JMS

On this Page:

Tagged Values

<<JMSAlias>>

Tagged Value	Description	Allowed Values / Example	
acknowledgeMode	Specify the message acknowledge mode. We recommend using acknowledge mode **transacted**.	Auto	Each single message sent to a JMS provider will be acknowledged by the JMS provider (not the recipient) after recipience. Messages received from a JMS provider within an activity are acknowledged irrespective of subsequent activities. Consequently, if an error occurs during the execution of the activity diagram after message receipt, no rollback occurs. Using Auto acknowledge mode in a model, a client must be prepared for possible loss of messages.
		Duplicate	**Duplicate** acknowledge mode corresponds to **Auto** acknowledge mode. Additionally, the JMS provider may send the message more than once to the same destination. The receiving application must be tolerant of receiving duplicate messages.
		Transacted	Messages sent to or received from a JMS provider within an activity are acknowledged explicitly after processing. Thus, the activity plays the role of a transactional lock. If the **acknowledgeMode** is specified as **Transacted** , a Bridge JMS client acknowledges a consumed message only after the activity diagram that implements the JMS adapter functionality completes without throwing an exception. This holds even if the activity diagram receives more then one message from, and/or sends messages to the queue during its execution.
timeToLive	Specify the expiration time of a sent message (refer also to the description of **JMSExpiration** in the JMS headers/properties).		
selector	Specify a selector to filter the received messages. Refer to the official Java Message Service Specification for the selector statement syntax.	e.g. `JMSType='alpha'`	
destinationName	Specify the name of the queue or topic.		
options	Specify a list of comma separated options in form of <name>=<value>. These options will be interpreted as native options. The available options depend on the JMS provider.	e.g. `queue.JMSDestination=aTestQueue`	

user	Specify the JMS credentials in form of <user> /<password>.	e.g. `system/manager`	
port	Specify the port the JMS provider is listening to.		
protocol	Protocol used to communicate with the JMS provider. The protocol normally is set automatically.		
jndiPath	Specify a path to the JNDI file (.bindings), if a JNDI provider is used and protocol is file.		
host	Specify the host name of the JMS provider.	**localhost**	Default.
		any other qualified host name	
isJNDIProvider	Specify whether the JMS provider is also a JNDI provider. Default is **false** if not specified.	true	The JMS provider is a JNDI provider.
		false	The JMS provider is not a JNDI provider (default).

<<JMSClient>>

Tagged Value	Description	Allowed Values / Example	
jmsConnectionFactory			
initialContextFactory	Specify the Java class that is used to access the LDAP service. This value is typically set automatically (depending on the JMS provider).	`org.apache.activemq.jndi.ActiveMQInitialContextFactory`	
connectionFactoryName	Specify the name of the JMS connection factory.		
Connection Pooling			
connectionTimeBetweenEvictionRunsMillis	Specify the time between two test cycles in milliseconds. A test cycle checks whether an open connection is idle for more than **connectionMinEvictableIdleTimeMillis** (see below).	-1	No test calls.
		540000	Default value, if not specified.
		0 or any other positive value	
connectionMinEvictableIdleTimeMillis	Specify the minimum idle time in milliseconds of an open connection before it can be removed from the connection pool. Relates to **connectionTimeBetweenEvictionRunsMillis** above.	**600000**	Default value, if not specified.
		0 or any other positive value	
connectionNumTestsPerEvictionRun	Specify the minimum number of open connections to be tested with every test run.	**3**	Default value, if not specified.
		0 or any other positive value	
connectionMaxIdle	Specify the maximum number of unused connections in the pool.	**5**	Default value, if not specified.
		0 or any other positive value	
connectionMinIdle	Specify the minimum number of open connections that will be hold available.	**0**	Default value, if not specified.
		any other positive value	
connectionMaxTotal	Specify the maximum number of available connections in the pool.	**-1**	No limit.
		0 or any other positive value	
connectionMaxWaitSec	Specify the maximum time in seconds to wait for an idle connection, if an adapter call is triggered. If no connection is available, an exception will be thrown.	**300**	Default value, if not specified.
		0 or any other positive value	
Session Pooling			
sessionMaxIdle	Specify the maximum number of unused open sessions in the pool.	**5**	Default value, if not specified.
		0 or any other positive value	

sessionNumTestsPerEvictionRun	Specify the time between two test cycles in milliseconds. A test cycle checks whether an open session is idle for more than **sessionMinEvictableIdleTimeMillis** (see below).	**3**	Default value, if not specified.
		0 or any other positive value	
sessionMinEvictableIdleTimeMillis	Specify the minimum idle time in milliseconds of an open session before it can be removed from the session pool. Relates to **sessionTimeBetweenEvictionRunsMillis** above.	**600000**	Default value, if not specified.
		0 or any other positive value	
sessionTimeBetweenEvictionRunsMillis	Specify the minimum number of open sessions to be tested with every test run.	**540000**	Default value, if not specified.
		0 or any other positive value	
sessionMaxTotal	Specify the maximum number of available sessions in the pool.	**-1**	Default value, if not specified.
		0 or any other positive value	
sessionMinIdle	Specify the minimum number of open sessions that will be hold available.	**0**	Default value, if not specified.
		any other positive value	
sessionWaitSec	Specify the maximum time in seconds to wait for an idle session, if an adapter call is triggered. If no session is available, an exception will be thrown.	**300**	Default value, if not specified.
		0 or any other positive value	

JMS Adapter Parameters

Action "receive"

Name	Type	Direction	Mandatory	Description
connectionInfo	JMSConnectionInfo	in		Specify the JMS connection details. See type JMSConnectionInfo for more details.
receiveParameter	JMSReceiveParameter	in		Specify the **receive** action details. See type JMSReceiveParameter for more details.
messageString	String	out	(✓)	This parameter returns the JMS message content, if it contains a String. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
jmsStringMessage	JMSStringMessage	out	(✓)	This parameter returns the JMS message including all message properties, if it is a String message. See type JMSStringMessage for more details. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
messageBlob	Blob	out	(✓)	This parameter returns the JMS message, if it contains a Blob. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
jmsBlobMessage	JMSBlobMessage	out	(✓)	This parameter returns the JMS message including all message properties, if it is a Blob message. See type JMSBlobMessage for more details. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.

Action "send"

Name	Type	Direction	Mandatory	Description
connectionInfo	JMSConnectionInfo	in		Specify the JMS connection details. See type JMSConnectionInfo for more details.
sendParameter	JMSSendParameter	in		Specify the **send** action details. See type JMSSendParameter for more details.
messageString	String	in	(✓)	This parameter contains the JMS message to be send, if it contains a String. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
jmsStringMessage	JMSStringMessage	in	(✓)	This parameter contains the JMS message to be send including all message properties, if it is a String message. See type JMSStringMessage for more details. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
messageBlob	Blob	in	(✓)	This parameter contains the JMS message to be send, if it contains a Blob. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.
jmsBlobMessage	JMSBlobMessage	in	(✓)	This parameter contains the JMS message to be send including all message properties, if it is a Blob message. See type JMSBlobMessage for more details. One of **messageString**, **messageBlob**, **jmsStringMessage** or **jmsBlobMessage** must be specified.

JMS Adapter Header Fields/JMS Properties

JMS Standard

The JMS Message parameter contains the following headers/JMS properties:

Name	Type	Description	Allowed Values	
JMSCorrelationID	String	A client can use this header field to link one message with another. A typical use is to link a response message with its request message. You can override **JMSCorrelationID** with a custom value.		
JMSPersistentDeliveryMode	Boolean	This header field contains the delivery mode specified when the message is sent.	**true**	The JMS provider has to take extra care to ensure that the message is not lost in transit due to a JMS provider failure (default). The JMS provider must deliver the message *once and only once*.
			false	The JMS provider may lose the message due to a JMS provider failure.
JMSDestination	String	This header field contains the destination, to which the message is sent.		
JMSExpiration	Integer	When a message is sent, its expiration time is calculated as the sum of the time-to-live value specified on the send method and the current GMT value. On return from the send method, the **JMSExpiration** header field of the message contains this value. When a message is received, its **JMSExpiration** header field contains the same value.	0	The JMS does not expire.
			>0	The JMS message expires after the given period of time.

JMSMessageID	String	This header field contains a value that uniquely identifies each message sent by a provider. A **JMSMessageID** is a string value, which should function as a unique key for identifying messages in a historical repository. A provider defines the exact scope of uniqueness. Combined with **JMSCorrelationID** , this header field is used to link messages, and thus to implement synchronous messaging.		
JMSPriority	Integer	This header field contains the message's priority. JMS defines a ten-level priority value, with 0 as the lowest priority and 9 as the highest.	0-**4**	The JMS message has a *normal* priority (**4** is the default value).
			5-9	The JMS message has an *expedited* priority.
JMSRedelivered	Boolean	If a client receives a message with the **JMSRedelivered** indicator set to true, it is likely, but not guaranteed, that this message was delivered, but not acknowledged, in the past. In general, a provider must set this message header field to true whenever he redelivers a message.	true	The JMS message has been redelivered. This is an indication to the consuming application that the message may have been delivered in the past and that the application should take extra precautions to prevent duplicate processing.
			false	The JMS message has not been redelivered.
JMSReplyTo	String	This header field contains the name of a destination, to which the recipient of the message should send a reply. The client sending the message specifies the **JMSReplyTo** field value.		
JMSTimestamp	Integer	This header field contains the time a client passes the message to a provider to be sent. The time is not the actual transmission time of the message; the actual send may occur later due to transactions or other client side queuing of messages.		
JMSType	String	This header field contains a message type identifier supplied by a client when the message is sent. Some messaging systems require that a message type definition is created for each application message and that each message specifies its type. In order to work with such JMS providers, JMS clients should assign a value to **JMSType** whether the application makes use of it or not. The assignment of a value ensures that the field is set properly for those providers that require a **JMSType** value.		
JMSCorrelationIDAsBytes	Blob	Contains the **JMSCorrelationID** in Blob format.		

Custom

The JMS Message parameter contains arrays to define custom headers/JMS properties as name/value pairs (see JMSMessage below).
The following custom headers are provided automatically by the xUML Runtime:

Name	Type	Description	Allowed Values
xSenderHost	String	This custom JMS header field contains the sender host.	
xSenderService	String	This custom JMS header field contains the sender service.	
xTransactionId	String	This custom JMS header field contains a unique transaction ID (see Transaction ID).	

JMS Adapter Parameter Types

JMSConnectionInfo

Class	Attribute	Type	Description	Allowed Values	
JMSConnectionInfo	**acknowledgeMode**	JMSAcknowledgeModes	Message acknowledge mode. We recommend using acknowledge mode **transacted**.	Auto	Each single message sent to a JMS provider will be acknowledged by the JMS provider (not the recipient) after recipience. Messages received from a JMS provider within an activity are acknowledged irrespective of subsequent activities. Consequently, if an error occurs during the execution of the activity diagram after message receipt, no rollback occurs. Using Auto acknowledge mode in a model, a client must be prepared for possible loss of messages.
				Duplicate	**Duplicate** acknowledge mode corresponds to **Auto** acknowledge mode. Additionally, the JMS provider may send the message more than once to the same destination. The receiving application must be tolerant of receiving duplicate messages.

				Transacted	Messages sent to or received from a JMS provider within an activity are acknowledged explicitly after processing. Thus, the activity plays the role of a transactional lock. If the **acknowledgeMode** is specified as **Transacted** , a Bridge JMS client acknowledges a consumed message only after the activity diagram that implements the JMS adapter functionality completes without throwing an exception. This holds even if the activity diagram receives more then one message from, and/or sends messages to the queue during its execution.
	name	String	Arbitrary name of the JMS session to distinct multiple sessions.		
	user	String	JMS user.		
	password	String	JMS user password.		
	path	String	Specify a path to the JNDI file (. bindings), if a JNDI provider is used and protocol is file.		
	protocol	String	Protocol used to communicate with the JMS provider. The protocol normally is set automatically.		
	host	String	Specify the host name of the JMS provider.		
	port	Integer	Specify the port the JMS provider is listening to.		
	properties	Array of JMSStringProperty	Array of key/value pairs used to specify additional JMS Provider properties (e.g. the message queue using ActiveMQ).		

JMSMessage

Class	Attribute	Type	Description	Allowed Values
JMSMessage	**longProperties**	JMSLongProperty	JMS properties.	
	byteProperties	JMSByteProperty	JMS properties.	
	booleanProperties	JMSBooleanProperty	JMS properties.	
	doubleProperties	JMSDoubleProperty	JMS properties.	
	stringProperties	JMSStringProperty	JMS properties.	
	integerProperties	JMSIntegerProperty	JMS properties.	
	floatProperties	JMSFloatProperty	JMS properties.	
	shortProperties	JMSShortProperty	JMS properties.	
	JMSCorrelationID	String	A client can use this header field to link one message with another. A typical use is to link a response message with its request message. You can override **JMSCorrelationID** with a custom value.	

JMSPersistentDeliveryMode	Boolean	This header field contains the delivery mode specified when the message is sent.	**true**	The JMS provider has to take extra care to ensure that the message is not lost in transit due to a JMS provider failure (default). The JMS provider must deliver the message *once and only once*.
			false	The JMS provider may lose the message due to a JMS provider failure.
JMSDestination	String	This header field contains the destination, to which the message is sent.		
JMSExpiration	Integer	When a message is sent, its expiration time is calculated as the sum of the time-to-live value specified on the send method and the current GMT value. On return from the send method, the **JMSExpiration** header field of the message contains this value. When a message is received, its **JMSExpiration** header field contains the same value.	0	The JMS does not expire.
			>0	The JMS message expires after the given period of time.
JMSMessageID	String	This header field contains a value that uniquely identifies each message sent by a provider. A **JMSMessageID** is a string value, which should function as a unique key for identifying messages in a historical repository. A provider defines the exact scope of uniqueness. Combined with **JMSCorrelationID** , this header field is used to link messages, and thus to implement synchronous messaging.		
JMSPriority	Integer	This header field contains the message's priority. JMS defines a ten-level priority value, with 0 as the lowest priority and 9 as the highest.	0-**4**	The JMS message has a *normal* priority (**4** is the default value).
			5-9	The JMS message has an *expedited* priority.
JMSRedelivered	Boolean	If a client receives a message with the **JMSRedelivered** indicator set to true, it is likely, but not guaranteed, that this message was delivered, but not acknowledged, in the past. In general, a provider must set this message header field to true whenever he redelivers a message.	true	The JMS message has been redelivered. This is an indication to the consuming application that the message may have been delivered in the past and that the application should take extra precautions to prevent duplicate processing.
			false	The JMS message has not been redelivered.
JMSReplyTo	String	This header field contains the name of a destination, to which the recipient of the message should send a reply. The client sending the message specifies the **JMSReplyTo** field value.		
JMSTimestamp	Integer	This header field contains the time a client passes the message to a provider to be sent. The time is not the actual transmission time of the message; the actual send may occur later due to transactions or other client side queuing of messages.		
JMSType	String	This header field contains a message type identifier supplied by a client when the message is sent. Some messaging systems require that a message type definition is created for each application message and that each message specifies its type. In order to work with such JMS providers, JMS clients should assign a value to **JMSType** whether the application makes use of it or not. The assignment of a value ensures that the field is set properly for those providers that require a **JMSType** value.		
JMSCorrelationIDAsBytes	Blob	**JMSCorrelationID** as a Blob.		

JMSBlobMessage

Class	Attribute	Type	Description	Allowed Values
JMSBlobMessage	Possible type of a JMS message body. Derives from JMSMessage.			
	message	Blob	Blob message content.	

JMSStringMessage

Class	Attribute	Type	Description	Allowed Values
JMSStringMessage	Possible type of a JMS message body. Derives from JMSMessage.			

	message	String	String message content.	

JMSReceiveParameter

Class	Attribute	Type	Description	Allowed Values
JMSReceiveParameter	**queueName**	String	Specifies the name of the queue/topic, from which messages are received and is mandatory in order to receive messages	
	selector	String	Used to filter the received messages (e.g. JMSType='alpha'). Refer to the official Java Message Service specification for the selector statement syntax.	
	millisecondsToWait	Integer	Specifies the time the JMS adapter is waiting for a message from the specified queue/topic	

JMSSendParameter

Class	Attribute	Type	Description	Allowed Values
JMSSendParameter	**queueName**	String	Specifies the name of the queue/topic, to which messages are sent.	
	timeToLive	Integer	Specifies the expiration time of a sent message (refer also to the description of **JMSExpiration** in the JMS headers/properties).	

JMSSessionParameter

Class	Attribute	Type	Description	Allowed Values	
JMSSessionParameter	**acknowledgeMode**	JMSAcknowledgeModes	Message acknowledge mode. We recommend using acknowledge mode **transacted**.	Auto	Each single message sent to a JMS provider will be acknowledged by the JMS provider (not the recipient) after recipience. Messages received from a JMS provider within an activity are acknowledged irrespective of subsequent activities. Consequently, if an error occurs during the execution of the activity diagram after message receipt, no rollback occurs. Using Auto acknowledge mode in a model, a client must be prepared for possible loss of messages.
				Duplicate	**Duplicate** acknowledge mode corresponds to **Auto** acknowledge mode. Additionally, the JMS provider may send the message more than once to the same destination. The receiving application must be tolerant of receiving duplicate messages.
				Transacted	Messages sent to or received from a JMS provider within an activity are acknowledged explicitly after processing. Thus, the activity plays the role of a transactional lock. If the **acknowledgeMode** is specified as **Transacted** , a Bridge JMS client acknowledges a consumed message only after the activity diagram that implements the JMS adapter functionality completes without throwing an exception. This holds even if the activity diagram receives more then one message from, and/or sends messages to the queue during its execution.
	name	String	Arbitrary name of the JMS session to distinct multiple sessions.		
	connectionFactoryClass	String	Name of the connection factory class.		
	connectionFactoryProperties	JMSStringProperty	Properties of the connection factory.		
	user	String	JMS user.		
	password	String	JMS user password.		

Custom JMS Properties

Class	Attribute	Type	Description	Allowed Values

JMSBooleanProperty	**key**	String	Key of the JMS property.	
	value	Boolean	Boolean property value.	
JMSByteProperty JMSIntegerProperty JMSLongProperty JMSShortProperty	**key**	String	Key of the JMS property.	
	value	Integer	Integer property value.	
JMSDoubleProperty JMSFloatProperty	**key**	String	Key of the JMS property.	
	value	Float	Float property value.	
JMSStringProperty	**key**	String	Key of the JMS property.	
	value	String	String property value.	

JMSAcknowledgeModes

Enumeration	Type	Value	Description
JMSAcknowledgeModes	String	Auto	Each single message sent to a JMS provider will be acknowledged by the JMS provider (not the recipient) after recipience. Messages received from a JMS provider within an activity are acknowledged irrespective of subsequent activities. Consequently, if an error occurs during the execution of the activity diagram after message receipt, no rollback occurs. Using Auto acknowledge mode in a model, a client must be prepared for possible loss of messages.
		Duplicate	**Duplicate** acknowledge mode corresponds to **Auto** acknowledge mode. Additionally, the JMS provider may send the message more than once to the same destination. The receiving application must be tolerant of receiving duplicate messages.
		Transacted	Messages sent to or received from a JMS provider within an activity are acknowledged explicitly after processing. Thus, the activity plays the role of a transactional lock. If the **acknowledgeMode** is specified as **Transacted** , a Bridge JMS client acknowledges a consumed message only after the activity diagram that implements the JMS adapter functionality completes without throwing an exception. This holds even if the activity diagram receives more then one message from, and/or sends messages to the queue during its execution.