JMS Concepts

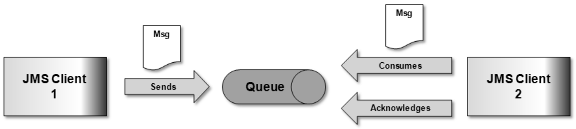
On this Page:

JMS Domains

Messaging products can be classified broadly as either point-to-point (PTP) or publish-subscribe (Pub/Sub) systems. These two messaging styles are often referred to as messaging domains.

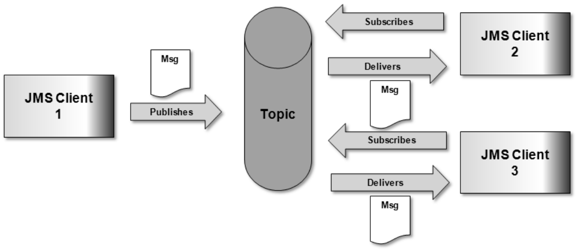
PTP products are built around the concept of message queues. Each message is addressed to a specific queue; clients extract messages from the queue(s). Each message is consumed by one (and only one) consumer, which acknowledges the recipience of the message to the JMS provider. If the client is temporarily not available, the message can be stored and then received at any time later.

Figure: Concept of Point-to-Point Messaging



Pub/Sub clients address messages to some node (topic) in a content hierarchy. Publishers and subscribers are generally anonymous and may publish or subscribe dynamically to the content hierarchy. The system takes care of distributing the messages arriving from multiple publishers of a node to its multiple subscribers (or none).

Figure: Concept of Public-Subscribe-Messaging



The Bridge only supports non-durable subscribers. That implies that subscribers to a topic receive messages only when they are active listening on that topic. Messages sent during inactivity of the subscriber will never get delivered.

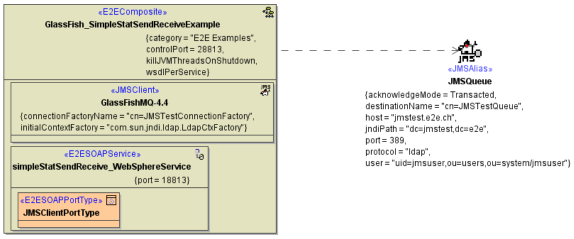
When using JMS API version 1.1, a developer can use interfaces and methods that support both models of messaging. The JMS adapter supports both messaging styles.

JMS Provider

A JMS provider is an entity that implements JMS for a messaging product, e.g. WebSphere ^{TM} MQ from IBM.

> The JMS adapter requires JMS providers to be JMS 1.1-compliant.

JMS providers are defined in the component diagram. For more information on how to draw the component diagram of a service using JMS refer to chapter Defining the Components.



«E2EComposite»
GlassFish_SimpleStatSendReceiveExample
{category = "E2E Examples", controlPort = 28813, killJVMThreadsOnShutdown, wsdlPerService}
«JMSClient»
GlassFishMQ-4.4
{connectionFactoryName = "cn=JMSTestConnectionFactory", initialContextFactory = "com.sun.jndi.ldap.LdapCtxFactory"}
«E2ESOAPService»
simpleStatSendReceive_WebSphereService
{port = 18813}
«E2ESOAPPortType»
JMSClientPortType
JMS
«JMSAlias»
JMSQueue
{acknowledgeMode = Transacted, destinationName = "cn=JMSTestQueue", host = "jmstest.e2e.ch", jndiPath = "dc=jmstest,dc=e2e", port = 389, protocol = "ldap", user = "uid=jmsuser,ou=users,ou=system/jmsuser"}

JMS Client

All users of JMS are referred to generically as clients. These clients use the JMS implementations supplied by their JMS provider.

JMS Transaction Concept

A JMS session is strongly coupled to an xUML Runtime session (see also Fast-running ACID Transactions). In every Runtime session exists exactly one JMS session connected to a JMS Provider.