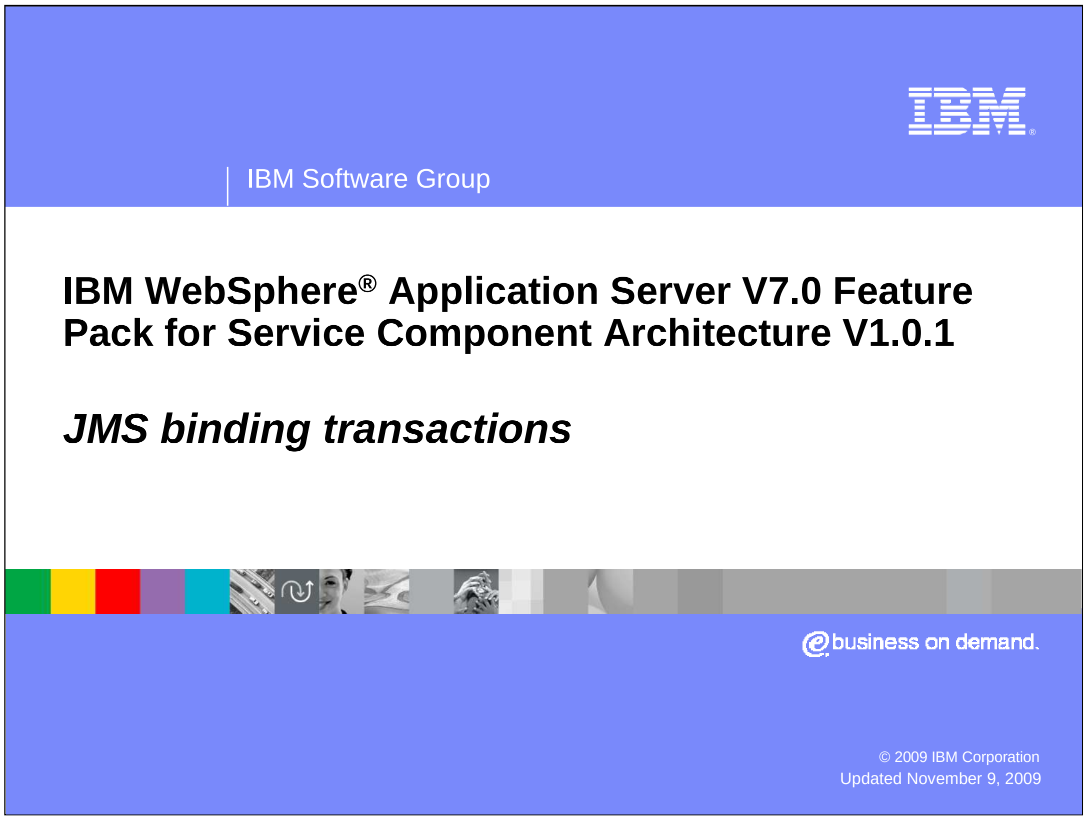
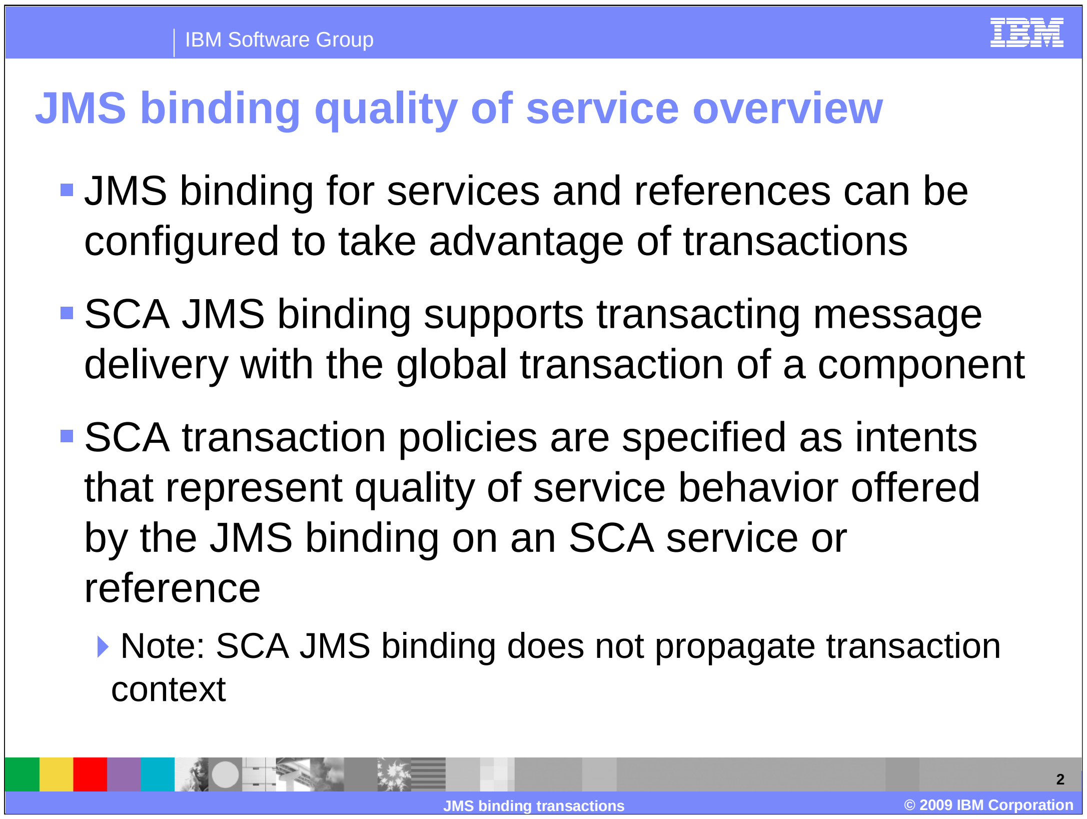


IBM Software Group

IBM WebSphere® Application Server V7.0 Feature Pack for Service Component Architecture V1.0.1

JMS binding transactions

This presentation will discuss JMS binding transactions.

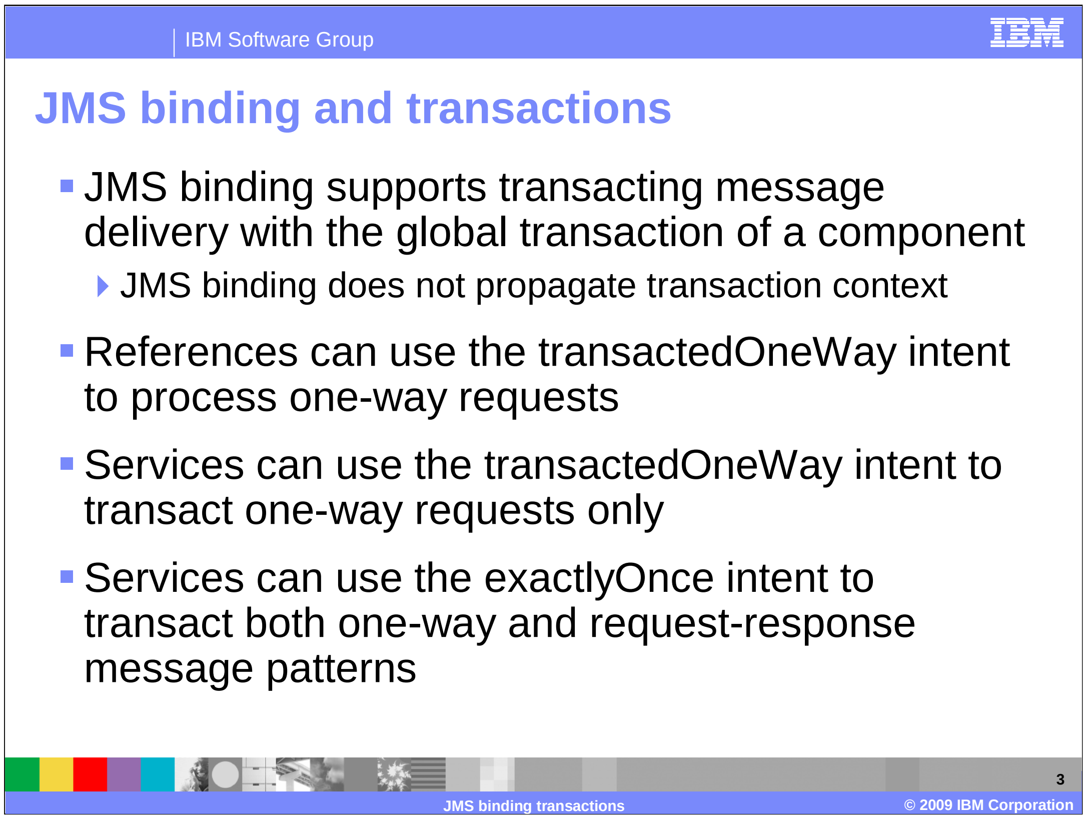


JMS binding quality of service overview

- JMS binding for services and references can be configured to take advantage of transactions
- SCA JMS binding supports transacting message delivery with the global transaction of a component
- SCA transaction policies are specified as intents that represent quality of service behavior offered by the JMS binding on an SCA service or reference
  - Note: SCA JMS binding does not propagate transaction context

You can configure the Service Component Architecture (SCA) Java™ Message Service (JMS) binding for services and references to take advantage of transaction quality of service behaviors.

The SCA transaction policies are specified as intents that represent quality of service behavior offered by the JMS binding on an SCA service or reference. However, note that the SCA JMS binding does not propagate transaction context; therefore, the client and service cannot participate in the same global transaction.



IBM Software Group

JMS binding and transactions

- JMS binding supports transacting message delivery with the global transaction of a component
  - JMS binding does not propagate transaction context
- References can use the transactedOneWay intent to process one-way requests
- Services can use the transactedOneWay intent to transact one-way requests only
- Services can use the exactlyOnce intent to transact both one-way and request-response message patterns

3

JMS binding transactions

© 2009 IBM Corporation

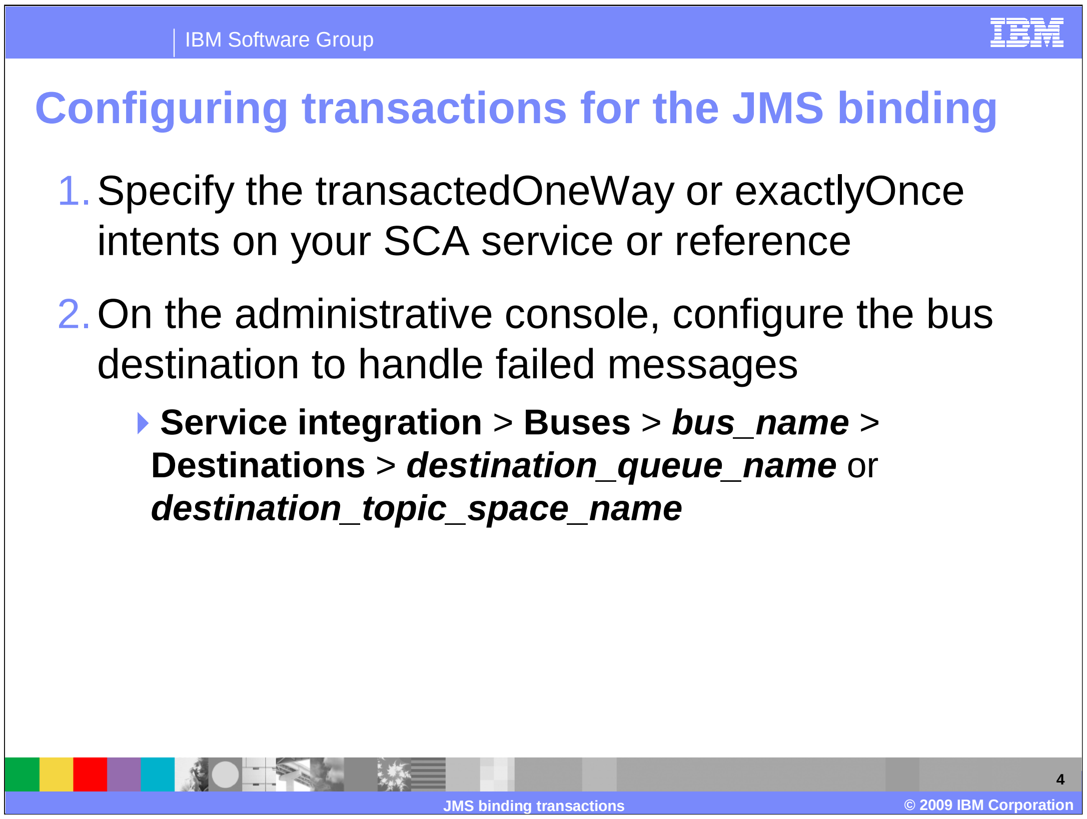
The SCA JMS binding supports transacting message delivery with the global transaction of a component as mentioned in the previous slide.

SCA references can use the transactedOneWay intent to transact one-way requests. Services can use the transactedOneWay intent to transact one-way requests only or the exactlyOnce intent to transact both one-way and request-response message patterns.

When the transactedOneWay intent is used on an SCA reference, a one-way request on the reference is not sent until the global transaction of the client is committed. If the global transaction of the client is rolled back, the request is not sent.

When the transactedOneWay intent is used on an SCA service, a one-way request is received from the JMS binding as part of the global transaction of the client. When the exactlyOnce intent is used on a service, both one-way and request-response message patterns are received from the JMS binding as part of the global transaction of the client. The receipt of the message and the sending of the response for request-response messaging, is not committed until the service transaction commits. If the service transaction is rolled back, the message is again made available for delivery or the message is sent to an exception destination that is based upon the configuration of the bus destination.

The SCA runtime environment typically performs a rollback of a global transaction only if the component produces an unchecked exception error. An unchecked exception error is a subclass of java.lang.RuntimeException or java.lang.Error class. A checked exception does not force a rollback. The component can force a rollback by using the UOWSynchronizationRegistry interface.



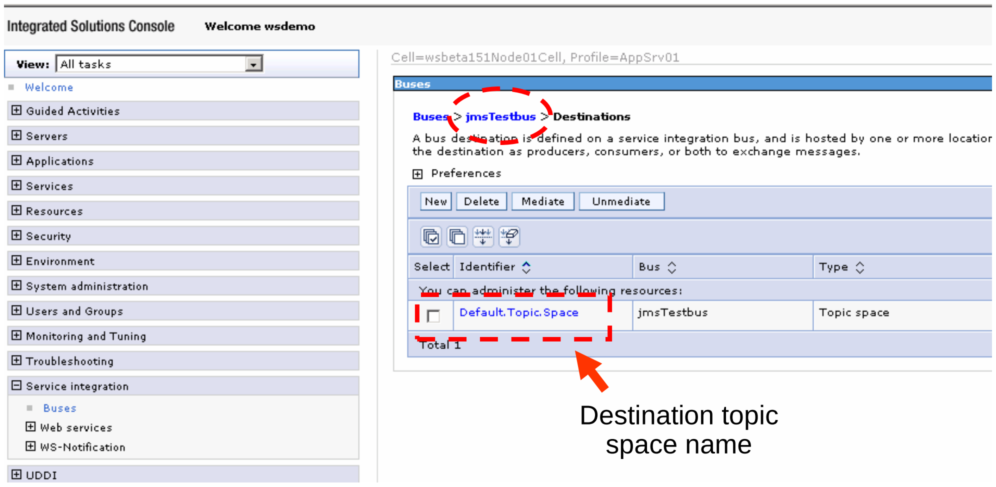
Configuring transactions for the JMS binding

1. Specify the transactedOneWay or exactlyOnce intents on your SCA service or reference
2. On the administrative console, configure the bus destination to handle failed messages
   - **Service integration** > **Buses** > ***bus_name*** > **Destinations** > ***destination_queue_name*** or ***destination_topic_space_name***

To configure transactions for the JMS binding, first, specify the transactedOneWay or exactlyOnce intents on your SCA service or reference enable transacting message delivery with the global transaction of your component. See the component example on the next slide.

Secondly, using the administrative console, configure the bus destination to handle failed messages, **Service integration** > **Buses** > ***bus_name*** > **Destinations** > ***destination_queue_name*** or ***destination_topic_space_name***. Reference the next slide to see the administrative console window

Destination topic space for failed messages



Cell=wsbeta151Node01Cell, Profile=AppSrv01

Buses

Buses > jmsTestbus > Destinations

A bus destination is defined on a service integration bus, and is hosted by one or more locatior the destination as producers, consumers, or both to exchange messages.

You can administer the following resources:

Select	Identifier	Bus	Type
☐	Default.Topic.Space	jmsTestbus	Topic space

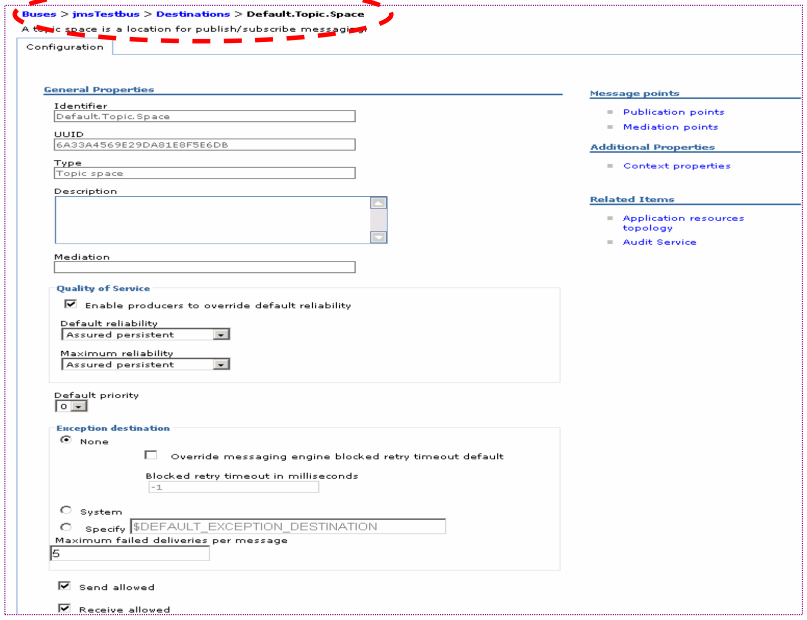
Total 1

On the administrative console, to configure the bus destination to handle failed messages, click

Service integration > **Buses** > ***bus_name*** > **Destinations** > ***destination_queue_name*** or ***destination_topic_space_name.***

Note that you can click the "**New**" button to create a new queue. For more information on JMS and administrative console, reference the JMS administration slides.

Configure failed messages



Buses > jmsTestbus > Destinations > Default.Topic.Space

A topic space is a location for publish/subscribe messaging.

Configuration

General Properties

Identifier: Default.Topic.Space

UUID: 6A33A4569E29DA81E8F5E6DB

Type: Topic space

Description:

Mediation:

Quality of Service

[✓] Enable producers to override default reliability

Default reliability: Assured persistent

Maximum reliability: Assured persistent

Default priority: 0

Exception destination

[✓ None]

[ ] Override messaging engine blocked retry timeout default

Blocked retry timeout in milliseconds: -1

System

Specify: $DEFAULT_EXCEPTION_DESTINATION

Maximum failed deliveries per message: 5

[✓] Send allowed

[✓] Receive allowed

Message points

- Publication points
- Mediation points

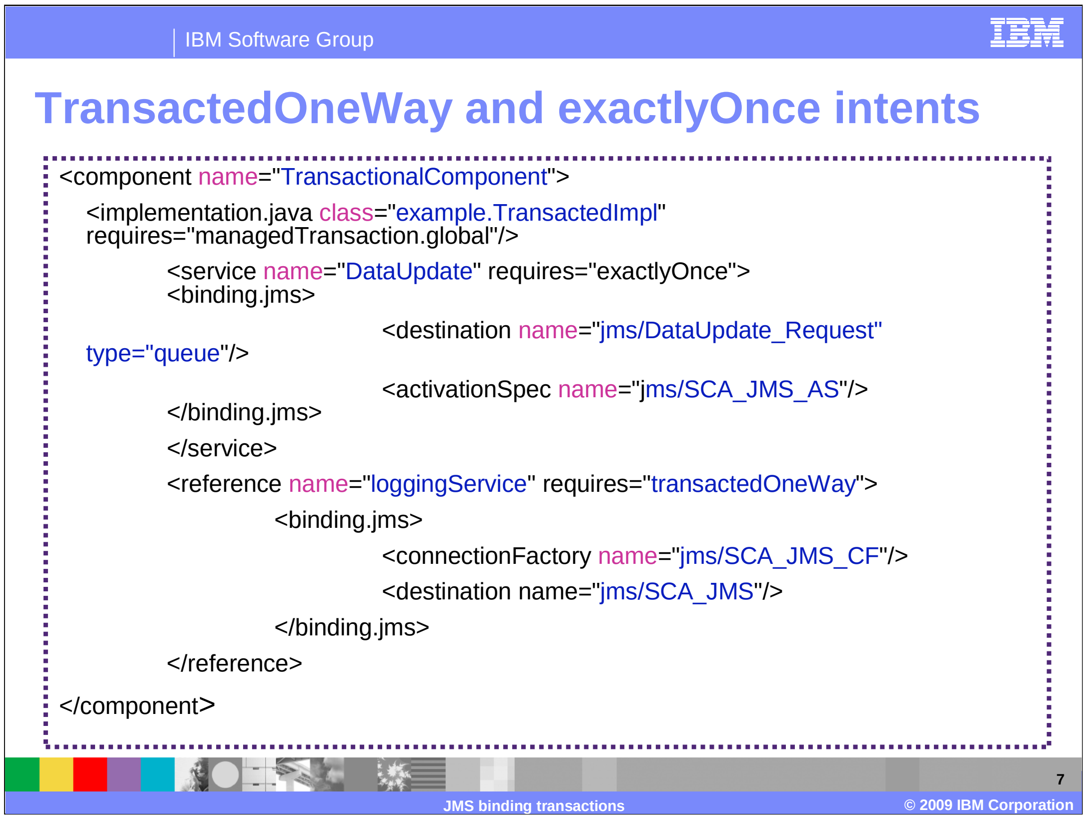
Additional Properties

- Context properties

Related Items

- Application resources topology
- Audit Service

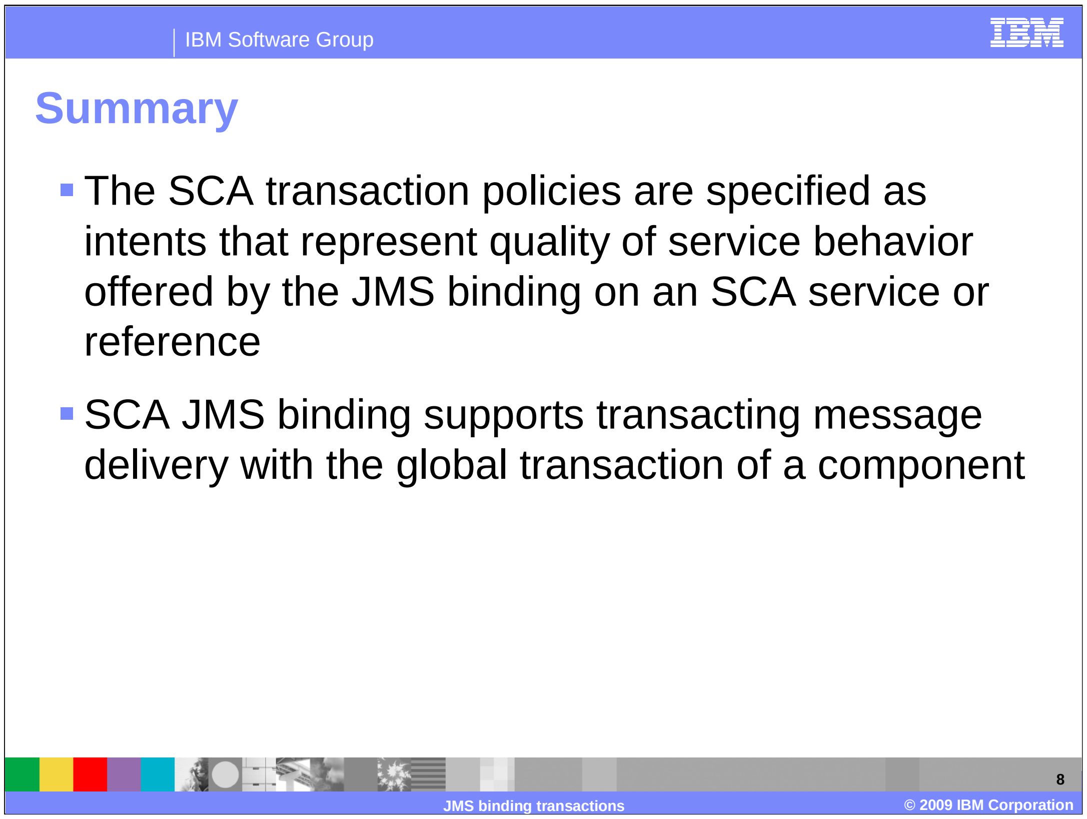
When you click the destination queue or topic, this configuration will come up. It’s from this window you configure how failed messages for the transactions are handled.



TransactedOneWay and exactlyOnce intents

```
<component name="TransactionalComponent">
    <implementation.java class="example.TransactedImpl"
    requires="managedTransaction.global"/>
            <service name="DataUpdate" requires="exactlyOnce">
            <binding.jms>
                                <destination name="jms/DataUpdate_Request"
    type="queue"/>
                                <activationSpec name="jms/SCA_JMS_AS"/>
            </binding.jms>
            </service>
            <reference name="loggingService" requires="transactedOneWay">
                    <binding.jms>
                                <connectionFactory name="jms/SCA_JMS_CF"/>
                                <destination name="jms/SCA_JMS"/>
                    </binding.jms>
            </reference>
</component>
```

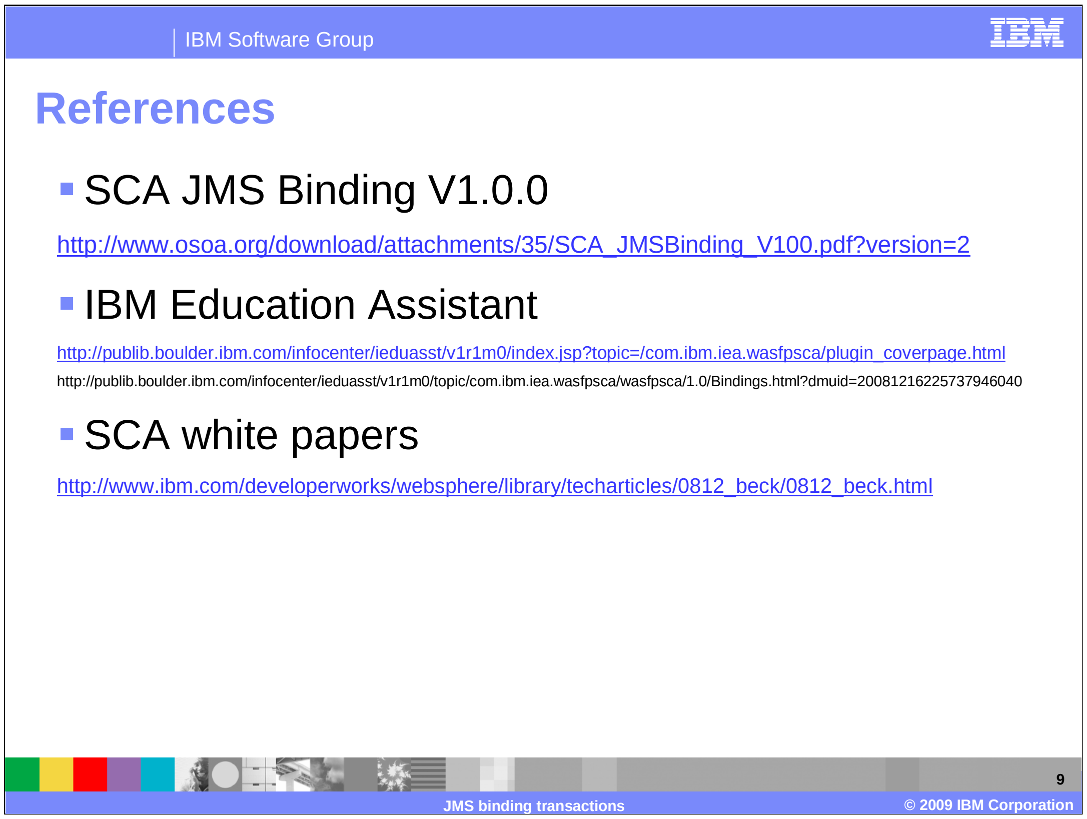
This example shows the use of the transactedOneWay and exactlyOnce intents. In this example, the component TransactionalComponent receives one-way and request-response messages from the DataUpdate service and subsequently sends one-way requests to the loggingService reference transactionally. If the component transaction rolls back, the active request is queued again and any requests to the reference are not sent. As mentioned earlier, note that the schema for binding.jms requires the destination element to appear before the connectionFactory element for the service



Summary

- The SCA transaction policies are specified as intents that represent quality of service behavior offered by the JMS binding on an SCA service or reference
- SCA JMS binding supports transacting message delivery with the global transaction of a component

The SCA transaction policies are specified as intents that represent quality of service behavior offered by the JMS binding on an SCA service or reference. SCA JMS binding supports transacting message delivery with the global transaction of a component.



IBM Software Group

References

- SCA JMS Binding V1.0.0

http://www.osoa.org/download/attachments/35/SCA_JMSBinding_V100.pdf?version=2

- IBM Education Assistant

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/plugin_coverpage.html

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wasfpsca/wasfpsca/1.0/Bindings.html?dmuid=20081216225737946040

- SCA white papers

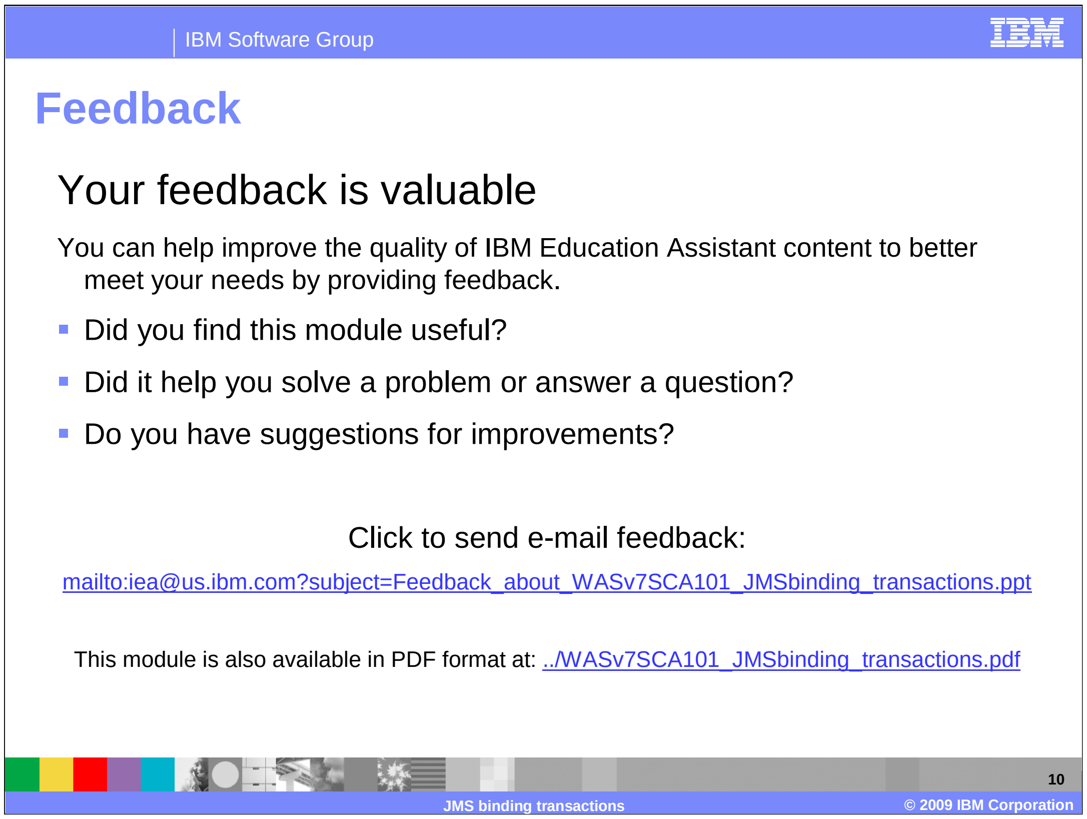
http://www.ibm.com/developerworks/websphere/library/techarticles/0812_beck/0812_beck.html

9

JMS binding transactions

© 2009 IBM Corporation

Here are some useful references



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

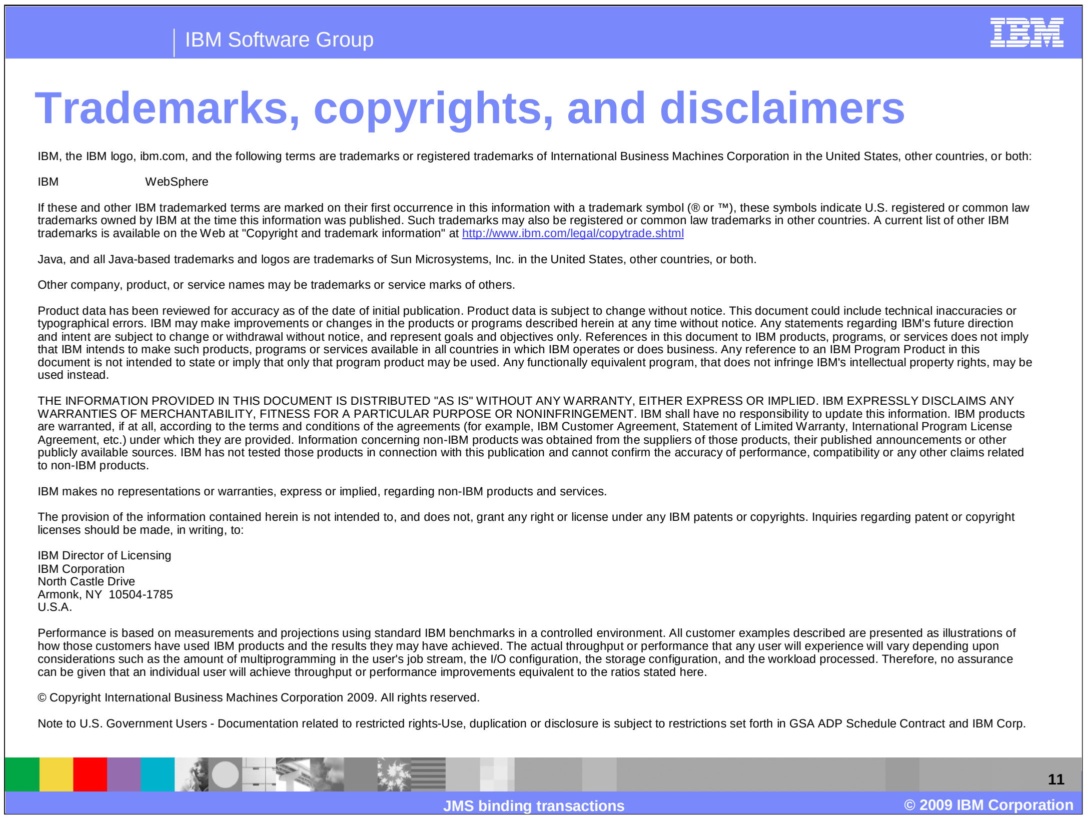
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASv7SCA101_JMSbinding_transactions.ppt

This module is also available in PDF format at: ../WASv7SCA101_JMSbinding_transactions.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers