

AMQP

ActiveMQ supports the [AMQP 1.0](#) protocol which is an OASIS standard.

Availability

 Available from ActiveMQ version [5.8](#) onward.

Enabling the ActiveMQ Broker for AMQP

To enable AMQP protocol support on the broker add the following transport connector configuration referencing the `amqp` scheme in its URI:


```
<transportConnectors>
  <transportConnector name="amqp" uri="amqp://0.0.0.0:5672"/>
</transportConnectors>
```

It is enabled in the default ActiveMQ server configuration. For more help see [Run Broker](#).

Security

The AMQP implementation fully supports an [ActiveMQ security](#) mechanism. This allows the broker to accept plain SASL authentication. Authorization policies are applied to a destination when it's accessed (read/write).

SSL

 For additional security AMQP can be configured to run over SSL as described in the following section.

Enabling AMQP over NIO

For better scalability (and performance) the AMQP protocol should be configured to use NIO, rather than the default of TCP. To use NIO use the transport scheme `amqp+nio` instead of `amqp`.

Example:

```
<transportConnector name="amqp+nio" uri="amqp+nio://localhost:5672"/>
```

This transport uses the [NIO transport](#) underneath and will generally use much less threads than the standard connector. This connector can help if you want to use [large number of queues](#)

Enabling AMQP over SSL

It's easy to configure ActiveMQ to use AMQP over a SSL connection. To use SSL use the transport scheme `amqp+ssl` instead of `amqp`.

Example:

```
<transportConnector name="amqp+ssl" uri="amqp+ssl://localhost:5671"/>
```

For more details on using SSL with ActiveMQ, see the following article ([How do I use SSL](#)).

Working with Destinations with AMQP

You should prefix destination address with `queue://` to use queue based destinations or `topic://` to use topic based destinations. The destination type defaults to queue when the destination prefix is omitted.

Prefetch Size and Credit

When AMQP receiver connects to the broker it's mapped to the JMS consumer. This JMS consumer has to have appropriate [prefetch size](#) set. The broker will honor the credit set by the client or use the default value of `1000` if client doesn't set it.

Example: tuning the default `prefetch` size:

```
<transportConnector name="amqp" uri="amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.
maxFrameSize=104857600&transport.prefetch=10"/>
```

In this case, client preference will be ignored and the configured value will be used.

You can also tune broker-side `amqp` receiver link that handles incoming messages. It will use credit of 1000 messages by default, but you can override this by using `producerCredit` property, like

```
<transportConnector name="amqp" uri="amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600&transport.producerCredit=10000"/>
```

Mapping to JMS

There are three basic conversion strategies that can be used with AMQP and inter-operating with the JMS API.

Strategy	Description
native	(Default) Wraps the bytes of the AMQP message into a JMS <code>BytesMessage</code> and maps the headers of the AMQP message to headers on the JMS message.
raw	Wraps the bytes of the AMQP message into a JMS <code>BytesMessage</code> .
jms	Maps headers of the AMQP message to JMS message headers and the body of the AMQP message to the JMS body.

Set the `transformer` transport option on the `transportConnector` to the desired mapping strategy. For example, to inter-operate with JMS at the payload level, set the `transformer` option to `jms`:

```
<transportConnector name="amqp" uri="amqp://localhost:5672?transport.transformer=jms"/>
```

How AMQP Message Headers are Mapped to JMS Headers

The following headers are mapped regardless of the transformer used:

AMQP Message	JMS Message	Notes
	JMS_AMQP_NATIVE	Will be set to <code>true</code> if the transformer is <code>native</code> or <code>raw</code> , <code>false</code> otherwise.
message-format	JMS_AMQP_MESSAGE_FORMAT	

The following header mappings apply when the transformer is either `native` or `jms`:

AMQP Message	JMS Message	Notes
application-properties.JMSXGroupID	JMSXGroupID	
application-properties.JMSXGroupSequence	JMSXGroupSequence	
application-properties.JMSXUserID	JMSXUserID	
application-properties. <u>name</u>	<u>name</u>	
delivery-annotations. <u>name</u>	JMS_AMQP_DA_ <u>name</u>	
footer. <u>name</u>	JMS_AMQP_FT_ <u>name</u>	
header.deliveryCount	JMSXDeliveryCount	
header.durable	JMSDeliveryMode	<code>javax.jms.Message.DEFAULT_DELIVERY_MODE</code> if not set.
header.first-acquirer	JMS_AMQP_FirstAcquirer	
header.priority	JMSPriority	<code>javax.jms.Message.DEFAULT_PRIORITY</code> if not set.
header.ttl	JMSExpiration	<code>javax.jms.Message.DEFAULT_TIME_TO_LIVE</code> if not set.
message-annotations. <u>name</u>	JMS_AMQP_MA_ <u>name</u>	
message-annotations.x-opt-jms-type	JMSType	
message-annotations.x-opt-reply-type	Type of the JMSReplyTo	Comma separated list of <code>queue</code> , <code>topic</code> , or <code>temporary</code> . Defaults to <code>queue</code> if not set.

message-annotations.x-opt-to-type	Type of the JMSDestination	Comma separated list of queue , topic , or temporary . Defaults to queue if not set.
properties.content-encoding	JMS_AMQP_ContentEncoding	
properties.content-type	JMS_AMQP_ContentType	
properties.correlation-id	JMSCorrelationID	
properties.creation-time	JMSTimestamp	
properties.group-sequence	JMSXGroupSequence	
properties.message-id	JMSMessageID	Auto generated if not set.
properties.reply-to	JMSReplyTo	The name of the JMSReplyTo
properties.reply-to-group-id	JMS_AMQP_ReplyToGroupID	
properties.subject	JMS_AMQP_Subject	
properties.to	JMSDestination	The name of the JMSDestination
properties.user-id	JMSXUserID	properties.user-id is decoded as a UTF-8 String.

AMQP property value types are converted as follows:

AMQP Type	Java Type	Notes
binary	String	Hex encoding of the binary value
bool	Boolean	
byte	Byte	
double	Double	
float	Float	
int	Integer	
long	Long	
short	Short	
symbol	String	
ubyte	Byte or Short	Short is used if: value > Byte.MAX_VALUE
uint	Integer or Long	Long is used if: value > Integer.MAX_VALUE
ulong	Long	
ushort	Short or Integer	Integer is used if: value > Short.MAX_VALUE

How a AMQP Messages Body is Mapped to a JMS Message

If the transformer is set to `jms` then the JMS message type will depend on the body type of the AMQP message.

Body Type	JMS Message Type
AmqpSequence	StreamMessage
AmqpValue	ObjectMessage
AmqpValue holding a null	Message
AmqpValue holding a String	TextMessage
AmqpValue holding a binary	BytesMessage
AmqpValue holding a list	StreamMessage

Data	BytesMessage
null	Message

AMQP 1.0 client library

You can use [Apache Qpid Proton](#).