

HPE NonStop Message Queue Software

Java Messaging System on NonStop



Insight

HPE NSMQ 1.1 is based on Apache ActiveMQ version 5.9.1.

What NSMQ is

The HPE NonStop Message Queue (NSMQ) product brings the open source Apache ActiveMQ on to NonStop. Apache ActiveMQ is one of the most popular Java Messaging Service (JMS) implementations. NonStop continues to expand its offerings in the Java and middleware domains so as to make the development and deployment easier for application developers. NSMQ is one more product from the NonStop stable in that direction. It offers a near-continuously available, near-linearly scalable, and standards compliant messaging system for applications to communicate with each other.

About HPE NSMQ 1.1 release

NSMQ is a value added port of Apache ActiveMQ version 5.9.1. It complies with the Java community specifications for JMS 1.1.¹ It offers plain Java application programming interfaces (APIs) for client applications to send and receive messages via the queue.

What's new in NSMQ 1.1?

- Web GUI based Management Interface

Key features from NSMQ 1.0

- Compliance to JMS 1.1.1 specifications (for Java native APIs)
- Java Connectors Architecture (JCA) compliant resource adaptor (RA) for application servers to connect
- Support for both queues and topics²
- Near-continuous availability and linear scalability by using ActiveMQ clustering mechanism
- Participation in XA transactions
- Message persistence using SQL/MX

^{1,2} NSMQ 1.1 release does not support durable receivers/subscribers for topics. This feature is targeted for a future NSMQ release.

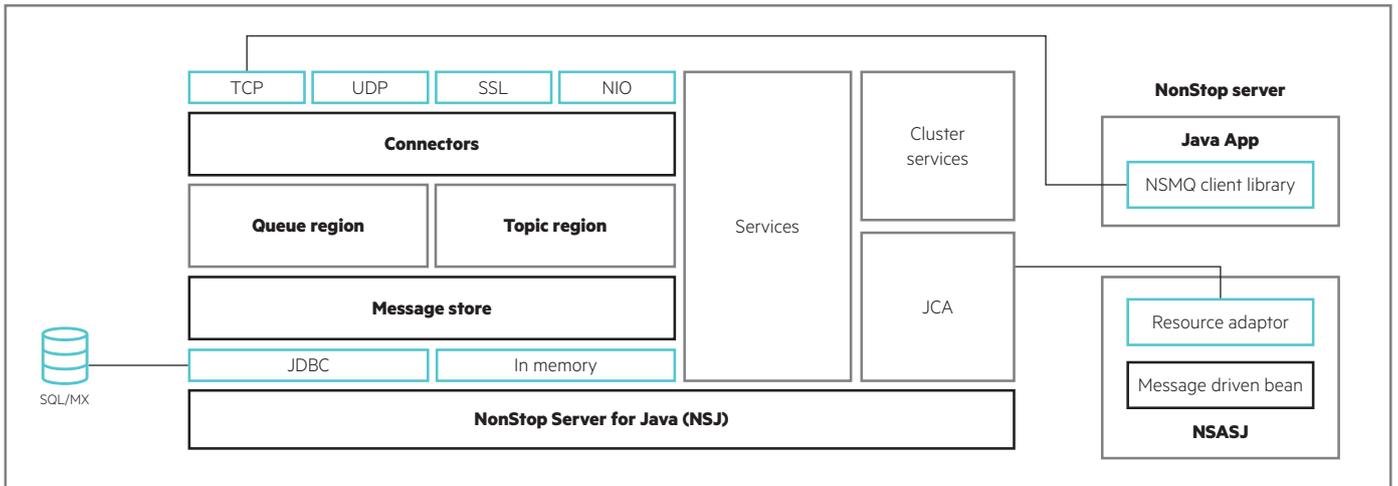


Figure 1: HPE NSMQ architecture

- Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Java New I/O (NIO) transport protocols for clients to connect to brokers. Communication with security is supported through SSL 3.0 protocol
- Java Authentication and Authorization Services (JAAS) authentication and authorization

Architectural components and description of features

Connectivity between JMS clients and the brokers

The applications that use NSMQ to transfer messages are called “clients.” There are different types of clients, and they are explained in the “Queues” and “Topics” sections.

The transport connectors provide connectivity to clients for transmitting and receiving their messages to and from the NSMQ brokers—TCP, UDP, NIO, and SSL protocols are supported. The client uses a simple URI as shown below to address the NSMQ broker. The client can use any one of the transport protocols listed.

```
[tcp][udp][nio][ssl]:
//hostname:portnumber?TransportOptions
```

The actual message itself is transported using the OpenWire protocol. OpenWire defines the message structures (data types and encodings). It is the default message transfer layer protocol of ActiveMQ, and therefore NSMQ.

For achieving near-continuous availability of NSMQ on NonStop, there is a “failover” protocol that enables clients to connect to multiple brokers so that single point of failure (SPOF) does not affect their operation. Its syntax is as written below. The clients use this syntax to get a connection to a broker instance. In the event of a broker failure, the failover protocol automatically reconnects the client to another available broker.

```
failover://URITransportOptions
```

NSMQ also provides a JCA compliant RA that can be used by application servers to integrate. The RA supports TCP protocol for transport.

Applications developed on ActiveMQ on other platforms will not require any change to be deployed on NSMQ from JMS functionality perspective and vice versa.³

³ NSMQ 1.1 release does not support durable receivers/subscribers for topics. This feature is targeted for a future NSMQ release.

Queues

NSMQ fully supports point-to-point, first-in-first-out JMS queues. Messages are consumed in the order they are received into the queue. A “producer” client sends a message into the queue and a “consumer” client picks it up. Each message in the queue is delivered to only one receiver. There can be multiple senders and receivers configured for a particular queue. The message is delivered on a first-come-first-served basis to a receiver. The message is deleted from the queue after successful delivery.

A sender can either use synchronous or asynchronous method to drop a message. In synchronous method, the sender waits till the queue broker acknowledges the receipt of the message. Similarly, the receiver may acknowledge receiving a message to the broker depending on the configuration. A message that does get picked by any receiver may get deleted after a certain expiry period.

Topics

Topics are used for broadcasting a message to multiple receivers. A “publisher” drops a message into the queue and each “subscriber” receives a copy of the same message. A message is deleted from the topic once all subscribers have received their copies or after a configured expiry time. A “publisher” can send a message either synchronously or asynchronously.

Composite destination

A string of comma-separated individual destinations is collectively called a composite destination. It can be considered as one combined destination while registering with Java Naming and Directory Interface (JNDI), but the messages destined for it are dispatched to the individual queues that constitute the composite destination. The composite destination may be made up of either queues or topics (homogenous) or it may be made up of a mixture of queues and topics (heterogeneous).

Wildcards

Wildcards allow the clients to bind to multiple destinations at the same time. Three wildcard characters namely “.”, “*”, and “>” can be used by JMS clients to attach to multiple queues or topics. The “.” separates segments in a destination path name. The “*” allows binding to all the segments in a destination path name that it substitutes. The “>” allows binding to all the segments that follow its position in a destination path name.

JMS transactions

NSMQ supports JMS transactions in full on both the message sending and receiving directions. Transaction, Commit, and Rollback are supported with respect to both sender and receiver. In a committed transaction the message is delivered to its consumer. In case the sender rolls back the transaction, the message is not delivered to the consumer(s). In case the receiver rolls back, the message is put back into the queue by that receiver, and it may get redelivered till the maximum redelivery attempts exceed or the lifetime of the message expires. JMS transactions invoked by client applications are not part of the NonStop TMF transactions.

XA transactions

A Java Transaction API (JTA) compliant XA transaction library is provided for NSMQ clients. XA transactions are characterized by a two-phase commit/or rollback which is supported by this library. Messages are cached in a transaction store by the broker without performing the action desired by the client. The client can issue a “prepare” command followed by either a “commit” or “rollback” command instructing the broker to complete the transaction. This library encapsulates NonStop transaction management facility (TMF) transaction inside the XA transaction, and hence NSMQ XA transactions are part of the NonStop TMF transactions.

Message persistence

NSMQ supports message delivery with persistence. This feature may be used by mission-critical clients where messages cannot be lost in transit between the sender/publisher and the receiver/subscriber. The NSMQ broker stores the message submitted by the sender/publisher in a non-volatile storage system. Once the

message is successfully picked up by the receiver/subscriber the stored message is deleted from the nonvolatile storage system. HPE NonStop SQL/MX is used for storing the messages that need to be persisted. When a failed broker restarts, it fetches the messages previously stored in the database (DB) and makes them available to the receiver/subscriber.

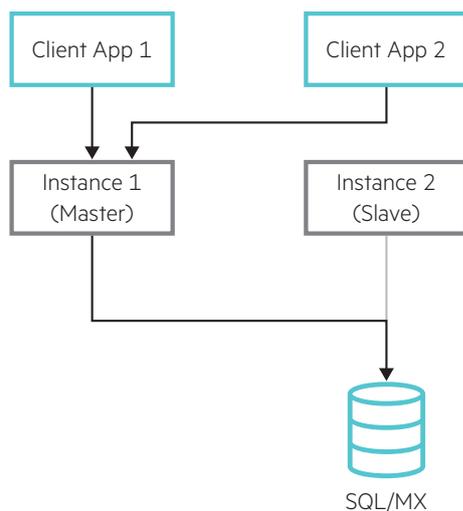


Figure 2: Before master fails

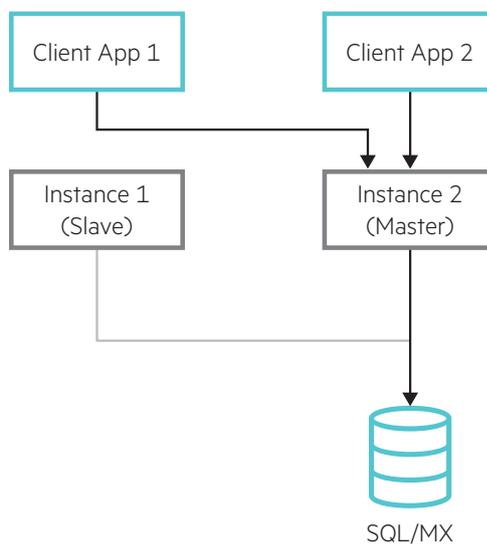


Figure 3: After master has failed and recovered

Near-continuous availability

NSMQ, being a NonStop product, is designed for mission-critical applications and hence has built-in redundancy to provide near-continuous availability.

NSMQ uses DB based master-slave cluster redundancy. Two brokers can be instantiated—one becomes the master and the other becomes the slave. The master handles all interactions with the clients and holds all messages in the SQL/MX database.

In the event of a master failure, the slave takes over as the new master and continues operations using the data persisted in the SQL/MX DB.

NonStop TS/MP is used for process management of the NSMQ brokers. This enables the failed master is brought up automatically. However, the restarted broker remains a slave and does not take over the client operations till the new master fails.

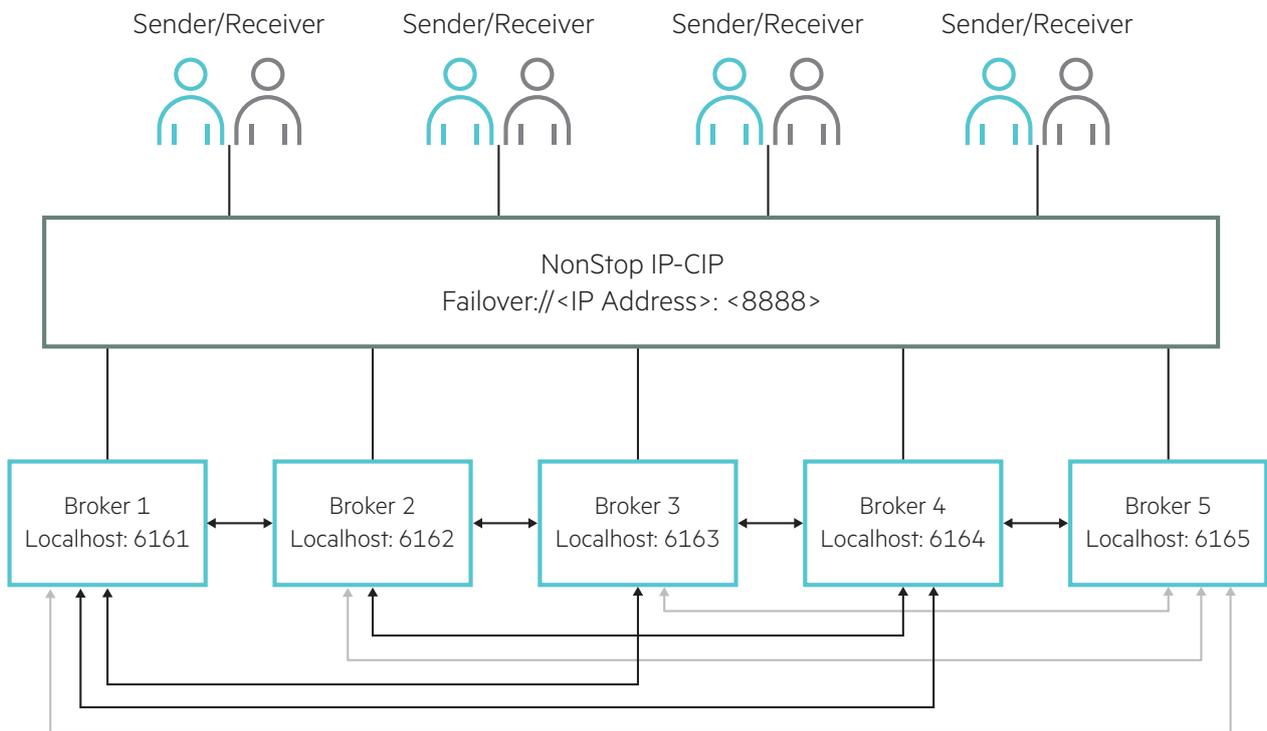


Figure 4: HPE NSMQ architecture

Scalability

A network of NSMQ brokers provides scalability. All brokers listen on a common published uniform resource identifier (URI). The NonStop parallel I/O clustering (IP-CIP subsystem) allows all the brokers to listen to a common URI. Thus, the incoming client requests are evenly load balanced across the available brokers. Additional brokers can be added, as shown in figure 4 where Broker 5 is being added, (or an existing broker can be removed) without disturbing the service.

Each broker is connected with a duplex connection to all the other brokers. The subscription information about all the clients is known to all the brokers in the network. Subscription contains information about a client being a sender/publisher or a receiver/subscriber to a particular queue/topic, the broker managing that queue/topic, and others. Therefore, messages get routed to the appropriate broker to which the client is connected to.

Security

Authentication

All clients may be authenticated by the broker using the username and password configured. Either a JAAS plug-in or a simple authentication plug-in XML file can be used to configure the usernames and passwords for the sender/publisher and receiver/subscriber clients. Users can also be combined into user groups.

The passwords can be encrypted. The encrypted passwords can be specified as variables in the simple authentication plug-in. NSMQ also supports a JAAS plug-in for configuring the authentication parameters. By default, NSMQ JAAS plug-in uses a plain-text properties file.

Authorization

There are three levels of access to a broker from a client in NSMQ. They are Read, Write, and Admin. As their names suggest, Read allows a client to consume a message from a queue/topic and Write allows it to drop a message to a queue/topic. Admin allows a client to dynamically create a new destination, which does not exist and then wither a Read or Write operation. The permissions for the three levels can be configured in an authorization plug-in file per user group on a per queue or topic basis.

Secure Socket Layer

SSL can be used by clients to protect the privacy and integrity of the messages that they either transmit to or receive from the NSMQ broker. NSMQ uses the inbuilt framework in HPE NonStop Server for Java (NSJ) to provide the SSL connectivity. The framework provides a keystore to save the X.509 certificates and private keys.

Management of NSMQ

NSMQ can be managed using Java Management Extensions (JMX). JConsole or JVisualVM can be used to connect to NSMQ for managing the brokers, connections, queues, topics, and subscriptions. Administrators of NSMQ can be provided with “readonly” or “readwrite” privileges for operations on the MBeans.

NSMQ also provides a minimal CLI for browsing messages in a queue or topic, purging messages, and querying JMX context of MBeans.

From NSMQ 1.1 onwards management through a Web-based GUI is supported. The NSMQ Web console is a standard Java EE Web application and is deployed in a standard Servlet container. Following operations are supported via the GUI management console:

- Administrative tasks such as adding/deleting NSMQ Web Console users, adding/updating/deleting NSMQ installations and assigning Web console users to NSMQ clusters
- Managing a broker
 - Queue management—Creating a queue, browsing messages in a queue, purging messages, and deleting the queue
 - Topic management—Creating and deleting a topic
 - Viewing details of active connections on a broker

Customize your IT lifecycle management, from acquisition of new IT, management of existing assets, and removal of unneeded equipment.

hp.com/go/hpfinancialservices

System requirements of the host

- HPE Integrity NonStop i servers or HPE Integrity NonStop X servers
- HPE NonStop OS release version H06.25/J06.14/L15.02 or later Open System Services (OSS) OS environment

Note: Please refer to the latest edition of Software Products Maintenance List (SPML) to know the support status for different RVU versions.

- HPE NonStop TCP/IPv6 or IP-CIP for networking

- HPE NonStop Server for Java 6 (NSJ 6) or later⁴ on HPE Integrity NonStop i servers
- HPE NonStop Server for Java 7 (NSJ 7) or later on HPE Integrity NonStop X servers
- HPE NonStop TS/MP version 2.3, 2.5, or later
- HPE NonStop SQL/MX version 3.2 or later (optional if persistence of messages is required)
- NonStop TMF version 3.7 or later
- HPE NonStop Application Server for Java version 1.0 (NSASJ 1.0) or later (optional)

Product ordering information

Series	Part number	Product name
L-series	BE143AC	HPE NonStop Message Queue SW
J-series	QNSMQ01V1	HPE NonStop Message Queue SW
H-series	HNSMQ01V1	HPE NonStop Message Queue SW

Get the support you need

HPE Technology Services offer flexible choices that span the entire technology lifecycle, and help build an infrastructure that is reliable, highly available, responsive, and rooted in proven best practices. We offer a support experience that is proactive, personalized, and simplified—delivering support when and how you need. HPE recommends the following services:

HPE Critical Service (Optimized Care)

High performance reactive and proactive support designed to help minimize downtime. It offers an assigned support team, which includes an account support manager (ASM). This service offers access to HPE's Global Mission Critical Solution Center, 24x7 hardware and software support, 6-hour call-to-repair commitment, enhanced parts inventory, and accelerated escalation management.

HPE Proactive 24 (Standard Care)

HPE Proactive 24 provides proactive and reactive support delivered under the direction of an ASM. It offers 24x7 hardware support with 4-hour onsite response, 24x7 software support with a 2-hour response and flexible call submittal.

HPE Support Plus 24 (Basic Care)

HPE Support Plus 24 provides reactive hardware and software support with remote problem diagnosis, 4-hour onsite response, and parts replacement. The software support includes installation advisory support, software updates for HPE, and selected third-party software products.

HPE Installation and Start-up Services

This service provides efficient and effective deployment of HPE hardware components.

For more information on our services, visit

hp.com/services/nonstop.

Learn more at
hp.com/go/NonStop



⁴ Web-based management GUI requires NSMQ Web Console to be deployed in a Servlet container running on NSJ 7. NSJ 7 requires H06.26/J06.15/L15.02 as the minimum RVU version.



Sign up for updates



© Copyright 2013–2016 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Java and the Java logo are registered trademarks of Oracle and/or its affiliates.

4AA4-7064ENN, November 2016, Rev. 3