content descriptor (mcd)
- Publish/subscribe command (psc)
- Publish/subscribe command reply (pscr)
- User-defined properties (usr)
- Java Messaging Service (jms)

To programmatically register a subscription, you can use Compute or JavaCompute node to fill the MQRFH2.psc header, and then send the message to MQ. When the integration node receives a `RegSub` command from `SYSTEM.BROKER.CONTROL.QUEUE`, it first checks the validity of the subscription. It is not possible to subscribe with a subscription queue that does not exist. So either a local queue or a local definition of a remote queue must exist on the integration node queue manager for `psc.QName`. If the header doesn't contain a `psc.QName` in the MQRFH2 header, the `MQMD.ReplyToQueue` is used as subscriber queue instead.

The integration node also checks subscription conditions to ensure that no registered subscription results in the same publication that is delivered to any subscriber queue twice.

You can check the success of the RegSub command by using the normal MQ request/reply mechanism. Integration Bus sends a reply message to the reply queue specified in the MQMD of the subscription message.

You can send a Delete Publication request in a similar way.

WebSphere Education

IBM

Publication node

- Publish a message through the IBM MQ queue manager that is specified on the integration node
- Filters output messages from a message flow and transmits them through an IBM MQ publish/subscribe broker to subscribers
- Only one copy of a message is sent to each subscriber queue, regardless of how many subscriptions match
- Uses the topic from MQ Input node
 - Can override in `Properties.Topic` or `MQRFH2.psc.Topic` by using a compute-type node

```
SET OutputRoot.MQRFH2.psc.Command = 'Publish';
SET OutputRoot.MQRFH2.psc.Topic   =
CASE InputBody.Message.Complaint.Type
  WHEN 'Order' THEN 'Complaint/Order'
  WHEN 'Delivery' THEN 'Complaint/Delivery'
  ELSE 'Complaint/Other'
END;
```

© Copyright IBM Corporation 2013, 2015

Publication node

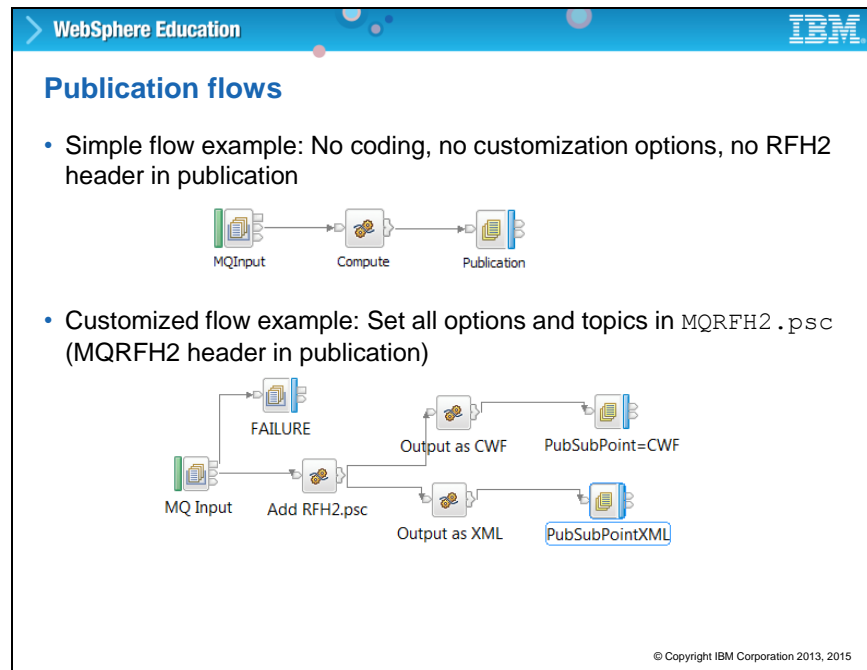
Integration Bus provides a Publication node that sends the publication message to the `SYSTEM.BROKER.CONTROL.QUEUE`.

You can identify an MQ subscription point within the properties of the Publication node. The subscription point differentiates multiple Publication nodes in the same message flow. It represents a specific path through the message flow, for example, if the same message is published in different formats.

Subscribers can choose to which subscription point they subscribe. An unnamed Publication node (that is, one without a specific subscription point) is known as the default Publication node.

The Publication node always tries to deliver its publication. If the queue is not reachable, the integration node attempts to put the message to the dead-letter queue. If it is not possible to use the dead-letter queue and the publication flow is transactional, all publications are rolled back.

Having zero registered subscribers does not cause an error. The incoming message is propagated through the **NoMatch** terminal of the Publication node.





Publication flows

This slide contains two examples of publication flows.

The simple flow gets a message from an MQ queue, processes the message in a Compute node, and then publishes the message.

The second flow manipulates the publish/subscribe header in the message and publishes in two different formats.

 WebSphere Education 

Content-based filtering

- Integration Bus can act as a content filtering provider for IBM MQ to allow extended filters to message body based on subscribers selection string
 - Uses ESQL expressions to filter on the entire message by using the `Root`, `Body`, and `Properties` correlation names
 - Accesses the message tree as read-only
- Integration Bus provides content filtering services for 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 2013, 2015

Content-based filtering

MQ supports a limited amount of filtering on the header and message properties, but it cannot filter on the body of the message because it typically cannot parse this part of the message. Because Integration Bus can parse the message body, it extends the MQ filtering capability.

Content-based filtering allows a subscriber to restrict or filter messages that it wants to receive, in addition to the topics that it specified. This filter is specified as an SQL expression.

If you want to support content-based filtering, you must enable it on at least one integration server on the integration node.



Content-based filtering configuration

- Enabled by using the `mqsichangeproperties` command with the `-o ContentBasedFiltering` and `-n cbfEnabled` properties
Example:

```
mqsichangeproperties IBNODE -e ISPUSSUB  
-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

© Copyright IBM Corporation 2013, 2015

Content-based filtering configuration

You enable content-based filtering on an integration server by using the `mqsichangeproperties` command with the `-o ContentBasedFiltering` option, as shown here. You can also enable content-based filtering by using the Integration web user interface.

On distributed operating systems such as Windows and Linux, you must grant system authorization to the queue manager from the integration node by using the MQ `setmqaut` command.

You must restart the integration server for the changes to take effect.

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

The screenshot shows the WebSphere Integration web interface. On the left, a navigation tree shows the hierarchy: IB10NODE > Servers > default > IS1. The main panel displays the 'Resource Manager Properties' for the selected integration server. The 'Content Based Filtering' section is highlighted with a red box, showing the following settings:

Property	Value
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 2013, 2015



Configure content-based filtering the Integration web interface

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, as shown in the example.

The slide is titled "Publish/subscribe with MQTT" in bold black text. It features a diagram on the left side consisting of two overlapping circles, one light blue and one light green, with several small colored dots (blue, purple, red) scattered around them. The slide has a blue header bar with "WebSphere Education" on the left and the "IBM" logo on the right. At the bottom, there is a copyright notice: "© Copyright IBM Corporation 2013, 2015" and "Course materials may not be reproduced in whole or in part without the prior written permission of IBM."

Topic 3: Publish/subscribe with MQTT

This topic describes how Integration Bus integrates with the built-in MQTT broker.

 WebSphere Education 

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 one time
 - 1 = message is delivered but might be duplicated
 - 2 = one time 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 MQTT clients create

© Copyright IBM Corporation 2013, 2015

MQTT overview

MQTT 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.

MQTT supports three qualities of service for message delivery:

- "At most once" delivers messages according to the best efforts of the underlying Internet Protocol network. Message loss or duplication can occur. This level might be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.
- "At least once" assures that messages arrive but duplicates might occur.
- "Exactly once" assures that messages arrive exactly once. This level might be used, for example, with billing systems where duplicate or lost messages might lead to applying incorrect charges.

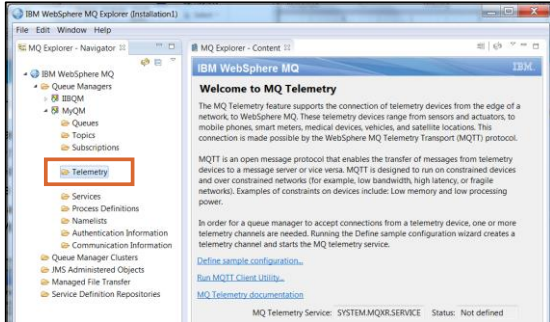
Integration Bus includes a built-in MQTT broker for publishing Integration Bus events. As you learn later in the topic, Integration Bus also includes message processing nodes for MQTT publish and subscribe.

WebSphere Education

IBM

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 IBM MQ Explorer when you install the **Telemetry** option





© Copyright IBM Corporation 2013, 2015

MQTT administration with IBM MQ Explorer

If you installed MQ with the Telemetry option, you can use MQ to manage MQTT topics and subscriptions.

MQ Explorer also provides a sample MQTT configuration and client utilities.

 WebSphere Education



MQTT support in IBM Integration Bus


- Integration Bus provides built-in message processing nodes for processing MQTT messages
- MQTT broker that is provided with Integration Bus provides publish/subscribe for Integration Bus events
 - Enabled by default for Administrative and Operational events
- Events can be routed to IBM MQ (when a queue manager is associated with an integration node) or MQTT broker
 - MQTT publications can be routed to an external MQTT broker such as IBM MessageSight
 - Events are published to IBM MQ queue manager associated with integration node for compatibility with an earlier version

© Copyright IBM Corporation 2013, 2015

MQTT support in IBM Integration Bus

Integration Bus publishes status and event information in known topics. Integration Bus includes a built-in MQTT broker and MQTT publication is enabled by default for all events that the integration node generates, apart from business events. If you choose not to use the built-in MQTT broker, you can specify an alternative MQTT broker, such as MQ.

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.

WebSphere Education


Configuring the Integration Bus MQTT broker

- Modify the configuration 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 that the broker uses (default is `11883`)
- View the current configuration 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
```
- Use the `enabled` property of the `mqsichangeproperties` command to enable or disable the publication of event messages by a specified publish/subscribe broker (`OperationalEvents`, `AdminEvents`, and `BusinessEvents`)

Example:

```
mqsichangeproperties IBNODE -b pubsub -o BusinessEvents/MQTT -n enabled -v true
```

© Copyright IBM Corporation 2013, 2015

Configuring the Integration Bus MQTT broker

You can modify the configuration of the built-in MQTT broker by using the `mqsichangeproperties` command.

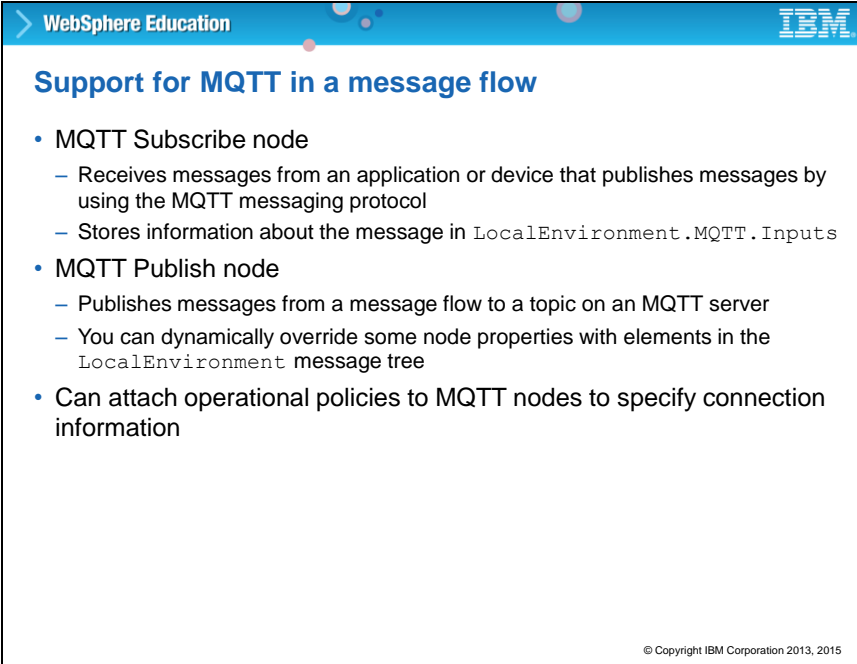
Use the `enabled` property for the `-o MQTTServer` object in the `-b pubsub` component to enable or disable the built-in MQTT broker. Use the `-n port` property to specify the port to be used by the broker. By default, the `enabled` property is set to `true` and the port is set to `11883`.

You can view the current configuration of the built-in MQTT broker by using the `mqsireportproperties` command.

When an integration node starts, the built-in MQTT broker starts on the port that MQTT server port property specifies. 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.

By default, Integration Bus publishes Administration and Operational events to the built-in MQTT broker. You can change this behavior by using the `mqsichangeproperties` command.



The slide is titled "Support for MQTT in a message flow" and is part of a WebSphere Education presentation. It features a blue header with the IBM logo. The content is organized into a bulleted list describing the MQTT Subscribe and Publish nodes, their dynamic configuration capabilities, and the ability to attach operational policies.

- MQTT Subscribe node
 - Receives messages from an application or device that publishes messages by using the MQTT messaging protocol
 - Stores information about the message in `LocalEnvironment.MQTT.Inputs`
- MQTT Publish node
 - Publishes messages from a message flow to a topic on an MQTT server
 - You can dynamically override some node properties with elements in the `LocalEnvironment` message tree
- Can attach operational policies to MQTT nodes to specify connection information

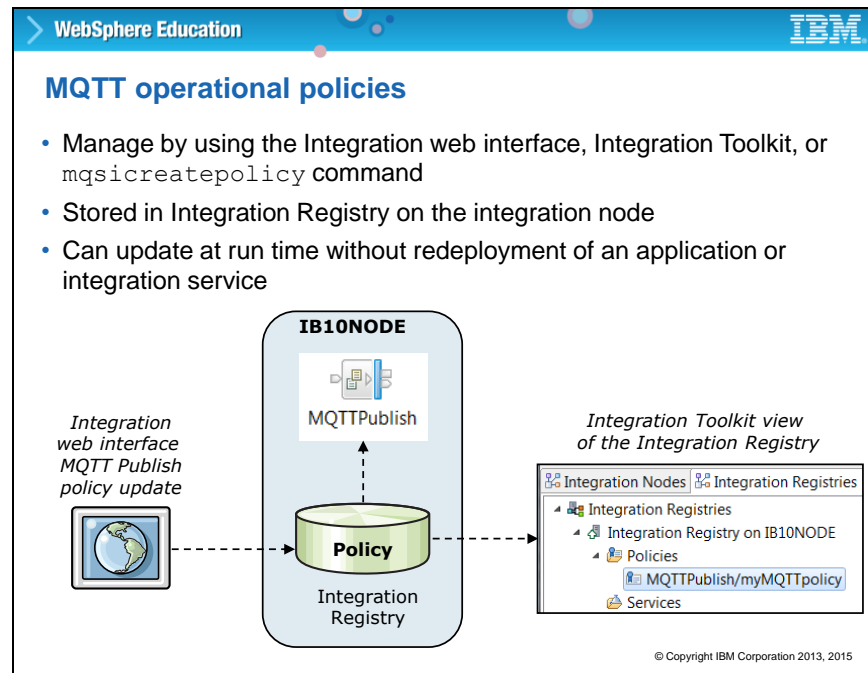
© Copyright IBM Corporation 2013, 2015

Support for MQTT in a message flow

You 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.

You 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.

Similar to MQEndpoint policies, you can define MQTT endpoint policies that contain the connection information to an MQTT server. You learn about MQTT policies next.



MQTT operational policies

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.

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.

Policies can be created and updated at any time in the solution lifecycle. During development, the developer can create the policies in the Integration Toolkit. Later in the cycle, the administrator can change an existing policy or create a policy.

Similar to MQEndpoint policies, the MQTT policies are stored in an integration node's Integration Registry.

WebSphere Education

IBM

Creating an MQTT policy in the Integration Toolkit

1. In the Message Flow editor, select the MQTT node that you want to use to generate a policy document.
2. In the **Properties** view for this node, click the **Policy** tab.
3. Click **Generate new policy** to open the Policy editor.
4. Edit the values of the operational properties as required.
5. Click **Save**, enter a policy name, select the target integration node, and click **Finish** to store the policy document in the Integration Registry.

*Policy Editor

Specify MQTT Subscribe node properties

Description

Specify description of MQTT Subscribe policy

Short description: myMQTTPolicy

Long description: MQTT Policy generated by the Toolkit

Basic

Specify basic set of properties for MQTT Subscribe

Client ID: FromMQTT.IN

Topic name: testTopic

Host name: localhost

Port: 1883

Quality of Service: 0 - At most once



© Copyright IBM Corporation 2013, 2015

Creating an MQTT policy in the Integration Toolkit

As shown here, you can define an MQTT policy in the Integration Toolkit.

1. In the Message Flow editor, select the MQTT node that you want to use to generate a policy document.
2. In the **Properties** view for this node, click the **Policy** tab.
3. Click **Generate new policy** to open the Policy editor.
4. Edit the values of the operational properties as required.
5. Click **Save**, enter a policy name, select the target integration node, and click **Finish** to store the policy document in the Integration Registry.

By default, the generated policy is attached to this message flow node at run time. To save the policy without attaching it to the node, clear the **Attach the generated policy to the node** check box.

 WebSphere Education 

Updating an attached MQTT policy

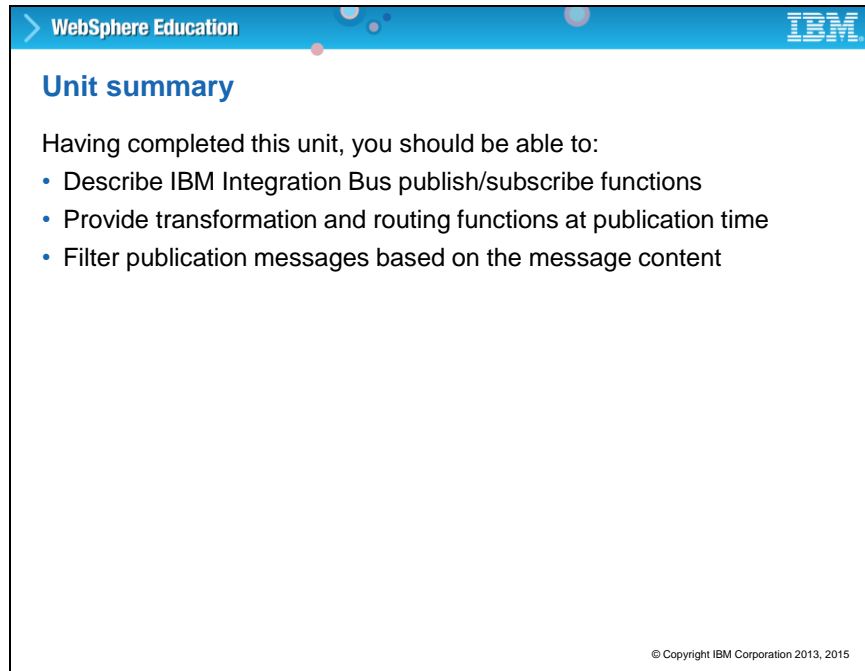
- Use the Integration web interface to retrieve and update a policy that is attached to a message flow node
 1. Expand **Servers > IntServer > Application**, where *IntServer* is the name of your integration server, and *Application* is where you stored your message flow.
 2. Expand **Message Flows**, and select the name of the message flow, or subflow to view
 3. Select 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


© Copyright IBM Corporation 2013, 2015

Updating an attached MQTT policy

The administrator or a developer with the permissions can use the message flow view in the Integration web user interface to retrieve and update a policy that is attached to a message flow node.

1. In the Integration web user interface navigation tree, expand **Servers > IntServer > Application**, where **IntServer** is the name of your integration server, and **Application** is where you deployed your message flow.
2. Expand Message Flows, and select the name of the message flow, or subflow, you want to view.
3. Select the **Operational Policy** tab from the top of the message flow pane.
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.

The slide features a blue header bar with the text 'WebSphere Education' on the left and the IBM logo on the right. The main content area is white and contains the title 'Unit summary' in blue. Below the title, it states 'Having completed this unit, you should be able to:' followed by a bulleted list of three items. At the bottom right, there is a small copyright notice.

> WebSphere Education 

Unit summary

Having completed this unit, you should be able to:

- Describe IBM Integration Bus publish/subscribe functions
- Provide transformation and routing functions at publication time
- Filter publication messages based on the message content

© Copyright IBM Corporation 2013, 2015

Unit summary

IBM Integration Bus uses publish/subscribe to notify applications of significant events that occur in integration nodes. IBM Integration Bus can also act as a content filtering provider for IBM MQ to allow subscribers to refer to elements in the body of publications. This unit described how IBM Integration Bus supports these publish/subscribe applications.

Having completed this unit, you should be able to:

- Describe IBM Integration Bus publish/subscribe functions
- Provide transformation and routing functions at publication time
- Filter publication messages based on the message content