

JMS Parameters in Detail

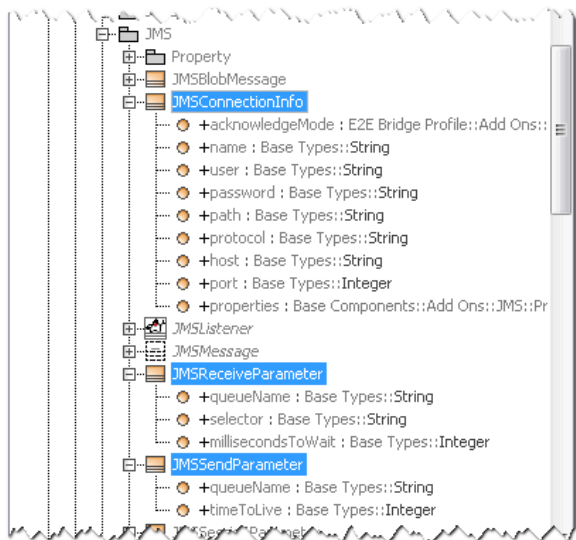
The **JMSSessionParameter** holds the connection information.

After connecting to a message queue or topic, three operations may be executed on the queue or topic: Sending a message, receiving a message, or listening to messages. The sending and receiving of messages is mapped onto JMS adapter action nodes with a specific action (send or receive) and specific parameters (**JMSSendParameter** and **JMSReceiveParameter**). The listening to a topic is implemented by listener operations, which are part of the central class JMSListener (stereotype <<E2EJMSListener>>) in the Bridge template (**processJMSStringMessage** and **processStringMessage**).

For how to draw sending and receiving JMS messages, see [Modeling the Sending and Receiving of JMS Messages](#).

The JMS adapter provides three types of parameters for sending and receiving JMS messages (see figure below).

Figure: JMS Parameter



Parameter Name	Description
JMSReceiveParameter	This parameter specifies the name of the queue, through which the JMS adapter receives messages, a selector statement as a String value as may be necessary and the millisecondsToWait attribute value. (For more details see chapter JMS Receive Parameter .)
JMSSendParameter	This parameter specifies the name of the queue, through which the JMS adapter sends messages and the time-to-live attribute value. (For more details see chapter JMS Send Parameter .)
JMSConnectionInfo	This parameter specifies information concerning the connection to the JMS provider such as host, port, protocol, ... (For more details see chapter JMS Connection Info .)

To send a message via the JMS adapter, values for **JMSConnectionInfo**, **JMSSendParameter**, and the message content must be set. To receive a message via the JMS adapter, values for **JMSConnectionInfo** and **JMSReceiveParameter** must be set.

JMSConnectionInfo values may be specified in the component diagram of the UML model on a dependency of an JMS adapter (refer to [Defining the Components](#)). However, in doing so, the connection to a JMS provider's queue is static and the **JMSConnectionInfo** values are applicable to the whole service.

Alternatively, the connection to a JMS provider's queue can be dynamic by specifying the necessary parameters each time the service sends a message to, or receives a message from, the JMS provider's queue. For how to draw sending and receiving JMS messages dynamically, see [Dynamic Sending and Receiving](#).

On this Page:

- [JMS Connection Info](#)
 - [acknowledgeMode](#)
 - [Other JMS Connection Info Parameters](#)
 - [Example: Specifying the parameter using static access to the JMS provider](#)
 - [Example: Specifying the parameter using dynamic access to the JMS provider](#)
- [JMS Send Parameter](#)
 - [Example: Specifying the parameter using static access to the JMS provider](#)
 - [Example: Specifying the parameter using dynamic access to the JMS provider](#)
- [JMS Receive Parameter](#)
 - [Example: Specifying the parameter using static access to the JMS provider](#)
 - [Example: Specifying the parameter using dynamic access to the JMS provider](#)

JMS Connection Info

Figure: Attributes of the JMS Connection Info

JMSConnectionInfo
+acknowledgeMode : JMSAcknowledgeModes
+host : String
+name : String
+password : String
+path : String
+port : Integer
+properties : JMSStringProperty [0..*]
+protocol : String
+user : String

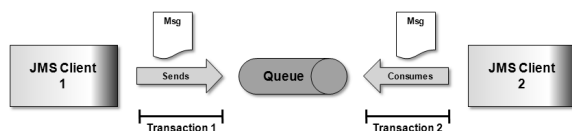
The JMS connection info defines the connection parameters of the connection to the JMS Provider. Implementing static JMS, these parameters are supplied via the component diagram. Implementing dynamic JMS, these parameters may be supplied by the component diagram as well (as a default), but may also be overwritten afterwards by action script.

acknowledgeMode

Value	Description
Auto	Each single message sent to a JMS provider will be acknowledged by the JMS provider (not the recipient) after receipt. 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. We recommend to use Transacted acknowledge mode.

Provided that 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 than one message from, and/or sends messages to the queue during its execution.

Figure: Transaction Handling



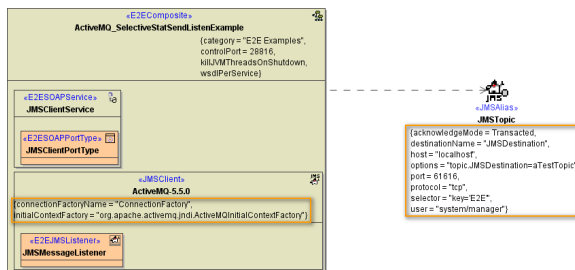
Other JMS Connection Info Parameters

Name	Description
host	holds the name of the host, on which JMS provider service is running

name	holds an arbitrary name of the JMS session to distinct multiple sessions
path	holds the path to the JNDI properties file (.bindings), if the JMS Provider is accessed via a JNDI with file binding
port	holds the port number the JMS Provider is listening on
properties	holds an array of key/value pairs used to specify additional JMS Provider properties (e. g. the message queue using ActiveMQ)
protocol	holds the protocol used to communicate with the JMS Provider
user and password	hold the authentication information

Example: Specifying the parameter using static access to the JMS provider

Figure: Connection Info in Component Diagram



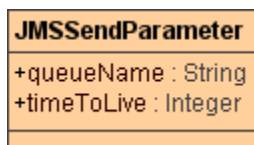
Example: Specifying the parameter using dynamic access to the JMS provider

```
create aConnectionInfo;
set aConnectionInfo.acknowledgeMode = cast("Transacted");
set aConnectionInfo.host = "oracle.e2e.ch";
set aConnectionInfo.port = 7001;
set aConnectionInfo.protocol = "t3";
set aConnectionInfo.user = "weblogic";
set aConnectionInfo.password = "password";
```

In the dynamic case the parameter needs not to be specified in the component diagram. However, component diagram definitions will serve as a default to the connection info parameter and may be overwritten afterwards by action script.

JMS Send Parameter

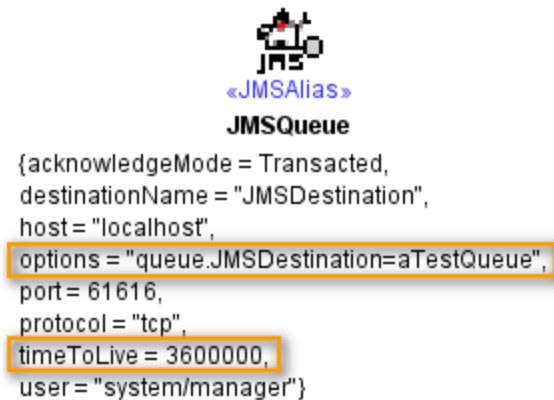
Figure: Attributes of the JMS Send Parameter



Name	Description
queueName	specifies the name of the queue/topic, to which messages are sent
timeToLive	specifies the expiration time of a sent message (refer also to the description of JMSExpiration in JMS Message Header Fields)

Example: Specifying the parameter using static access to the JMS provider

Figure: Send Parameter in Component Diagram



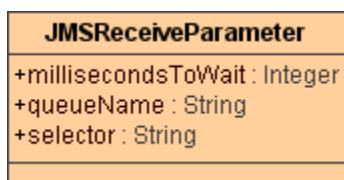
Example: Specifying the parameter using dynamic access to the JMS provider

```
create aJMSSendParameter;  
set aJMSSendParameter.queueName = "aTestQueue";  
set aJMSSendParameter.timeToLive = 0;
```

In the dynamic case the parameter needs not to be specified in the component diagram. However, component diagram definitions will serve as a default to the send parameter and may be overwritten afterwards by action script.

JMS Receive Parameter

Figure: Attributes of the JMS Receive Parameter



Name	Description	Allowed Values	
queueName	specifies the name of the queue/topic, from which messages are received and is mandatory in order to receive messages		
selector	filters the received messages (e.g. JMSType='alpha') Refer to the official Java Message Service Specification for the selector statement syntax.		
millisecondsToWait	specifies the time the JMS adapter is waiting for a message from the specified queue/topic	0	The JMS adapter waits as long as getting the message takes.
		-1	The JMS adapter does not wait for a message. If there is one, the adapter receives it - otherwise the process steps forward.
		>0	The JMS adapter waits for a (specific) message as long as indicated by the positive number.

If the parameter `millisecondsToWait` is not explicitly set, the default value is 1000 milliseconds. The default is also valid in the static case and can not be changed in the component diagram.

Example: Specifying the parameter using static access to the JMS provider

Figure: Receive Parameter in Component Diagram



```
{acknowledgeMode = Transacted,  
destinationName = "JMSDestination",  
host = "localhost",  
options = "topic.JMSDestination=aTestTopic",  
port = 61616,  
protocol = "tcp",  
selector = "key='E2E'",  
timeToLive = 3600000,  
user = "system/manager"}
```

Example: Specifying the parameter using dynamic access to the JMS provider

```
create aJMSReceiveParameter;  
set aJMSReceiveParameter.queueName = "aTestQueue";  
set aJMSReceiveParameter.selector = "key='E2E'";  
set aJMSReceiveParameter.millisecondsToWait = 0;
```

In the dynamic case the parameter needs not to be specified in the component diagram. However, component diagram definitions will serve as a default to the receive parameter and may be overwritten afterwards by action script.