



89 Fifth Avenue, 7th Floor

New York, NY 10003

[www.TheEdison.com](http://www.TheEdison.com)

212.367.7400

[Info@TheEdison.com](mailto:Info@TheEdison.com)

## 백서

---

**IBM WebSphere MQ 7.5 대 Apache  
ActiveMQ 5.9 비교: 페일오버, 트랜잭션  
무결성 및 관리**

Printed in the United States of America

Copyright 2014 Edison Group, Inc. New York.

Edison Group 은 여기 수록된 정보에 대해 명시적 또는 묵시적인 어떠한 보증도 제공하지 않으며 이 문서의 사용으로 인해 발생하는 오류를 책임지지 않습니다.

모든 제품은 해당 소유권자의 상표입니다.

초판: 2014 년 5 월

발행: Bill Karounos(메시징 전문가), Hanny Hindi(애널리스트), Manny Frishberg(편집자), Barry Cohen(편집장)

## 목차

---

전체 개요 .....	2
안정성 및 페일오버 .....	2
트랜잭션 무결성 및 관리 .....	7
성능.....	11
설치 및 구성.....	13
관리 및 운영.....	14
부록 1: 기능 비교.....	27
부록 2: 이 백서에서 다루지 않은 항목.....	30

## 전체 개요

---

이 백서에서는 IBM WebSphere MQ 7.5 와 Apache Software Foundation 의 ActiveMQ 5.9 를 비교하고, 안정성, 신뢰성, 사용 편의성, 성능 및 관리 기능 등의 기술 요소를 자세히 분석합니다.

ActiveMQ 및 WebSphere MQ 는 모두 기본적인 메시징 요구사항을 충족합니다. 그러나 고가용성 및 강력한 페일오버 기능을 원하는 기업 고객이라면 다음과 같은 이유로 WebSphere MQ 를 고려해 볼 필요가 있습니다.

- **페일오버:** ActiveMQ 는 "정전" 및 "네트워크 중단" 상황에서 메시지가 유실되거나 중복되었습니다. 이것은 기업 환경에서는 용납할 수 없는 사고입니다.
- **문서화:** IBM 의 설명서는 특히 구성, 관리, API 문서는 물론 클러스터링, 로드 밸런싱, 고가용성 같은 고급 기능에서 Apache 설명서보다 훨씬 자세하며 최신 내용을 담고 있습니다.
- **성능:** 퍼시스턴트 테스트 결과, WebSphere MQ 는 256 바이트에서 1MB 에 이르는 메시지를 60 ~ 90% 더 빠른 속도로 처리했습니다. (네트워크의 제약으로 인해 non-persistent 테스트는 결론에 이르지 못했으나, 초기 결과에서는 역시 WebSphere MQ 가 우수한 성능을 보였습니다.)
- **트랜잭션 관리:** 두 시스템 간의 가장 큰 차이점은 트랜잭션 관리의 용이성이었습니다. WebSphere MQ 는 기본 기능으로 데이터베이스와 메시징 서버 사이의 트랜잭션을 관리할 수 있었습니다. ActiveMQ 로 2PC 트랜잭션을 관리하기 위해서는 XA 가 지원되는 외부 애플리케이션 서버가 필요합니다.
- **운영:** ActiveMQ 의 웹 콘솔은 매우 제한적인 기능을 제공합니다. 여러 가지 기본 기능은 물론 큐 편집이나 메시지 최대 크기 변경 등 고급 기능을 사용하려면 사용자가 직접 구성 파일을 편집해야 합니다. 게다가 ActiveMQ 는 브로커마다 별도의 브라우저 창을 열고 고유한 URL 을 사용해야 하는 반면, WebSphere MQ Explorer 는 사용자가 인터페이스 하나에서 브로커 여러 개를 운영할 수 있습니다.
- **플랫폼 호환성:** WebSphere MQ 는 Windows, Linux, Solaris, HP-UX 등 다양한 플랫폼을 "지원"할 뿐만 아니라 정식 인증을 받았습니다. 반면 ActiveMQ 는 이러한 플랫폼 대부분에서 작동하기는 하지만, 많은 실 운영 환경에서 여전히 중요한 역할을 담당하는 System z 메인프레임 등의 플랫폼 환경에서는 확실한 인증을 받지는 못했습니다.

고가용성, 신뢰성, 사용 편의성, 자세한 설명서 및 플랫폼 호환성이 중요하지 않다면, ActiveMQ 도 좋은 플랫폼이 될 수 있습니다. 그러나 신뢰성을 요구하는 기업 고객이라면 WebSphere MQ 가 훨씬 더 나은 선택입니다.

## 안정성 및 페일오버

Edison은 "정전" 및 "네트워크 중단"이라는 두 가지 공통 시나리오에서 WebSphere MQ와 ActiveMQ의 페일오버 기능 및 안정성을 테스트했습니다.

"정전" 상황은 메인 서버를 실행 중인 가상 시스템의 전원을 끄는 방식으로 시뮬레이션했습니다. "네트워크 중단" 상황은 NIC(네트워크 인터페이스 컨트롤러)의 전원을 끄는 방식으로 시뮬레이션했습니다. 예상한 바와 같이, 정전된 시간 동안에는 일부 메시지가 클라이언트 시스템에서 서버로 전달되지 않았습니다. 그러나 중단 상태에서 복구하는 과정은 두 시스템이 차이가 있었습니다.

WebSphere MQ의 복구 방식이 확실히 더 우수했습니다.

테스트 시나리오	ActiveMQ	WebSphere MQ
"정전"	평균적으로 약 2%의 메시지가 중복되었으며, 유실된 메시지는 없음	유실되거나 중복된 메시지 없음
"네트워크 중단"	평균적으로 약 2%의 메시지가 중복되었으며, 재시작 시 메시지 100% 유실됨. ActiveMQ는 네트워크가 복구된 후 서버 간에 원래의 마스터-슬레이브 관계를 복원하지 못함.	유실되거나 중복된 메시지 없음

메시징 시스템의 핵심 기능은 메시지를 *한 번, 단 한 번만* 안정적으로 전달해야 한다는 것입니다. 메시지가 중복되고 유실된 ActiveMQ는 이 핵심 원칙을 제대로 지키지 못한 것입니다.

### ActiveMQ 구성 #1: 마스터-슬레이브

- 공유 파일 시스템(NFSv4)
- 마스터-슬레이브 구성의 VM 두 대에서 ActiveMQ 브로커 두 개 실행
  - 서버 하나는 VM1에서 마스터 브로커로 실행
  - 다른 하나는 VM2에서 슬레이브 브로커로 실행

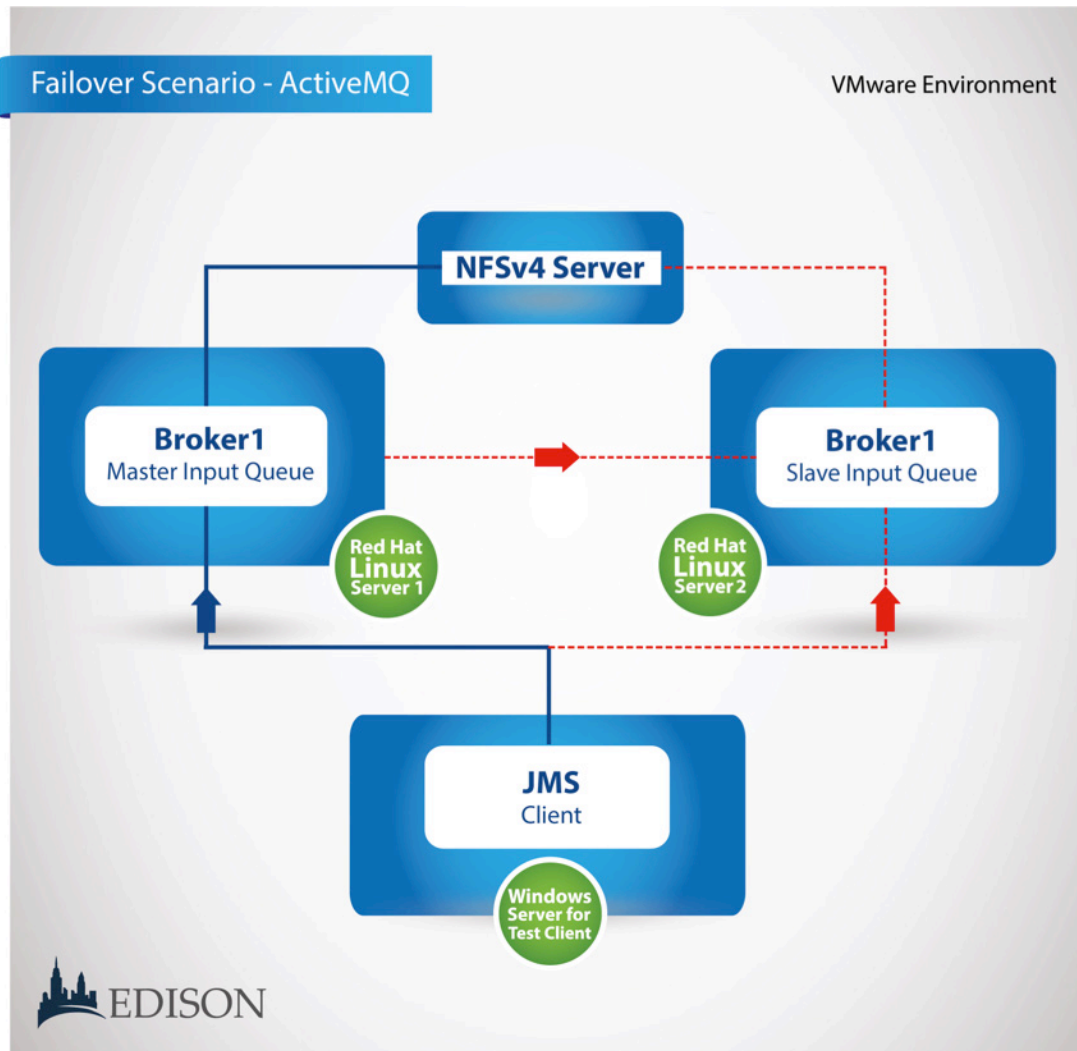


그림 1: ActiveMQ 5.9 페일오버 테스트 환경  
WebSphere MQ 구성 #1: 활동 대기

- 공유 파일 시스템(NFSv4)<sup>1</sup>
- 다중 인스턴스 큐 매니저로 구성된 WebSphere MQ 서버 두 대<sup>2</sup>
  - 서버 하나에서 액티브 큐 매니저 실행

<sup>1</sup> Edison의 테스트 시나리오에서는 공유 파일 시스템을 이용한 HA/페일오버 기능을 테스트하지 않았습니다. 실제 운영 환경에서는 가용성 및 중복성을 보장하기 위해 HA 서버, NAS 또는 다른 구성요소가 꼭 필요합니다.

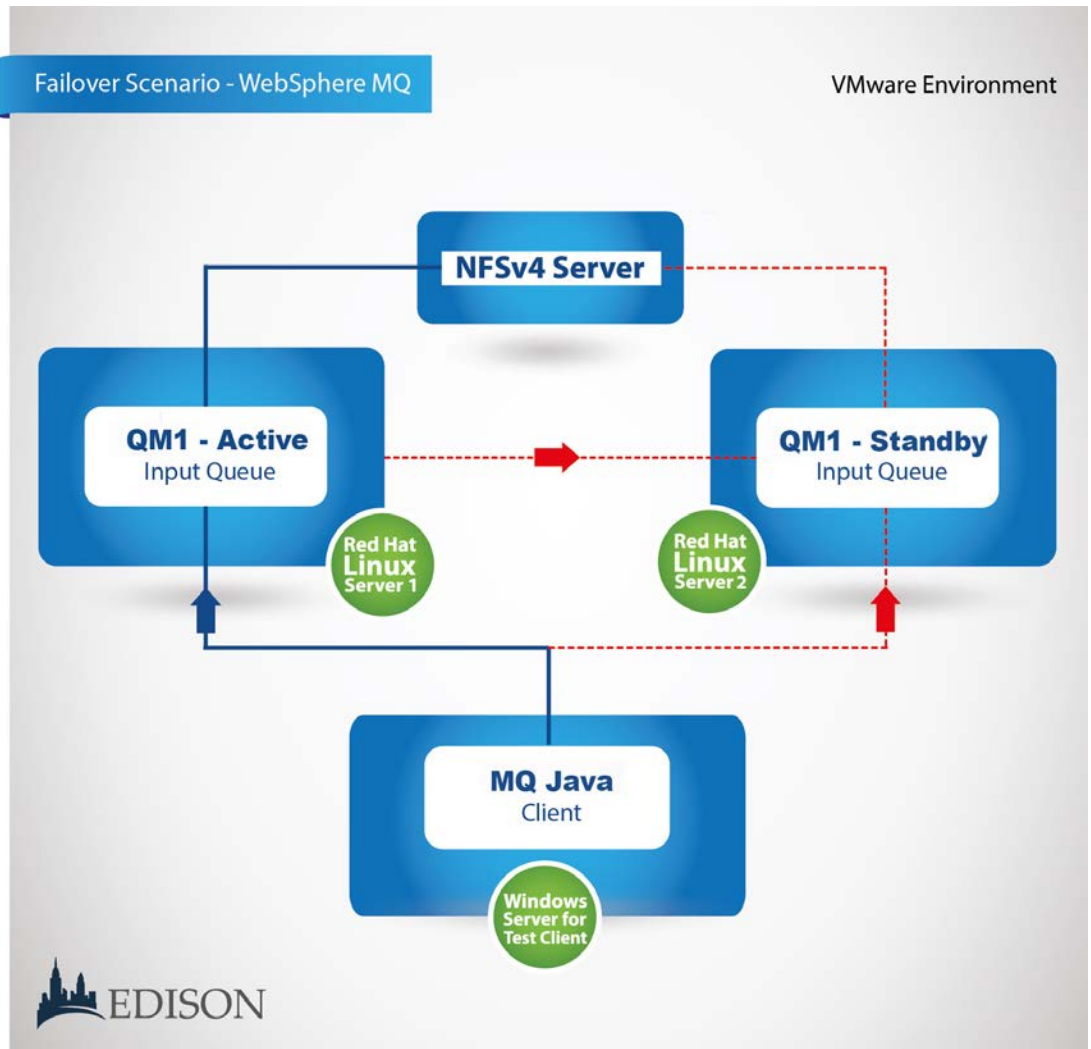
<sup>2</sup> 다중 인스턴스 큐 매니저를 만들고 구성하는 방법에 관한 설명서:

[http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp?topic=%2Fcom.ibm.mq.con.doc%2Fq018150 .htm](http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp?topic=%2Fcom.ibm.mq.con.doc%2Fq018150.htm)

- 다른 서버는 스탠바이 큐 매니저 대기 중

참고: WebSphere MQ 용어로 "다중 액티브/스탠바이"는 Active MQ 의 "마스터-슬레이브" 구성에 해당합니다.-{-}

그림 2:  
IBM



WebSphere MQ 7.5 페일오버 테스트 환경

### 네트워크 중단 시뮬레이션

Edison은 LoadRunner Java 스크립트를 사용하여 큐 매니저(WMQ) 또는 브로커(AMQ)와 양쪽 시스템 간에 메시지를 보내고 받아 보았습니다. WebSphere MQ에는 표준 MQ Java

API(com.ibm.mq.\*)를 사용했습니다.<sup>3</sup> ActiveMQ에는 Java API(org.apache.ActiveMQ.\*) 및 JMS를 사용했습니다.

네트워크 중단 상황을 시뮬레이션하여 파일오버를 일으키기 위해 클라이언트에서 메시지를 보내고 받는 동안 VMware vSphere Client 로 네트워크 인터페이스 컨트롤러(NIC)의 연결을 끊었습니다. 정전 상황을 시뮬레이션하기 위해서는 VM(가상 시스템)의 전원을 차단했습니다.

Edison 은 ActiveMQ 와 WMQ 의 대응 방식을 보여 주는 제어 환경의 파일오버 시나리오를 동영상으로 제작했습니다. YouTube 의 다음 주소에서 이 동영상을 보실 수 있습니다.

- **ActiveMQ:** <https://www.youtube.com/watch?v=ObFSO38lq1k>
- **WebSphere MQ:** [https://www.youtube.com/watch?v=bS5r3\\_gkU9k](https://www.youtube.com/watch?v=bS5r3_gkU9k)

*참고:* 위에 링크된 동영상은 이 백서에서 설명하는 완전한 테스트 결과가 아니라 제어 환경의 간단한 파일오버 시나리오만을 보여 줍니다. 즉, VMware 환경에서 NIC 를 끄는 네트워크 중단 시나리오를 통해 ActiveMQ 의 마스터/슬레이브 설정을 WebSphere MQ 의 다중 인스턴스 관리자와 비교한 것입니다. 이 테스트에서는 ActiveMQ 의 "브로커 네트워크" 또는 WMQ 의 "클러스터링"을 구현하지 않았습니다. 이 설정은 아래의 "관리 및 운영" 섹션에서 설명합니다.

주요받은 메시지를 비교하는 과정에서 다음과 같은 사실을 발견했습니다.

### **"정전": ActiveMQ 결과**

- 예상한 바와 같이, 원래의 마스터가 중단된 후 "슬레이브" 서버가 "마스터"가 되었습니다.
- 실제 파일오버 과정에서는 클라이언트가 처음에는 마스터로부터 메시지를 받다가 파일오버가 끝난 뒤에는 새 슬레이브 서버로부터 메시지를 받으면서 메시지 몇 개가 중복되었습니다.

### **"정전": WebSphere MQ 결과**

- 유실되거나 중복된 메시지가 없었습니다.

---

<sup>3</sup> Edison 은 핵심 API 를 사용하는 테스트 외에 JMS 를 사용하는 격리 테스트도 실시하여 유사한 결과를 얻었습니다.



- 클라이언트는 네트워크 작동이 중단된 동안 전송에 실패한 모든 메시지에 대해 통보를 받았습니다.

### "네트워크 중단": ActiveMQ 결과

- 원래의 마스터에서 NIC가 복원되자, 기본 브로커에 연결할 수 없을 때는 페일오버하도록 구성된 수신 클라이언트로 여러 개의 메시지가 중복되어 보내졌습니다.
- 웹 콘솔을 검토한 결과, 클라이언트 시스템으로부터 수많은 메시지가 "보내진" 것은 사실이나 복원 후 이러한 메시지를 받지 못했으며 "전송 실패"로 기록되지도 않았습니다.
- 원래 마스터와의 연결이 복원되면 그 마스터가 슬레이브 서버가 되어야 합니다. 왜냐하면 원래의 슬레이브 서버가 액티브 마스터가 되어 공유 파일 시스템을 잠갔기 때문입니다. 그러나 원래 마스터는 그 대신 또 하나의 마스터 서버로 시작되었으며, 원래 슬레이브로부터 파일 잠금을 가져왔습니다. 원래 슬레이브는 "마스터" 상태를 유지했으나 연결된 클라이언트로부터 더 이상 메시지를 받지 못했습니다. 이 구성에서는 이후의 페일오버가 불가능합니다. 슬레이브 서버의 하드웨어와 네트워크가 정상적으로 작동 중임에도 불구하고, 구성요소 중 하나라도 장애를 일으키면 메시징 서비스 전체가 완전히 중단될 것입니다.
- VM을 재시작하자 NIC가 복원된 후 슬레이브 서버의 웹 콘솔에 표시되었던 지속적 메시지 수가 드러났습니다. 또한 마스터-슬레이브 구성도 중단 사태 후 복원되지 않았습니다. 이것은 ActiveMQ의 한계입니다.<sup>4</sup> 대신에 "마스터" 브로커가 두 개가 되었습니다. 다시 연결된 마스터 브로커는 마스터 파일 잠금 소유권을 다시 가져오고, 예전 마스터 브로커는 메시지를 보내고 받는 역할을 했습니다.
- 마스터-슬레이브 구성을 복원하기 위해서는 서버를 재시작해야 했습니다. 그러나 저장된 메시지에 대한 시스템 경고는 표시되지 않았으며, 재시작과 동시에 해당 메시지는 *유실되었습니다*. 이 메시지들을 복원하려면 운영자가 메시지가 있었다는 사실을 알고, 수동으로 백업한 다음<sup>5</sup>, 재시작 후 복원해야 합니다. 실제 운영 환경에서 이러한 작업을 수동으로 관리하기는 어렵습니다.

### "네트워크 중단": WebSphere MQ 결과

- 유실되거나 중복된 메시지가 없었습니다.

**클라이언트는 중단된 동안 전송에 실패한 모든 메시지에 대해 통보 받았습니다.**

---

<sup>4</sup> 순수한 마스터-슬레이브에 대한 설명서: <http://activemq.apache.org/pure-master-slave.html>

<sup>5</sup> 이 백서에서는 ActiveMQ의 수동 백업 및 복원 기능을 평가하지 않았습니다.

## 트랜잭션 무결성 및 관리

Edison은 두 가지 시나리오를 사용하여 WebSphere MQ 및 ActiveMQ의 기본 기능을 비교했습니다. 두 시나리오에서 모두 "저장 및 전달" 아키텍처를 사용하되, 첫 번째는 기본 API로 트랜잭션을 관리하고 두 번째는 WAS(WebSphere Application Server) v8.5.5를 사용했습니다. Whereas WebSphere MQ에는 데이터베이스 서버 간에 무결성을 보장하는 기본 트랜잭션 관리자가 있지만 ActiveMQ로 "저장 및 전달" 기능을 구현하려면 "브로커 네트워크"를 만들어야 했습니다(아래 "관리 및 운영" 섹션에서 설명).

또한 WebSphere MQ의 트랜잭션 관리자는 사용자가 콘솔 하나에서 해당 환경을 관리할 수 있지만, ActiveMQ의 "브로커 네트워크"는 별도의 콘솔이 필요하고 브로커별로 구성 파일을 편집해야 합니다. 아래의 "관리 및 운영" 섹션에서 설명하겠지만, 이런 이유 때문에 브로커 인스턴스가 수백에서 수천 개에 이르는 실 운영 환경은 고사하고 서버가 두 개 이상인 환경만 되어도 ActiveMQ 시스템을 관리하기가 훨씬 더 어렵습니다.

### 시나리오 1: 저장 및 전달 - 기본 WebSphere MQ/ActiveMQ API 사용

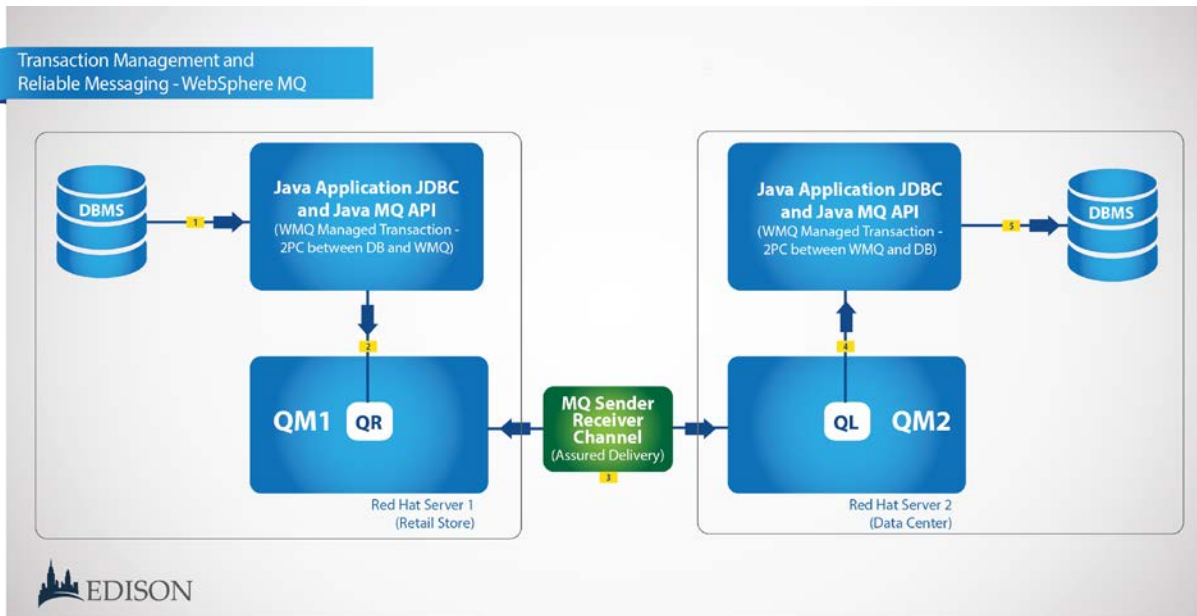


그림 3: 기본 WebSphere MQ 트랜잭션 관리자를 사용하여 저장 및 전달

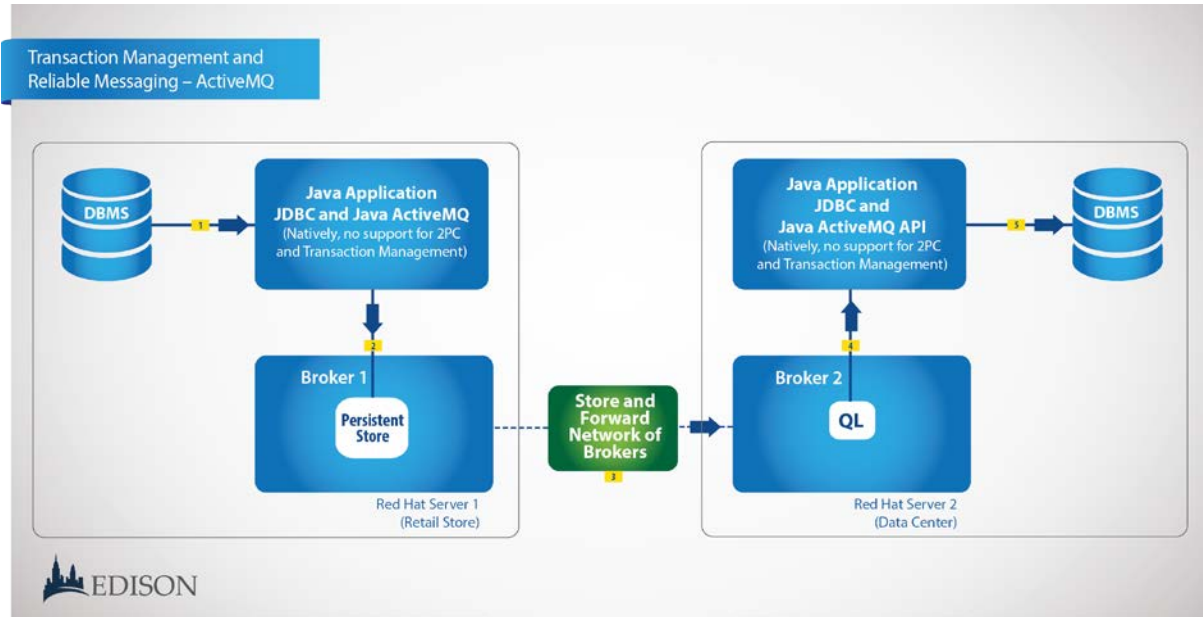


그림 4: ActiveMQ의 "브로커 네트워크"를 사용하여 저장 및 전달

첫 번째 시나리오에서 Edison은 WebSphere MQ의 기본 기능과 ActiveMQ의 "브로커 네트워크"를 사용했습니다. 그리고 일반적인 환경에서 트랜잭션을 실행하여 로드 밸런싱 기능을 테스트하기 위한 기준을 설정했습니다.

#### 시나리오 설명:

- 클라이언트(예: 소매 매장에서)는 소매 매장 데이터베이스의 데이터를 읽은 다음 큐에 "소매 매장 로컬 메시지"로 저장합니다. 시스템의 무결성을 보장하기 위해서는 이 과정을 반드시 XA 트랜잭션 제어(2PC - 2 단계 커밋) 환경에서 실행해야 합니다.
- 메시징 인프라 프로세스는 "소매 매장 서버로부터" 이 메시지를 가져와서(get) "데이터 센터"에 있는 다른 큐 매니저와 브로커에 전달(put)하며, 이때 둘 이상의 독립적 큐 매니저(또는 브로커)에 액세스하게 됩니다. 이 단계에서 메시지는 반드시 "한 번, 단 한 번만" 전달되어야 하며 메시지가 유실되거나 중복되어서는 안 됩니다.
- 데이터 센터에서 실행 중인 클라이언트 애플리케이션은 데이터 센터 큐의 메시지를 읽은 다음 데이터 센터에 있는 데이터베이스에 그 데이터를 기록합니다. 시스템의 무결성을 보장하기 위해 이 단계 역시 반드시 XA 트랜잭션 제어(2PC - 2 단계 커밋) 환경에서 처리해야 합니다.

WebSphere MQ의 기본 트랜잭션 관리자로는 데이터베이스 읽기/쓰기와 메시지 입력/가져오기(put/get)를 트랜잭션 하나로 조합할 수 있는 반면(2PC 사용), ActiveMQ에는 데이터베이스 및 브로커 작업의 트랜잭션 조합 기능이 없습니다. 위의 시나리오를 보면, WMQ의 트랜잭션 관리 기능은 데이터베이스와 큐 사이에서 데이터를 운반하는 "매장 내" 클라이언트와 "데이터 센터" 클라이언트 애플리케이션 양쪽 모두에 적용됩니다. 두 WMQ 큐 관리자(매장 내 QM 및 데이터 센터의 QM) 사이에서 데이터를 전송하는 프로세스는 원격 큐, XMIT 큐, 그리고 WMQ의 클라이언트 발신자/수신자

채널을 사용하여 처리되며, 이로써 메시지 전달의 신뢰성을 보장합니다. 매장 내 클라이언트 애플리케이션은 매장 내 QM 에 정의되어 있는 소위 "원격 큐"에 연결하여 메시지를 추가합니다. 따라서 매장과 데이터 센터 사이의 네트워크가 작동하지 않을 때도 클라이언트 애플리케이션은 작동할 수 있습니다. 모든 메시지는 매장 QM 의 이 원격 큐에 보관됩니다. 매장과 데이터 센터 사이의 네트워크가 연결되면 WMQ 가 메시지 처리 프로세스를 자동으로 수행하고, 모든 메시지는 매장 내 QM 에 있는 원격 큐에서 데이터 센터에 있는 로컬 큐로 안전하게 운반됩니다. 이 과정은 네트워크 또는 하드웨어 작동이 중단되더라도 메시지가 유실되거나 중복되지 않는 방식으로 이루어지며, 서비스 품질은 2PC 트랜잭션과 동일합니다. 메시지가 원격 큐에서 로컬 큐로 운반되고 나면, 데이터 센터의 클라이언트 애플리케이션은 이 메시지를 사용하거나 WMQ 2PC 가 제어하는 데이터 센터 데이터베이스에 메시지를 기록할 수 있습니다. 위의 설명으로 미루어 볼 때 매장 내 데이터베이스에서 데이터 센터 데이터베이스로 데이터를 운반하는 프로세스 전체가 신뢰성 있게 비동기적으로 이루어지며, 네트워크, 소프트웨어 또는 하드웨어 장애가 발생할 경우 롤백 및 복구된다는 것을 알 수 있습니다.

ActiveMQ 는 2PC 트랜잭션 관리자를 제공하지 않을 뿐더러 데이터베이스와 브로커 간의 2PC 트랜잭션을 조율하지도 못합니다.

이 시나리오에서는 원격 큐와 발신자/수신자 채널을 사용하는 WebSphere MQ의 분산 큐 처리<sup>6</sup> 및 ActiveMQ의 브로커 네트워크<sup>7</sup>가 서버 1 과 서버 2 간의 메시지 전달과 동일한 기능을 수행하는 것으로 묘사했습니다. 그러나 이 구성요소들은 구현 아키텍처가 다르며, 브로커 네트워크의 신뢰성에 대한 상세한 분석 테스트는 이 조사에서 수행하지 않았습니니다.

---

<sup>6</sup> WMQ 분산 큐 처리에 관한 추가 설명서:

[http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp?topic=%2Fcom.ibm.mq.con.doc%2Fq015280 .htm](http://pic.dhe.ibm.com/infocenter/wmqv7/v7r5/index.jsp?topic=%2Fcom.ibm.mq.con.doc%2Fq015280.htm)

<sup>7</sup> ActiveMQ 의 브로커 네트워크 및 분산 큐 처리에 관한 추가 설명서:

<http://activemq.apache.org/how-do-distributed-queues-work.html>

## 시나리오 2: WAS 를 트랜잭션 관리자로 사용하여 저장 및 전달

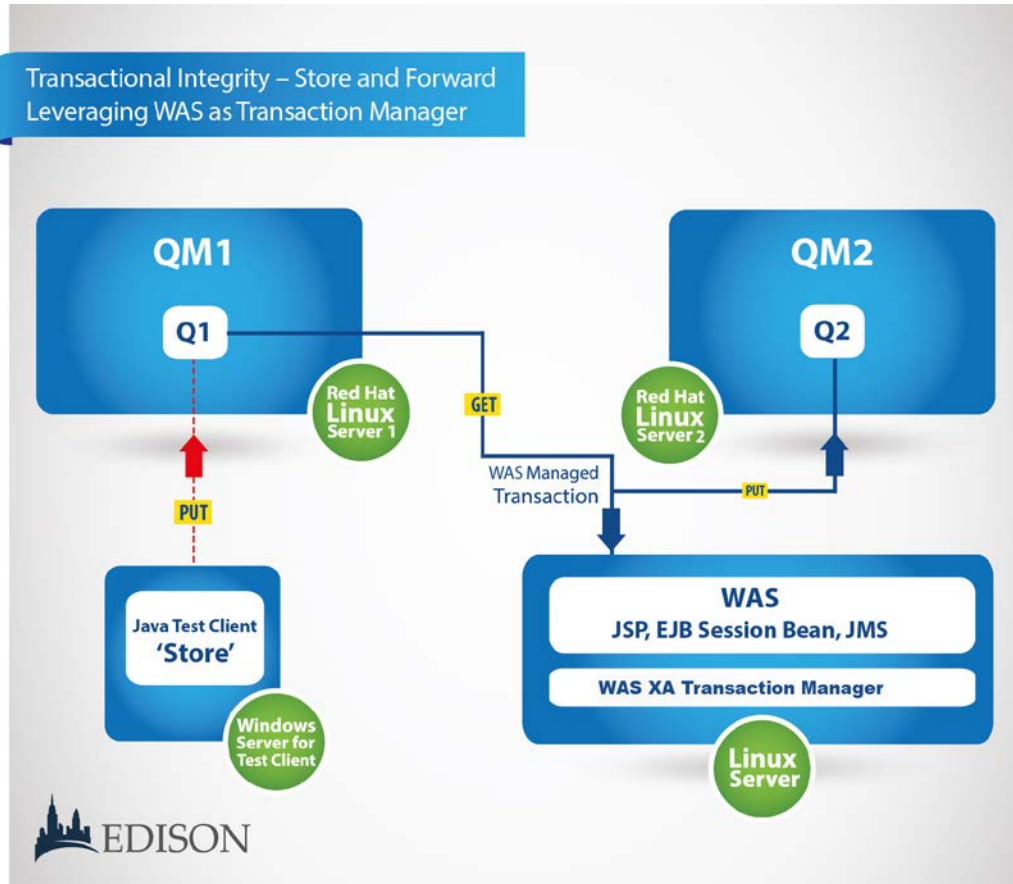


그림 5: 시나리오 2: WAS 를 트랜잭션 관리자로 사용하여 저장 및 전달

두 번째 시나리오에서도 Edison 은 동일한 기준군과 트랜잭션 장애군을 만들었으나 WAS(WebSphere Application Server)를 사용하여 트랜잭션을 관리했습니다.

### 시나리오 설명:

- 클라이언트(소매 매장 사례)는 서버 1 큐에 메시지를 저장합니다.
- WAS 서버(데이터 센터에서 호스팅)는 JMS 로 이 메시지를 처리하여 서버 1로부터 메시지를 받은 다음 서버 2의 큐로 보냅니다.
- 세션 빈(Beans) EJB 는 WAS 의 2PC 트랜잭션 제어에 따라 JMS 수신 및 JMS 송신을 모두 처리하여 전체 프로세스의 무결성을 보장했습니다.

두 시스템 모두 WAS(WebSphere Application Server)에서 관리했으며, 기준 조건과 두 가지 장애 조건 양쪽에서 메시지 유실 또는 중복이 발생하지 않았습니다.

## 성능

참고: 성능 테스트는 Edison Group 이 아니라 IBM 직원인 Roman Kharkovski 가 수행했습니다.

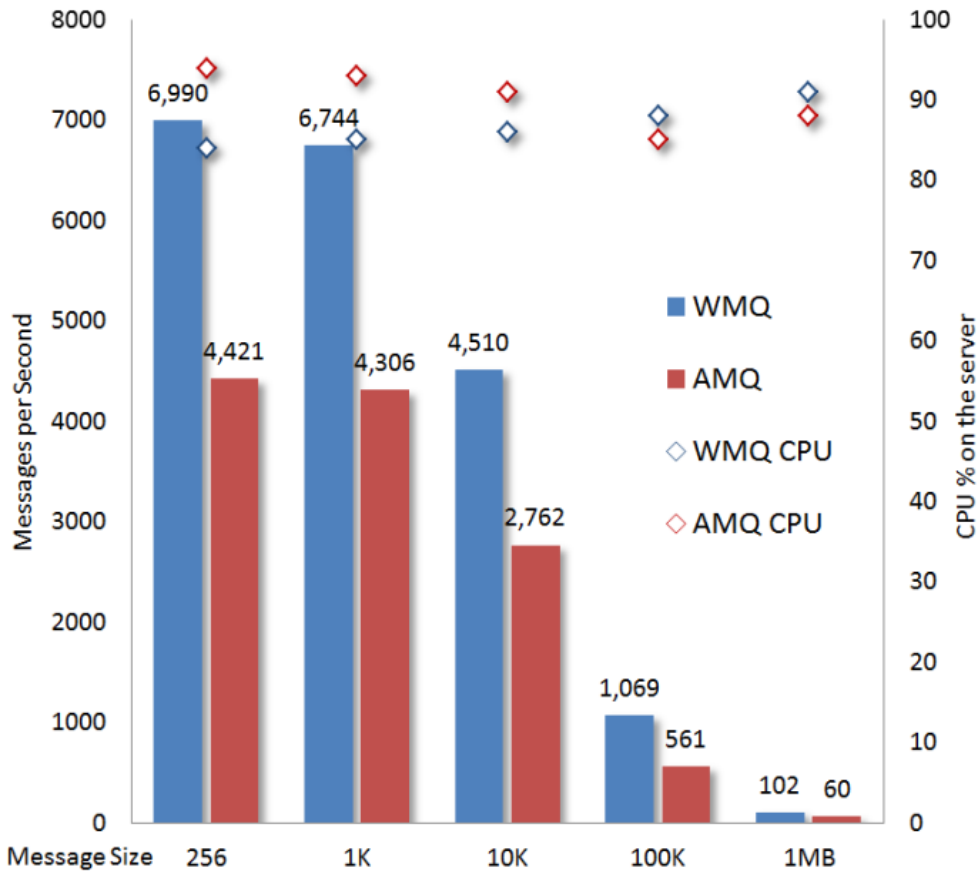


그림 6: IBM WebSphere MQ 대 ActiveMQ 성능 벤치마크 결과

지속적 메시징의 성능 테스트 결과는 WebSphere MQ 및 ActiveMQ의 반복 조정을 통해 얻었습니다. JMS를 위한 IBM 성능 벤치마크를 실행하고 WebSphere MQ 큐 관리자 인스턴스 네 개와 Apache ActiveMQ 브로커 네 개를 실행하는 클라이언트 동시 스레드 80 개에서 두 제품 모두 최고의 결과를 발휘했습니다. 각 인스턴스에 자체 SSD를 두고 여기에 데이터를 기록했습니다. 20분 테스트를 실행할 때마다 256 바이트, 1K, 10K, 100K 또는 1MB의 다섯 가지 메시지 크기 중 하나를 사용했습니다. 위의 측정치는 여섯 번

실행한 평균값입니다. 평균적으로, WebSphere MQ 가 ActiveMQ 에 비해 60 ~ 90% 더 빨랐습니다. 따라서 ActiveMQ 로 비슷한 결과를 얻으려면 다음이 필요합니다.

- 60 ~ 90% 더 많은 하드웨어 비용
- 60 ~ 90% 더 큰 데이터 센터 공간
- 60 ~ 90% 더 강한 쿨러
- 60 ~ 90% 더 강한 전원
- 60 ~ 90% 더 많은 소프트웨어 설치
- 전체 관리를 위한 60 ~ 90% 더 많은 운영 비용

이 테스트의 수행 방법과 결과에 대한 자세한 내용은 [왜 WebSphere 인가? 블로그](#)의 다음 게시물을 참조하십시오.

## 설치 및 구성

---

WebSphere MQ and ActiveMQ 모두 설치 절차는 매우 간단합니다. 두 제품 모두 기본 설치 몇 분 만에 완료되었고 반복해야 하는 단계 또는 오류는 거의 없었습니다.

WebSphere 는 [왜 WebSphere 인가? 블로그](#)에 공개된 Red Hat Linux 에 WMQ v7.5 를 설치하기 위한 스크립트 등의 맞춤 스크립트를 사용하여 클릭 한 번으로 60 초 안에 설치할 수 있습니다. ActiveMQ 의 경우에도 비슷한 스크립트를 작성할 수 있습니다.

그러나 WebSphere MQ 에는 다른 부분이 몇 가지 있었습니다.

WebSphere MQ와 ActiveMQ의 설치 과정 중 중요한 차이점 중 하나는 설명서의 제공 여부 및 정확성입니다. Edison은 두 제품 모두 성공적으로 설치할 수 있었으나, 새로 설치하는 것과 새 버전 또는 최신 버전을 설치하는 것의 주된 차이점 등이 설명된 IBM 설명서가 훨씬 더 자세하고 정확했습니다.<sup>8</sup>

반면 ActiveMQ 설명서는 완성도와 신뢰성이 한참 떨어졌습니다([Active MQ 5.0 구성 설명서](#)의 "해야 할 일" 태그 수를 보면 알 수 있듯이 눈에 띄게 부족한 부분이 많음)/ 예를 들어, Apache 설명서에는 Active MQ 를 설치할 때 시스템이 정상 작동 중이면 다음 메시지가 표시된다고 설명되어 있습니다.

```
"INFO ActiveMQ JMS Message Broker (ID:apple-s-Computer.local-51222-1140729837569-0:0) has started."
```

그러나 그 대신 다음 메시지가 나타났습니다.

```
"INFO: pidfile created : '/opt/media/apache-activemq-5.9.0/data/activemq-TTRH-vActiveMQ-2.pid' (pid '4046')."
```

완전한 설명서가 없기 때문에 이 ActiveMQ 인스턴스의 결과가 어떠했는지는 분명하지 않습니다. 더 중요한 것은 WebSphere MQ 는 "인증"을 받았으며 ActiveMQ 보다 훨씬 더 많은 종류의 운영 체제를 지원한다는 사실입니다. 부록에서 두 제품의 시스템 요구사항과 기능을 추가로 비교했습니다.

이 문제를 제외하면, 두 제품 모두 Red Hat Linux 환경에서 심각한 설치 문제는 발생하지 않았습니다.

---

<sup>8</sup> WebSphere MQ 설명서에는 시스템 운영자가 알아 두어야 하는 한 가지 문제가 누락되어 있었습니다. 즉, "사용자 mqm" 및 "그룹 mqm"이 없는 경우 설치 과정에서 자동으로 생성되지만, 수동으로 생성하지 않았기 때문에 설치 후 권한을 설정해야 합니다.



## 관리 및 운영

---

Windows 또는 Linux 의 커맨드라인 환경에 익숙한 사용자라면, Windows 또는 Linux 환경에서 명령 쉘로 ActiveMQ 및 WebSphere MQ 에 액세스할 수 있습니다. WebSphere MQ 명령 쉘에서는 메시지 최대 크기 변경, 메시지 지속성 설정, 기타 여러 가지 구성 설정 변경 등 다양한 기본 및 고급 운영 작업을 사용자가 수행할 수 있습니다. 반면 ActiveMQ 쉘의 경우, 사용자가 할 수 있는 것은 ActiveMQ 브로커를 시작하거나 중지하는 등 아주 간단한 운영 작업뿐입니다. 그 이상의 고급 절차를 수행하려면 사용자가 파일을 직접 편집해야 하며, Active MQ 는 대개의 경우 공통 메시징 속성의 구성을 지원하지 않습니다(즉, 메시지 최대 크기 변경 또는 큐 깊이 변경 등의 속성을 운영자가 유연하게 관리하는 대신 OS 리소스로 제어). 또한 두 시스템 모두 상당히 완성도 있는 설명서를 제공하지만, ActiveMQ 설명서에는 일부 누락된 부분이 있어 숙련된 고급 사용자를 제외한 사람들은 살펴보기가 어렵습니다. 한 시스템 운영자는 ActiveMQ 포럼에 이런 글을 남겼습니다.

*"ActiveMQ 개발자가 되려면 그쪽 소스 코드를 익숙하게 다뤄야 할 겁니다. 자세한 설명서가 있기는 하지만 정확하진 않구요, 세부 설명도 부족합니다. 거의 모든 클래스가 나와 있지만 모든 것을 다 설명하고 있지 않습니다. 그리고 코드 품질은... 그냥 Microsoft 표준과 다르다는 것만 말씀 드릴게요."*

반대로 [IBM 의 Infocenter](#) 및 [IBM Redbooks](#) 에서 제공하는 설명서는 완성도가 높고 상세하며, 거의 모든 구성 및 WebSphere MQ 사용자들이 마주칠 수 있는 운영 문제를 다루고 있습니다.

GUI(그래픽 사용자 인터페이스)를 기대하는 사용자라면 두 제품 사이에서 커맨드라인 사용자가 느끼는 것만큼이나 커다란 차이를 느끼게 됩니다. ActiveMQ GUI 콘솔에서는 서버가 한 개 있는 환경을 시각화하거나 새 큐를 만드는 것 같은 기본 기능만 지원합니다. 메시지 최대 크기 설정, 보안 관리, 지속성 정의 등 그 밖의 모든 작업은 수행할 수 없거나, 사용자가 커맨드라인 및 파일 시스템에서 AMQ 구성 파일을 편집하거나, 프로그래밍을 통해 처리해야 합니다. ActiveMQ 는 첫 번째 제공자 또는 소비자가 대상에 연결할 때 필요에 따라 동적으로 큐를 만듭니다. 따라서 ActiveMQ 에는 큐를 만들기 위한 커맨드라인 기능이 없습니다. XML 구성 파일에서 브로커용 큐를 정의할 수는 있으나, 그러려면 네트워크의 브로커 인스턴스마다 XML 파일을 수동으로 편집해야 합니다.

또한 WebSphere MQ Explorer 의 경우, 운영자가 한 콘솔에서 수많은 브로커를 변경할 수 있습니다. ActiveMQ 에는 브로커마다 고유한 URL 이 있습니다. 브로커가 100 개 있는

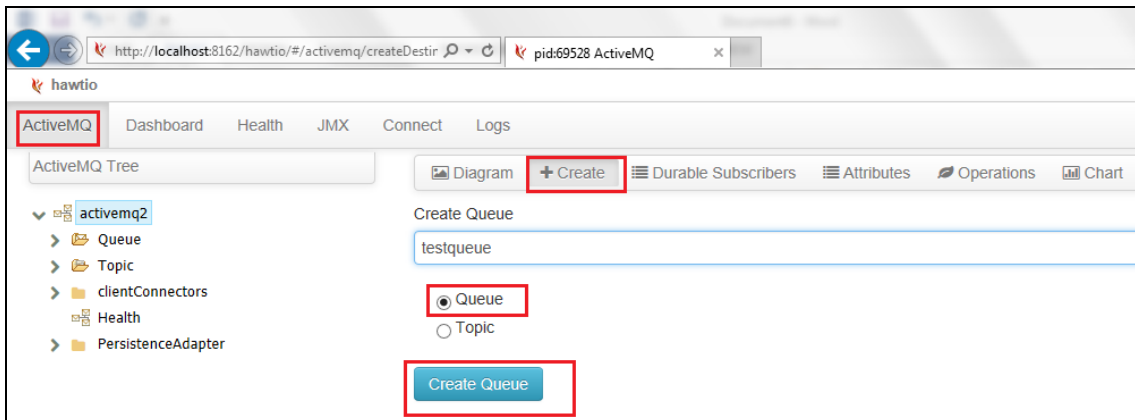
환경이라면 100 개의 URL 을 열어서 확인해야 각 브로커에 액세스할 수 있습니다. 게다가 이것은 마스터 브로커에만 해당하는 이야기입니다. *ActiveMQ 슬레이브 브로커에는 운영 콘솔마저 없으며, 오직 커맨드라인과 XML 구성으로만 관리할 수 있습니다.*

WebSphere MQ Explorer 인터페이스는 ActiveMQ 와 달리 사용하기 도 쉽고 사용 환경 전체를 보여 줍니다. 이와 함께 MQ 환경의 거의 모든 측면(특히 큐 관리자, 큐, 채널, 리스너, 주제, 간행물 및 구독 등)에 대한 상세 특성을 사용자가 관리, 조정 및 운영할 수 있는 기능도 제공합니다.

두 시스템의 강점과 한계를 명확한 사례로 제시하기 위해 Edison 은 큐 만들기, 공유 설정, 큐 정의 지정, 다중 큐 관리 등 ActiveMQ 및 WebSphere MQ 의 기본 기능을 단계별 프로세스로 설명합니다.

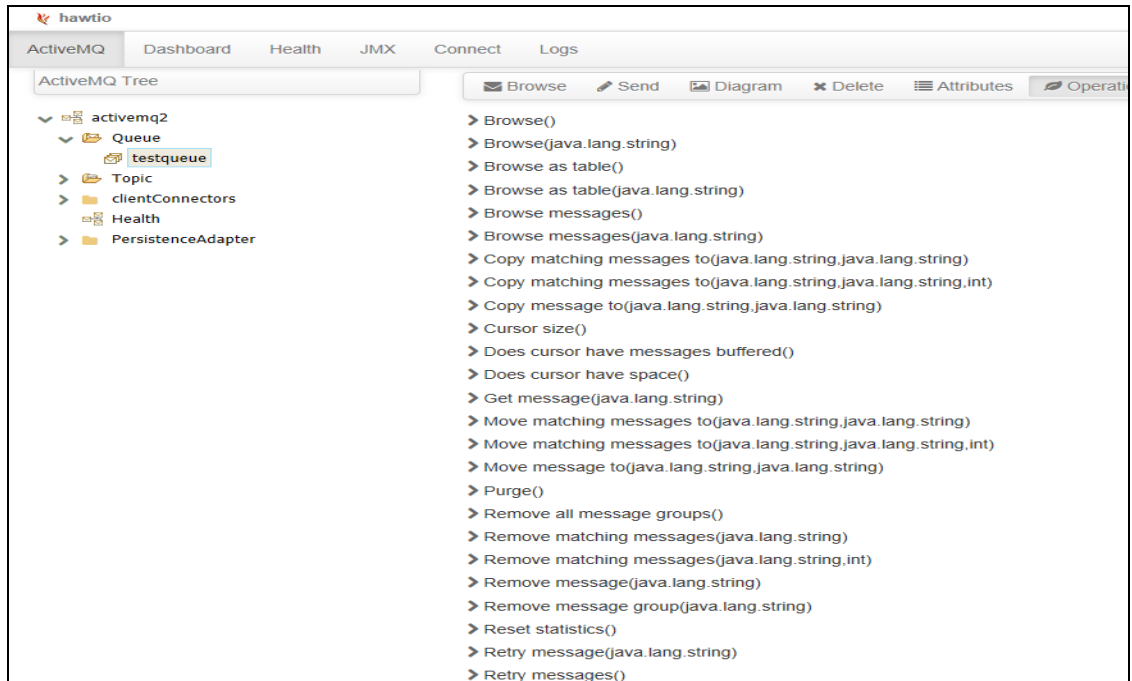
### ActiveMQ Hawtio 콘솔을 통해 큐 만들기

- hawtio 콘솔(<http://host:port/hawtio>)에 로그인
- "ActiveMQ" 클릭
- "Create(만들기)" 클릭
- "Queues(큐)" 클릭
- 큐 이름 입력 후 "Create Queue(큐 만들기)" 클릭



참고: 큐는 파일 시스템/데이터베이스에 저장됩니다. 사용자는 처음 시작할 때 만들고자 하는 큐의 XML 을 구성한 다음 이를 계속 유지할 수 있습니다. 큐가 그 자리에 없으면 다음 번에 프로그램이 액세스하려고 할 때 다시 만듭니다.

### JMX 세션 및 JNDI 를 통해 동적으로 큐 만들기



The screenshot shows the Hawtio interface for ActiveMQ. The 'ActiveMQ Tree' on the left has 'testqueue' selected under 'activemq2 > Queue'. The right pane lists the following JMX operations:

- > Browse()
- > Browse(java.lang.string)
- > Browse as table()
- > Browse as table(java.lang.string)
- > Browse messages()
- > Browse messages(java.lang.string)
- > Copy matching messages to(java.lang.string,java.lang.string,java.lang.string,int)
- > Copy matching messages to(java.lang.string,java.lang.string,int)
- > Copy message to(java.lang.string,java.lang.string)
- > Cursor size()
- > Does cursor have messages buffered()
- > Does cursor have space()
- > Get message(java.lang.string)
- > Move matching messages to(java.lang.string,java.lang.string,java.lang.string,int)
- > Move matching messages to(java.lang.string,java.lang.string,int)
- > Move message to(java.lang.string,java.lang.string)
- > Purge()
- > Remove all message groups()
- > Remove matching messages(java.lang.string)
- > Remove matching messages(java.lang.string,int)
- > Remove message(java.lang.string)
- > Remove message group(java.lang.string)
- > Reset statistics()
- > Retry message(java.lang.string)
- > Retry messages()

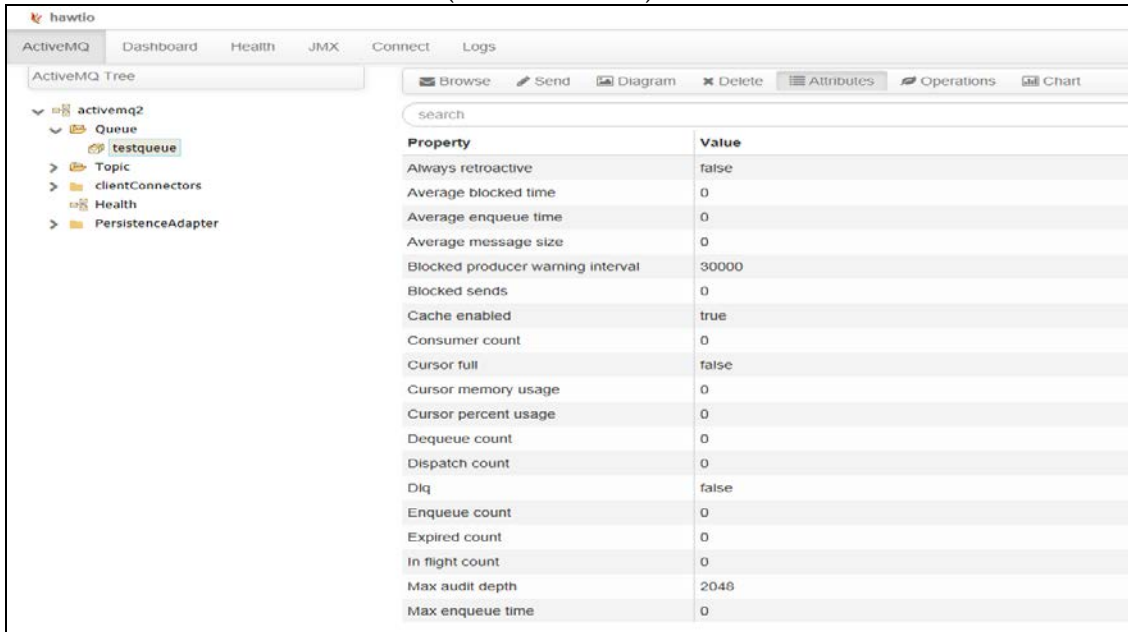


The continuation of the JMX operations list:

- > Remove all message groups()
- > Remove matching messages(java.lang.string)
- > Remove matching messages(java.lang.string,int)
- > Remove message(java.lang.string)
- > Remove message group(java.lang.string)
- > Reset statistics()
- > Retry message(java.lang.string)
- > Retry messages()
- > Send text message(java.lang.string)
- > Send text message(java.lang.string,java.lang.string,java.lang.string)
- > Send text message(java.util.map,java.lang.string)
- > Send text message(java.util.map,java.lang.string,java.lang.string,java.lang.string)
- > Send text message with properties(java.lang.string)

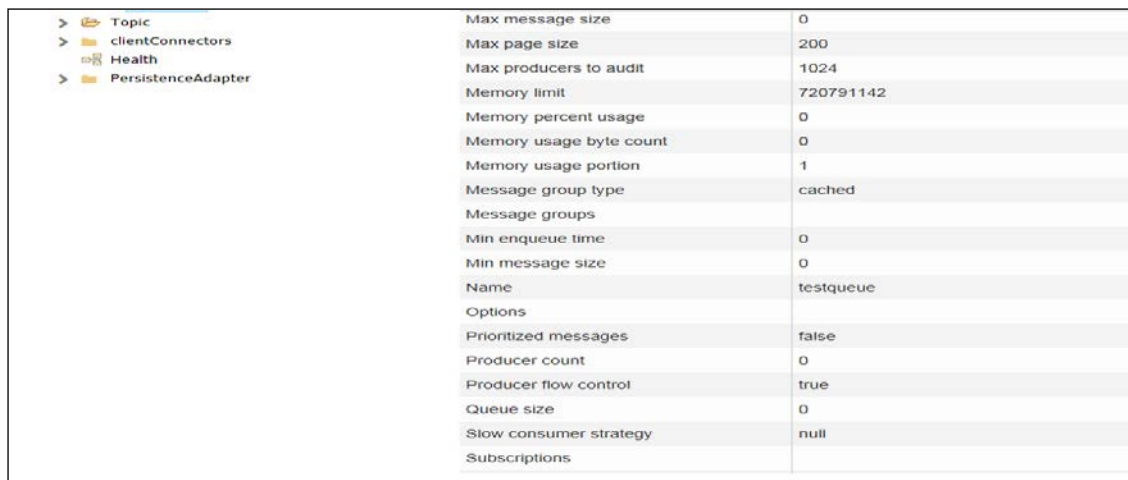
위의 콘솔을 사용하여 큐를 만들거나 동적으로 큐가 만들어지고 나면, 사용자는 그 큐에서 메시지 탐색, 삭제, 보내기, 소거(큐 지우기) 등 다양한 작업을 수행할 수 있습니다.

사용자는 특성을 볼 수만 있습니다(읽기 전용 모드).



The screenshot shows the hawtio interface for an ActiveMQ queue named 'testqueue'. The left sidebar shows the 'ActiveMQ Tree' with 'testqueue' selected. The main area displays a table of properties and their values.

Property	Value
Always retroactive	false
Average blocked time	0
Average enqueue time	0
Average message size	0
Blocked producer warning interval	30000
Blocked sends	0
Cache enabled	true
Consumer count	0
Cursor full	false
Cursor memory usage	0
Cursor percent usage	0
Dequeue count	0
Dispatch count	0
Diq	false
Enqueue count	0
Expired count	0
In flight count	0
Max audit depth	2048
Max enqueue time	0

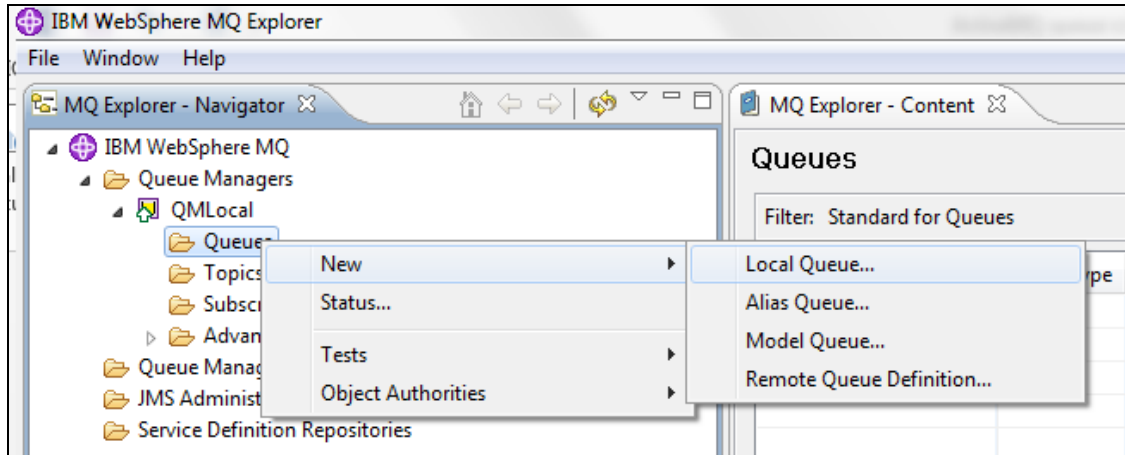


This screenshot shows the continuation of the 'testqueue' properties table from the previous screenshot.

Max message size	0
Max page size	200
Max producers to audit	1024
Memory limit	720791142
Memory percent usage	0
Memory usage byte count	0
Memory usage portion	1
Message group type	cached
Message groups	
Min enqueue time	0
Min message size	0
Name	testqueue
Options	
Prioritized messages	false
Producer count	0
Producer flow control	true
Queue size	0
Slow consumer strategy	null
Subscriptions	

### WebSphere MQ Explorer 를 통해 큐 만들기

- MQ Explorer - Navigator(탐색기)에서 Queues(큐) 선택
- "New(새로 만들기)" 선택
- "Local Queue...(로컬 큐)" 클릭



MQ Explorer 는 다음과 같이 ActiveMQ 웹 콘솔에는 없는 보다 유용한 큐 특성과 사용자 정의 옵션을 제공합니다.

- Put/Get 권한
- 메시지 지속
- 공유 가능
- 클러스터링
- 트리거링
- 메시지 크기 및 기타 다수(아래 스크린샷 참조)

대개의 경우, ActiveMQ 콘솔에 없는 기능은 커맨드라인에서 실행할 수 있습니다. 그러나 브로커가 여러 개인 환경에서 고유한 메시징 포트를 만들 때 같이 운영자가 *xml 파일*을 직접 편집해야 하는 경우도 많습니다(아래 "다중 큐 관리" 참조).

## WMQ 에서 큐 속성 관리

⊕ New Local Queue
\_ □ ×

### Change properties

Change the properties of the new Local Queue

- General
- Extended
- Cluster
- Triggering
- Events
- Storage
- Statistics

#### General

Queue name:

Queue type:

Description:

Put messages:

Get messages:

Default priority:

Default persistence:

Scope:

Usage:

- General
- Extended
- Cluster
- Triggering
- Events
- Storage
- Statistics

#### Extended

Max queue depth:

Max message length:

Shareability:

Default input open option:

Message delivery sequence:

Retention interval:

Definition type:

Distribution lists:

Default read ahead:

Default put response type:

Property control:

<ul style="list-style-type: none"> <li>General</li> <li>Extended</li> <li><b>Cluster</b></li> <li>Triggering</li> <li>Events</li> <li>Storage</li> <li>Statistics</li> </ul>	<h3>Cluster</h3> <p>Sharing in Clusters</p> <p> <input checked="" type="radio"/> Not shared in a cluster  <input type="radio"/> Shared in cluster <input type="text"/>  <input type="radio"/> Shared in a list of clusters <input type="text"/> </p> <p>Default bind type: <input type="text" value="On open"/></p> <p>CLWL queue rank: <input type="text" value="0"/></p> <p>CLWL queue priority: <input type="text" value="0"/></p> <p>CLWL use queue: <input type="text" value="Queue manager"/></p>
<ul style="list-style-type: none"> <li>General</li> <li>Extended</li> <li>Cluster</li> <li><b>Triggering</b></li> <li>Events</li> <li>Storage</li> <li>Statistics</li> </ul>	<h3>Triggering</h3> <p>Trigger control: <input type="text" value="Off"/></p> <p>Trigger type: <input type="text" value="First"/></p> <p>Trigger depth: <input type="text" value="1"/></p> <p>Trigger message priority: <input type="text" value="0"/></p> <p>Trigger data: <input type="text"/></p> <p>Initiation queue: <input type="text"/> <input type="button" value="Select..."/></p> <p>Process name: <input type="text"/></p>
<ul style="list-style-type: none"> <li>General</li> <li>Extended</li> <li>Cluster</li> <li>Triggering</li> <li><b>Events</b></li> <li>Storage</li> <li>Statistics</li> </ul>	<h3>Events</h3> <p>Queue depth max events: <input type="text" value="Enabled"/></p> <p>Queue depth high events: <input type="text" value="Disabled"/></p> <p>Queue depth high limit: <input type="text" value="80"/></p> <p>Queue depth low events: <input type="text" value="Disabled"/></p> <p>Queue depth low limit: <input type="text" value="20"/></p> <p>Queue service interval events: <input type="text" value="None"/></p> <p>Queue service interval: <input type="text" value="999999999"/></p>

<ul style="list-style-type: none"> <li>General</li> <li>Extended</li> <li>Cluster</li> <li>Triggering</li> <li>Events</li> <li><b>Storage</b></li> <li>Statistics</li> </ul>	<h3>Storage</h3> <p>Backout requeue queue: <input type="text"/> <span>Select...</span></p> <p>Backout threshold: <input type="text" value="0"/></p> <p>Harden get backout: <input type="text" value="Hardened"/></p> <p>NPM class: <input type="text" value="Normal"/></p>
<ul style="list-style-type: none"> <li>General</li> <li><b>Extended</b></li> <li>Cluster</li> <li>Triggering</li> <li>Events</li> <li>Storage</li> <li>Statistics</li> </ul>	<h3>Statistics</h3> <p>Queue monitoring: <input type="text" value="Queue Manager"/></p> <p>Queue statistics: <input type="text" value="Queue Manager"/></p> <p>Queue accounting: <input type="text" value="Queue Manager"/></p>

### runmqsc(WMQ 커맨드라인)로 큐 만들기

참고: ActiveMQ는 이 기능을 지원하지 않습니다. 운영자가 파일 시스템에서 파일을 직접 수정해야 합니다.

- WMQ에서 제공되는 커맨드라인 스크립트는 다음과 같이 사용합니다.

WMQ에서 큐를 만드는 커맨드라인:

```
DEFINE QLOCAL<queue>
  8 : DEFINE QLOCAL<queue>
AMQ8006: WebSphere MQ queue created.
```

WMQ에서 큐를 만들 때 사용할 수 있는 큐 특성:

```
DEFINE QLOCAL< q_name >
[ BOQNAME< string > ]
[ CLUSNL< namelist_name > ]
[ DEFBIND< NOTFIXED ! OPEN > ]
[ DEFPRTY< integer > ]
[ DEFREADA< NO ! YES ! DISABLED > ]
[ DEFSOPT< EXCL ! SHARED > ]
[ GET< ENABLED ! DISABLED > ]
[ LIKE< qlocal_name > ]
[ MAXMSGL< integer > ]
[ HARDENBO ! NOHARDENBO ]
[ SHARE ! NOSHARE ]
[ PROCESS< string > ]
[ PROPTLC< COMPAT ! NONE ! ALL ! FORCE > ]
[ QDEPTHHI< integer > ]
[ QDPHIEU< ENABLED ! DISABLED > ]
[ QDPMAXEU< ENABLED ! DISABLED > ]
[ QSUCINT< integer > ]
[ SCOPE< QMGR ! CELL > ]
[ TRIGDPTH< integer > ]
[ TRIGTYPE< FIRST ! EVERY ! DEPTH ! NONE > ]
[ USAGE< NORMAL ! XMITQ > ]
[ STATQC< QMGR ! ON ! OFF > ]
[ MONQC< OFF ! QMGR ! LOW ! MEDIUM ! HIGH > ]
[ CLWLRANK< integer > ]
[ CLWLUSEQC< LOCAL ! ANY ! QMGR > ]
[ BOTHRESH< integer > ]
[ CLUSTER< cluster_name > ]
[ DEFPRESP< SYNC ! ASYNC > ]
[ DEFPSIST< YES ! NO > ]
[ DESCR< string > ]
[ DISTL< YES ! NO > ]
[ INITQ< string > ]
[ MAXDEPTH< integer > ]
[ MSGDLUSQ< PRIORITY ! FIFO > ]
[ REPLACE ! NOREPLACE ]
[ TRIGGER ! NOTRIGGER ]
[ PUT< ENABLED ! DISABLED > ]
[ QDEPTHLO< integer > ]
[ QDPLOEU< ENABLED ! DISABLED > ]
[ QSUCIEU< NONE ! HIGH ! OK > ]
[ RETINTUL< integer > ]
[ TRIGDATA< string > ]
[ TRIGMPRI< integer > ]
[ NPMCLASS< NORMAL ! HIGH > ]
[ ACCTQC< QMGR ! ON ! OFF > ]
[ CLWLPRTY< integer > ]
```



WMQ 에서 큐를 수정할 때 사용할 수 있는 큐  
특성:

```

ALTER QLOCAL< q_name >
[ BOQNAME< string > ]
[ CLUSNL< namelist_name > ]
[ DEFBIND< NOTFIXED ; OPEN > ]
[ DEFPRTY< integer > ]
[ DEFREADA< NO ; YES ; DISABLED > ]
[ DEFSOPT< EXCL ; SHARED > ]
[ GET< ENABLED ; DISABLED > ]
[ INITQ< string > ]
[ MAXMSGL< integer > ]
[ NPMCLASS< NORMAL ; HIGH > ]
[ PROPCTL< COMPAT ; NONE ; ALL ; FORCE > ]
[ PUT< ENABLED ; DISABLED > ]
[ QDEPTHLO< integer > ]
[ QDPLOEU< ENABLED ; DISABLED > ]
[ QSUCIEU< NONE ; HIGH ; OK > ]
[ RETINTUL< integer > ]
[ TRIGDATA< string > ]
[ TRIGGER ; NOTRIGGER ]
[ TRIGTYPE< FIRST ; EVERY ; DEPTH ; NONE > ]
[ USAGE< NORMAL ; XMITQ > ]
[ MONQ< QMGR ; OFF ; LOW ; MEDIUM ; HIGH > ]
[ SHARE ; NOSHARE ]
[ CLWLRANK< integer > ]
[ CLWLUSEQ< LOCAL ; ANY ; QMGR > ]
[ BOTHRESH< integer > ]
[ CLUSTER< cluster_name > ]
[ DEFPRESP< SYNC ; ASYNC > ]
[ DEFPERSIST< YES ; NO > ]
[ DESCR< string > ]
[ DISTL< YES ; NO > ]
[ HARDENBO ; NOHARDENBO ]
[ MAXDEPTH< integer > ]
[ MSGDLUSQ< PRIORITY ; FIFO > ]
[ PROCESS< string > ]
[ QDEPTHHI< integer > ]
[ QDPHIEU< ENABLED ; DISABLED > ]
[ QDPMAXEU< ENABLED ; DISABLED > ]
[ QSUCINT< integer > ]
[ SCOPE< QMGR ; CELL > ]
[ TRIGDPH< integer > ]
[ TRIGMPRI< integer > ]
[ ACCTQ< QMGR ; ON ; OFF > ]
[ STATQ< QMGR ; ON ; OFF > ]
[ CLWLPRTY< integer > ]
[ FORCE ]

```

WMQ 에서 큐를 지우는 커맨드라인:

```

CLEAR QLOCAL<QUEUE>
16 : CLEAR QLOCAL<QUEUE>
AMQ8022: WebSphere MQ queue cleared.

```

WMQ 에서 큐를 삭제하는 커맨드라인:

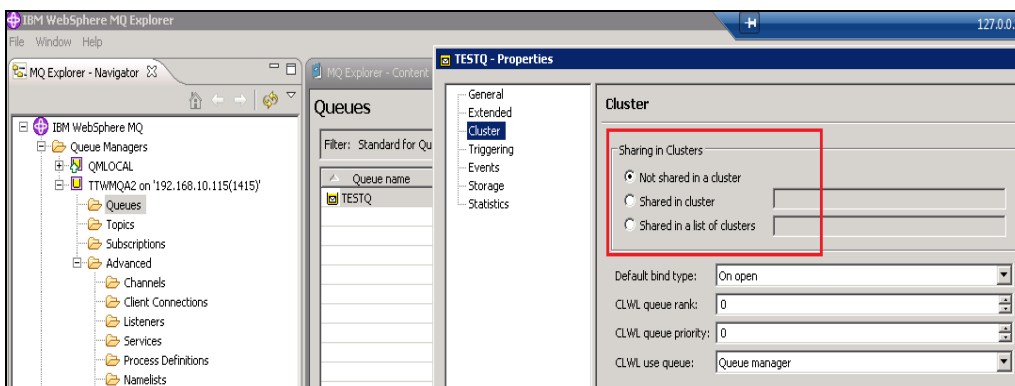
```

DELETE QLOCAL< q_name >
[ PURGE ; NOPURGE ]

```

### 큐 공유 설정

WebSphere MQ 에서는 클러스터에 있는 큐의 공유 여부를 손쉽게 정의할 수 있습니다.



ActiveMQ의 경우, 클러스터에 있는 두 브로커가 서로 연결되면 브로커의 모든 큐가 기본적으로 공유됩니다. *이 기본 설정은 변경할 수 없습니다.* WebSphere MQ 기능의 장점은 두 배입니다.

- 각 클러스터에서 공유할 큐를 지정하기가 더 쉬움
- 공유된 클러스터는 별도의 설정 없이도 브로커 간 로드 밸런싱 적용

반면 ActiveMQ에서는 "브로커 네트워크"를 구현하기 위해 운영자가 수동으로 구성 파일을 편집해야 합니다. 이 옵션에 대한 설명서는 매우 드물지만, Edison은 다음 단계에 따라 "브로커 네트워크"를 구현하는 데 성공했습니다.

*브로커 네트워크 설정 단계(예: 메시지를 전달하려면 원격 브로커에 소비자 필요)*

- 로컬 브로커 구성 xml 파일(/apache-activemq-5.9.0/conf/activemq.xml)로 이동
- 브로커에 다음 코드 추가:

```
<networkConnectors>

<networkConnector
uri="static://(tcp://remotehost1:61616,tcp://remotehost2:61616)"/>

</networkConnectors>
```

여기서 61616은 양쪽 원격 호스트에 있는 가공나선 전송 커넥터입니다.

- remotehost1 및 remotehost2 브로커 둘 다 시작
- 로컬 브로커 시작

*순수 브로커 네트워크 설정 단계(예: 메시지를 전달하려면 원격 브로커에 소비자 필요)<sup>9</sup>*

- 로컬 브로커 구성 xml 파일(/apache-activemq-5.9.0/conf/activemq.xml)로 이동
- 브로커에 다음 코드 추가:

```
<networkConnectors>
```

---

<sup>9</sup> Edison은 이 백서에서 브로커 네트워크의 신뢰성 또는 확장성을 평가하거나 WebSphere MQ 클러스터링과 ActiveMQ 브로커 네트워크의 기술적 특징, 아키텍처 및 기능을 자세히 비교하지 않았습니다.

```

<networkConnector
uri="static://(tcp://remotehost1:61616,tcp://remotehost2:61616)"
staticBridge="true">

<staticallyIncludedDestinations>

<queue physicalName="queueName"/>

</staticallyIncludedDestinations>

</networkConnector>

</networkConnectors>

```

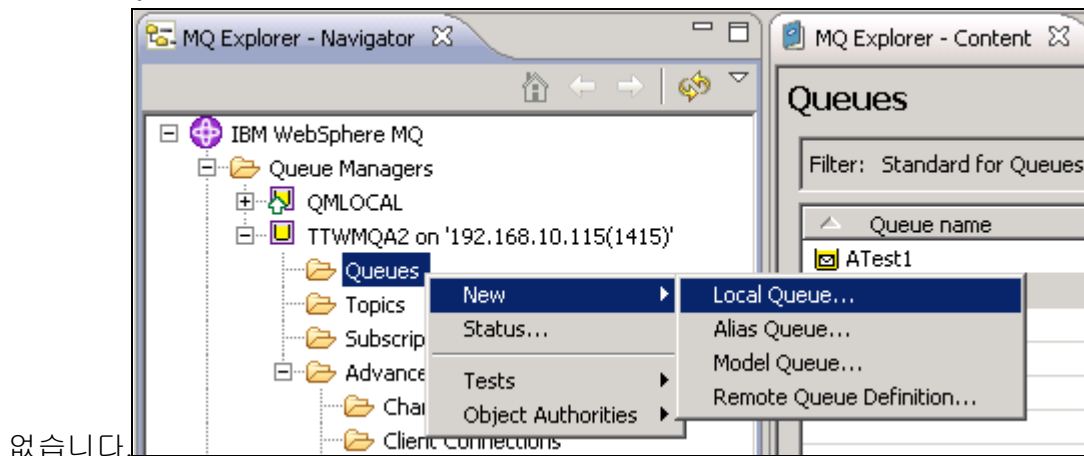
여기서 61616 은 양쪽 원격 호스트에 있는 가공나선 전송 커넥터입니다.

- remotehost1 및 remotehost2 브로커 둘 다 시작
- 로컬 브로커 시작

로컬 호스트 브로커로 메시지를 보내는 간단한 테스트를 통해 메시지에 로드 밸런싱이 적용되며 양쪽 원격 브로커에 모두 전달됨을 알 수 있었습니다.

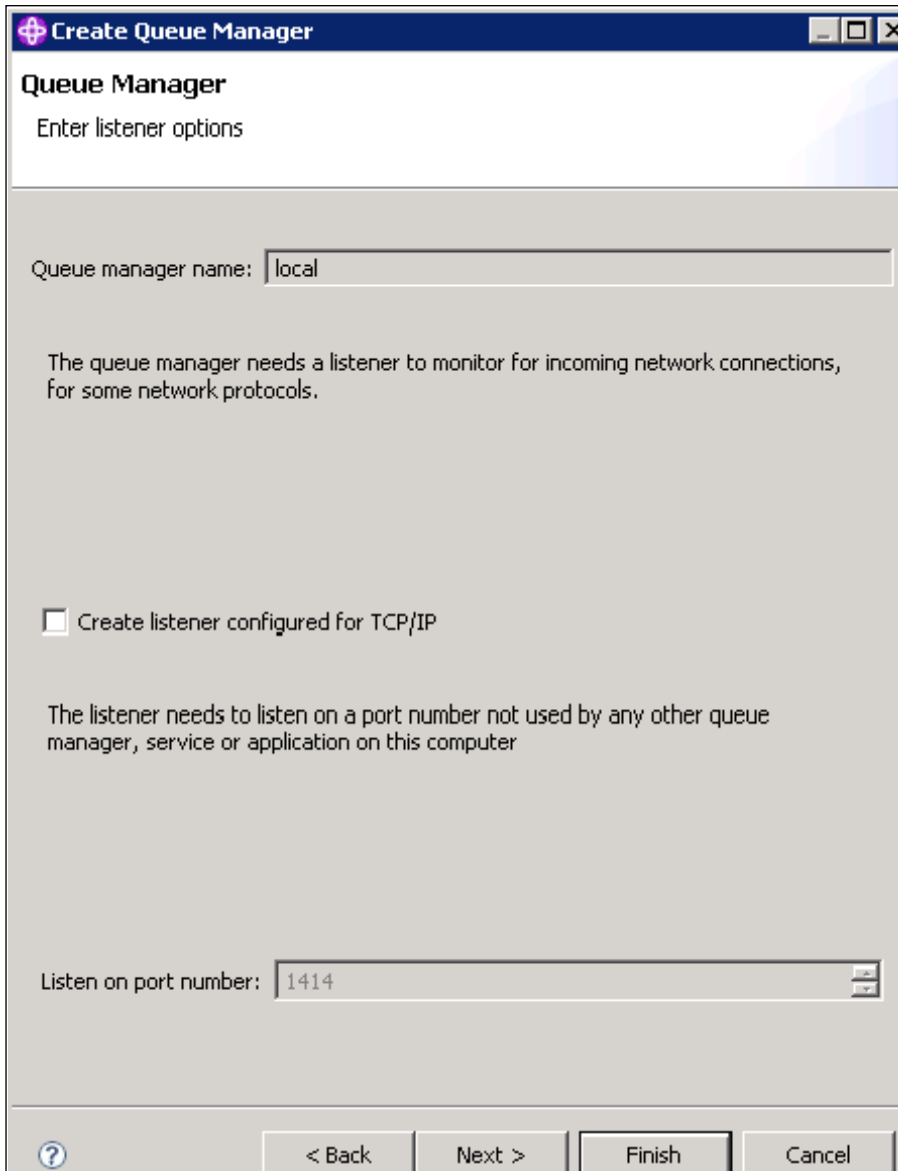
### 큐 정의 지정

WebSphere MQ 에서는 로컬, 원격, 별칭 및 모델 큐를 정의할 수 있습니다.  
ActiveMQ 에는 이 옵션이



## 다중 큐 관리

WebSphere MQ 에서는 포트가 없어도 로컬 큐 관리자를 여러 개 만들 수 있습니다. 그러나 수신되는 네트워크 연결 요청을 모니터하기 위한 리스너 포트는 필요합니다.



**Create Queue Manager**

**Queue Manager**  
Enter listener options

Queue manager name: local

The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

Create listener configured for TCP/IP

The listener needs to listen on a port number not used by any other queue manager, service or application on this computer

Listen on port number: 1414

? < Back Next > Finish Cancel

ActiveMQ 의 경우, 전송 커넥터의 포트(가공나선, amqp, stomp, mqtt 및 ws)가 동일 호스트에 있는 모든 브로커에 대해 고유해야 동시 실행이 가능합니다. activemq.xml 파일에서 이 부분을 수동으로 편집해야 합니다. 브로커 수가 그리 많지 않은 실 운영

환경이라 해도, activemq.xml 파일을 수동으로 편집하는 것은 시간이 오래 걸릴 뿐 아니라 변경 기록 및 표준 설명서가 없기 때문에 확장하는 것이 전적으로 불가능합니다.

```
<transportConnectors>

<!-- DOS protection, limit concurrent connections to 1000 and frame
size to 100MB -->

<transportConnector name="openwire"
uri="tcp://0.0.0.0:62001?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

<transportConnector name="amqp"
uri="amqp://0.0.0.0:5201?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

<transportConnector name="stomp"
uri="stomp://0.0.0.0:62011?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

<transportConnector name="mqtt"
uri="mqtt://0.0.0.0:2001?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

<transportConnector name="ws"
uri="ws://0.0.0.0:61611?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

</transportConnectors>
```

또한 실행하려는 브로커 웹 콘솔마다 고유한 포트가 있어야 합니다. 이 설정은 jetty.xml 파일에서 변경합니다.

```
<bean id="jettyPort" class="org.apache.activemq.web.WebConsolePort"
init-method="start">

<!-- the default port number for the web console -->

<property name="port" value="8168"/>

</bean>
```

## 부록 1: 기능 비교

아래 표 <sup>10</sup>에서는 IBM WebSphere MQ 7.5 와 Apache ActiveMQ 5.9 에서 주요 기능을 사용할 수 있는지 비교해 봅니다. 사용할 수 있는 차선책이나 타사 솔루션이 있는 경우, 각 기능의 참고 사항으로 나열했습니다.

기능	WebSphere MQ	ActiveMQ
<b>메시지 전송</b>		
JMS 1.1, JMS 2.0	JMS 1.1 정식 지원  JMS 2.0 은 WebSphere MQ v8 에서 정식 지원	JMS 1.1 만 지원
AMQP(고급 메시징 큐 프로토콜) 지원	AMQP 구성요소를 통해 JMS 구성요소를 외부로 전달하는 식으로 브리지에서 지원	AQMP 는 v5.8 이상에서 지원
Java, C++/C#, PHP 클라이언트	지원됨	지원됨
관리 대상 파일 전송	WebSphere MQ MFT 를 통해 유료 제공	제공하지 않음
문제 해결	구성요소에 대한 진단 오류 메시지	한정된 정보 및 설명서
<b>서비스 품질</b>		
페일오버	유실되거나 중복된 메시지 없음	메시지가 유실 및 중복되는 경우 많음
고가용성	입증된 고가용성	마스터-슬레이브 구성에 여러 가지 문제가 있으며, 네트워크 중단으로 인해 두 마스터가 경쟁하는 경우도 있음

<sup>10</sup> 이 표는 IBM WebSphere 경쟁력 팀의 도움을 받아 작성한 것입니다.

기능	WebSphere MQ	ActiveMQ
TMgr(트랜잭션 관리자)	제공됨(QM 과 DBMS 간 2PC)	타사 필요
XA 리소스 역할	외부 트랜잭션 관리자로 관리 가능	외부 트랜잭션 관리자로 관리 가능
성능	입증된 "동급 최강"	퍼시스턴트 메시지의 경우 WebSphere MQ 보다 60% ~ 90% 느림
<b>운영</b>		
관리 GUI	WebSphere MQ Explorer 는 기능이 매우 풍부	제한적인 기능 종종 파일 편집 필요
관리 CLI	풍부한 관리 명령 세트	몇 가지 명령은 제공되나 종종 파일 편집 필요
관리 API	풍부한 관리용 API	한정된 JMX 빈(bean) 세트 사용 가능
한 패널 관리	운영자가 콘솔 하나 또는 커맨드라인 하나로 모든 서버 관리 가능	각 서버를 개별 관리해야 함(서버가 100 개면 콘솔도 100 개)
배포 패턴	IBM Smart Cloud Orchestrator, IBM PureApplication System, IBM SoftLayer, IBM BlueMix PaaS 에서 제공	타사 필요
<b>보안</b>		
메시지 암호화	메시지 수준의 고급 보안 및 데이터 암호화(WMQ AMS 애드온 필요)	맞춤 프로그래밍 필요
감사 및 기록	대부분 제공되나 몇 가지 운영자 작업 필요	파일 편집 작업을 감사하지 않음

기능	WebSphere MQ	ActiveMQ
하트블리드(Heartbleed) 버그	WebSphere MQ 에 영향 없음	ActiveMQ 는 개방형 SSL 에 의존하므로 영향 있음
인증 및 권한 부여	기본 제공 기능	기본 제공 기능
<b>텔레메트리(MQTT)</b>		
<b>MQTT 지원</b>	IBM 은 MQTT 프로토콜의 공동 개발자. MQ 는 자체 클라이언트 및 브로커가 있으며, Eclipse PAHO 프로젝트는 물론 Mosquitto 등 다른 오픈 소스 프로젝트도 지원	v5.6 이상에서 MQTT 를 지원하나 보고된 버그가 많음.  Apollo 버전에서 MQTT 플러그인 구현
모바일 메시징	모바일 애플리케이션 및 기타 디바이스용 애플리케이션을 개발할 수 있는 모바일 및 디바이스 메시징 클라이언트 팩  IBM Worklight 와 통합됨	모바일 기기를 위한 MQTT 로 간단한 메시징 지원
<b>기타</b>		
설명서	자세하고 정확함	불완전하며 때때로 부정확함
디스크 및 메모리 설치 공간	650MB 디스크  RAM 1GB 이하	70MB 디스크  RAM 2GB 이상
DataPower 와 통합	완전 통합	지원되지 않음
플랫폼 지원	20 종 이상의 플랫폼 지원	한정된 플랫폼 세트에 대한 타사 지원
설치 시간	스크립트로 설치 시 60 초 소요	스크립트로 설치 시 60 초 소요



## 부록 2: 이 백서에서 다루지 않은 항목

---

이 백서에서는 WebSphere MQ 및 ActiveMQ 의 여러 가지 중요한 기능을 평가했으나, 시간 제약으로 인해 이 조사 범위에 포함시키지 않은 테스트 및 비교 결과도 많이 있습니다. 예를 들면 다음과 같습니다.

- 보안 기능
- x 명의 게시자 및 구독자를 지원할 수 있는지 여부를 기준으로 본 확장성(유지되거나 유지되지 않는 게시/구독의 경우)
- TCO(총 소유비용) 심층 분석
- 문제 해결/디버깅 기능
- 백업 및 복원 기능
- WebSphere MQ 의 채널, 클러스터링, 전송 큐와 ActiveMQ 의 "브로커 네트워크" 간 차이점 심층 분석
- 기업 차원에서 플랫폼을 지원하기 위한 기술 세트 및 교육 자료의 출시 여부
- 업데이트(예: 픽스팩, 핫픽스) 설치
- 개발 경험
- 클라우드 배포 시나리오를 위한 두 플랫폼의 기능
- 출시된 다른 통합 제품(ESB, EAI, B2B 또는 B2C 등)에 대한 지원 여부 및 사용 편의성
- 타사 및 업체에서 제공하는 모니터링 도구 및 커뮤니티 지원
- 포인트-투-포인트 모델 및 게시/구독 모델에서 비지속 메시징의 성능
- WebSphere MQ 와 Red Hat JBoss A-MQ 같은 업체 브랜드의 ActiveMQ 배포판 비교