Listening on a JMS Topic

Listening

Example Files (Builder project Add-ons/JMS):

<your example path>\Add-ons\JMS\uml\simpleStatSendListen.xml

The central class JMSListener with stereotype <<E2EJMSListener>> is part of the Base Components of the Bridge template. This class provides four operations to listen to a topic depending on the message type and offering additional information supply.

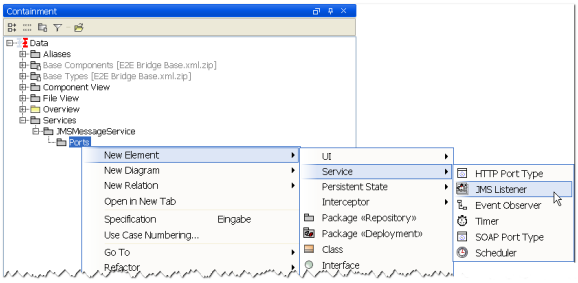
Figure: JMS Listener Class as Part of the Base Components

Operation	Description
processJMSStringMessage	listening to messages of type string and suppling the activity with additional header information. For more details on the JMS Header see chapter JMS Message Header Fields.
processStringMessage	listening to messages of type string
processJMSBlobMessage	listening to messages of type blob and suppling the activity with additional header information. For more details on the JMS Header see chapter JMS Message Header Fields.
processBlobMessage	listening to messages of type blob

If required, the JMS adapter converts the content of a message body of type String into an object of type Blob and vice versa.

Exemplarily, use the operation processStringMessage to listen to the JMS topic. Create a new JMS Listener Port Type (e.g. JMSMessageListener).

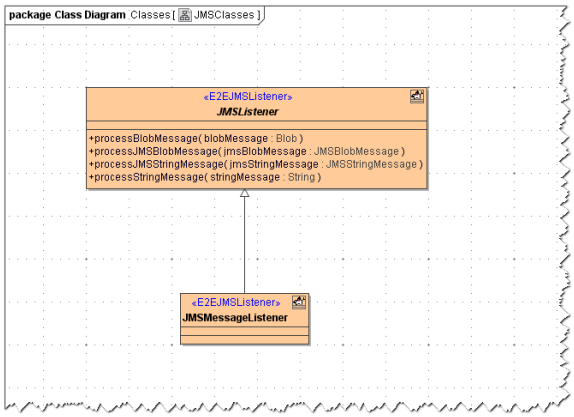
Figure: Creating a Listener Port Type



Create a class diagram and drag and drop both - the new port type JMSMessageListener and the central class JMSListener onto the diagram pane. Draw a Generalization from JMSMessageListener to JMSListe ner, with the result that JMSMessageListener inherits all operations from JMSListener.

Figure: JMSListener as Generalization of the Port Type Class

On this Page:

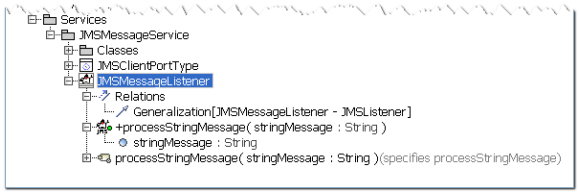


JMSListener contains four callback methods, from which one can be registered.

Create the listener port type operation processStringMessage in class JMSMessageListener to overwrite the operation with your own actions. You can copy it from the abstract definitions in the Base Components and change it as to be static.

The figure below illustrates the corresponding containment tree.

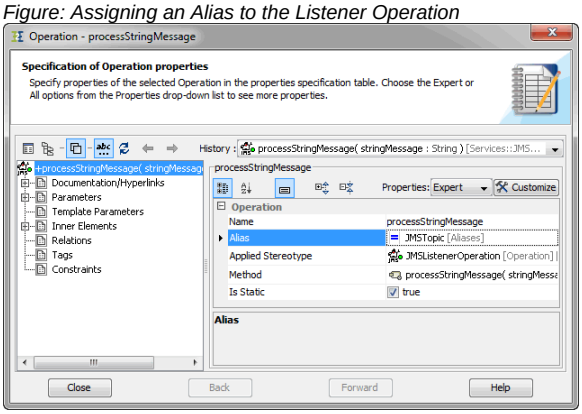
Figure: Listener Operation in Containment Tree



A component artifact in the component diagram allows one and only one dependency to a JMS provider's topic. Because of this constraint, the modeler has to decide, which listener method should be used. It is not possible to use all listener methods within one xUML service.

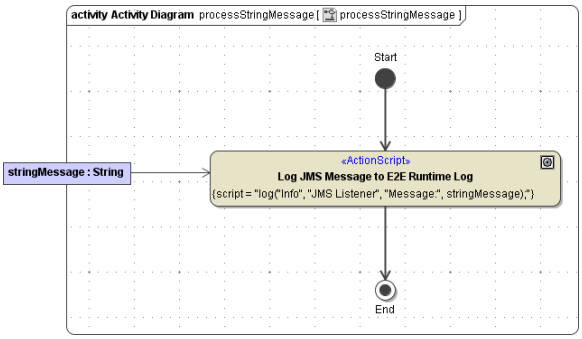
Assign an alias to the listener operation processStringMessage to specify the topic the listener is listening on. The component wizard helps with defining the JMS alias. So if you have not defined an alias yet, keep in mind to assign the alias to the listener operation later.

Figure: Assigning an Alias to the Listener Operation



In the next step, assign an activity diagram to the listener operation processStringMessage. The figure below illustrates the basic structure of such an activity diagram. In the example each received message is written to the Bridge log.

Figure: Activity Diagram Listening to a JMS String Message



The listener processStringMessage listens on all string messages of the defined topic. Wanting to listen to selected messages only, refer to Listening with a Selector.

Listeners maintain their own JMS session. They do not share their session with other JMS operations (send and receive). If the connection to the JMS provider gets lost, the xUML Runtime will try to reconnect the listener every second. The following log entries will be created in the Bridge log (for more information on the Bridge log see Logging):

- Log entries for connection loss
  `[Error[Internal[ch.e2e.bridge.server.jms[JMS7[lost JMS connection (will try to reconnect every 1000 milliseconds). Reason: javax.jms.JMSException: java.io.EOFException`

  IBM Websphere provides a linked exception to determine the cause of the error. This linked exception is logged to the Bridge log additionally.
  `[Info][Internal][JAVA_ERR][0][JMSException linked exception: MQJE001: ...]`
- Log entry for successful reconnect
  `[Info[Internal[ch.e2e.bridge.server.jms[9[JMS reconnected successful`

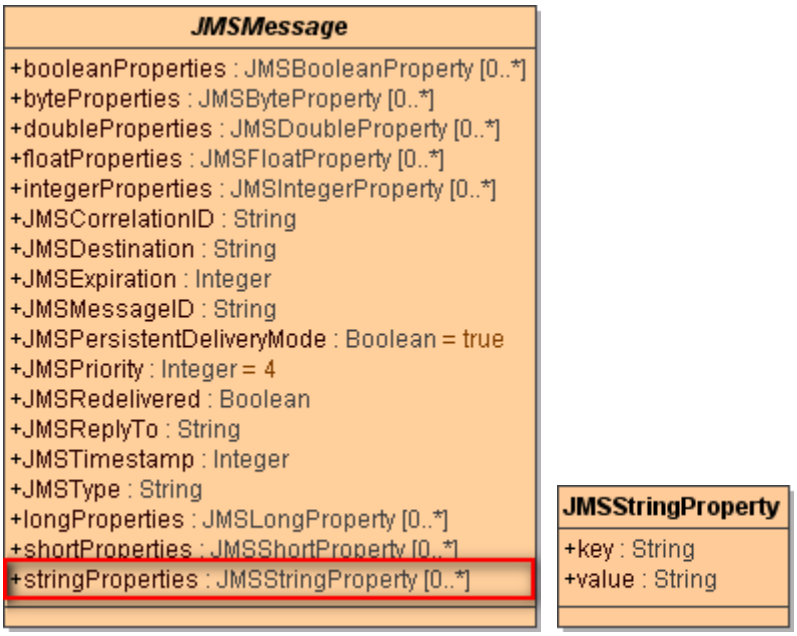
Listening with a Selector

Example Files (Builder project Add-ons/JMS):

<your example path>\Add-ons\JMS\uml\selectiveStatSendListen.xml

To select messages from a topic, messages must provide a least one (or more) message property. Message properties are part of the message header, which is provided to the JMS adapter by the use of either the parameter jmsStringMessage or the parameter jmsBlobMessage.

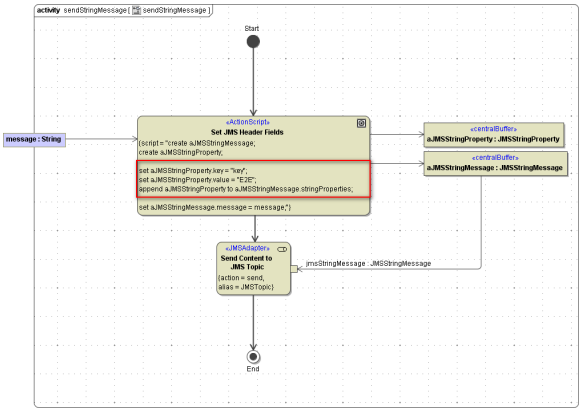
Figure: JMS Message Fields and JMS String Property



Define an activity diagram, e.g. sendStringMessage, that provides the JMS message with a property. The figure below illustrates the basic structure of such an activity diagram. The JMS settings are specified in the component diagram.

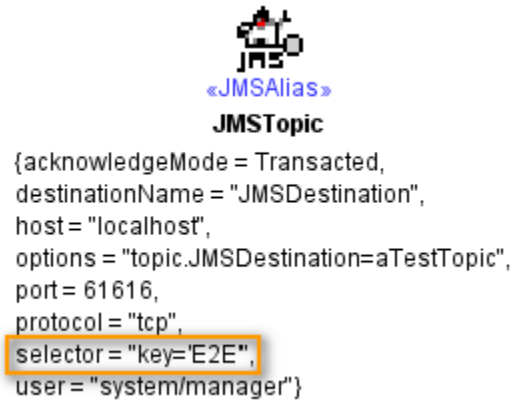
As aJMSMessage .stringPoperties is an array (see figure above), the operation append is used to append the newly defined property to the array.

Figure: Activity Diagram Providing a Property



Define a listener as described before (see Listening on a JMS Topic). In the component diagram on the <<JMSAlias>> of the JMS topic the listener is listening on, add a JMS selector, e.g. `key='E2E'` as shown in the picture below. `key` is the key part of the property, `E2E` is the value part of the property. As `key` is a string property, the value `E2E` must be quoted with single quotation marks.

Figure: JMS Alias with a Selector



For more information on how to draw the component diagram refer to Defining the Components.