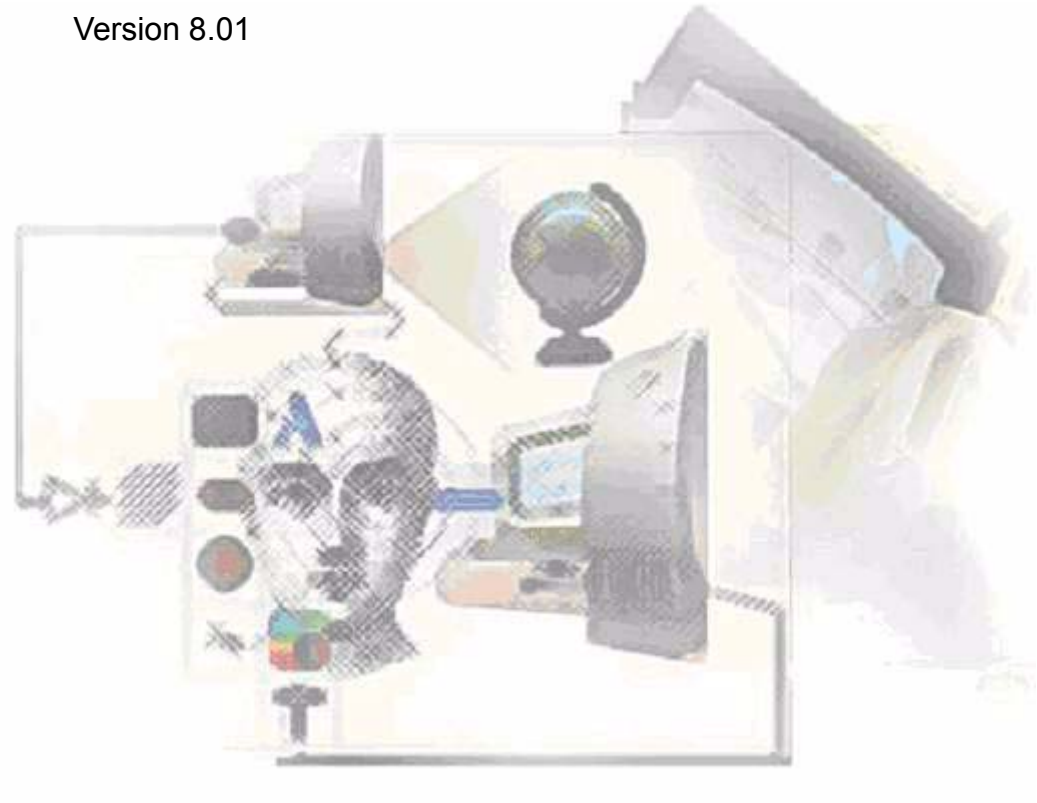
Fiorano Software, Inc.

FioranoMQ™ – Messaging Server

FioranoMQ High Availability Guide

Version 8.01



Fiorano Software, Inc.
718, University Avenue, Suite 212,
Los Gatos, California 95032 U.S.A.
Tel 1.408.354.3210
e-mail: info@fiorano.com

Contents

1

High Availability of FioranoMQ

Introduction

Today's real-time enterprise solutions often deploy a messaging middleware that enables communication between various sub components. This middleware is entrusted with important data that should be delivered reliably and as fast as possible to the recipient application. The middleware server might also be required to store this data in its data store until it is picked up.

A failure of this middleware message bus might bring the entire system to its knees within seconds. Hence it is absolutely imperative for the messaging back bone to provide its backup, which allows messaging operations to resume quickly in the event of a failure of the running server. This backup server should restore the state prior to failure of the original message server. Any data that was stored previously in the server's data store should be accessible through this backup server and most importantly this operation of shifting from one server to its back up should be automatic and transparent to the client application.

FioranoMQ introduces High Availability, which allows its applications to take advantage of its in-built fault tolerance capabilities. This Guide discusses the salient features of FioranoMQ's HA solution. It explains the working and the underlying architecture of the entire solution. It also provides step by step instructions on enabling HA in FioranoMQ.

Contents

The contents of this chapter are organized as follows.

Topic
Logging And Tracing
Limitations of HA

FioranoMQ's HA - An overview

FioranoMQ server when run in High - Availability mode has a designated backup server, which is started along with the primary FioranoMQ server. In case the primary server becomes un-available due to any reason, the backup server picks up all the messaging traffic immediately. This pair of primary and its backup server is known as an Enterprise Server and would be used to describe this pair throughout this document.

This Enterprise Server, represents a Highly Available entity, which appears as a single FioranoMQ server to its applications. JMS applications when initializing connect to the primary FMQ server, if available. If the primary server goes down due to any reason, all connections are automatically routed to the backup server and communications are restored immediately. Since, all this is transparent to the client application, it need not worry about re-connect logic in its code as it is handled by FioranoMQ's runtime internally.

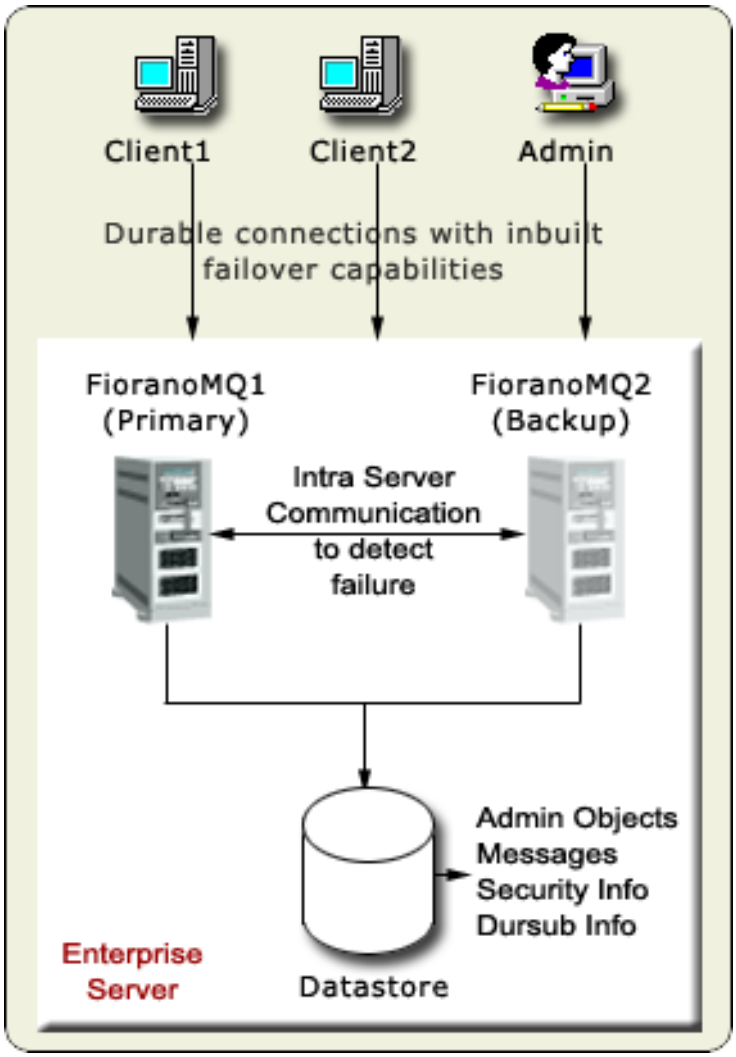


FIGURE 1-1 Enterprise Server

HA Components

The section below describes the components and some of the associated concepts that together make up an Enterprise Server.

Backup Server

Fiorano's HA solution requires running a backup FioranoMQ server. This server (also referred to as secondary server in this document) can be started on the same or different physical machine. This server takes up all the messaging traffic immediately as soon as it detects un-availability of its peer.

Server States

FioranoMQ Servers that make up the Enterprise Server can be in either Active or Passive State. Active State refers to the normal working mode of the server, while in passive mode, the server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. Upon startup, the server would establish communication with its peer server and upon finding it alive, enter into passive mode. It leaves its passive mode and becomes active (accepts client connections) only when it detects that its peer is un-available due to any reason.

Intra-Enterprise Server Communication

Both Primary and Secondary server open and listen on a TCP/IP port that allows them to establish a dedicated connection between themselves. This port is different from the one used by client applications for connecting to the server and hence does not affect normal messaging operations going on in the active server. This connection is established during the initialization phase of the servers and is used for exchanging health and state information between the two servers. This information is used by the servers to switch state to Active from Passive if required.

Common Persistent Message Store

Both Primary and Secondary server would be required to see a common persistent message store. In order to achieve this, Fiorano MQ Administrator can either point both the servers to the same physical database or can set up replication between the databases of the two servers. Both these options are available with Fiorano's File based database as well as on any third party JDBC compliant RDBMS Server.

Common Admin and Security

Besides the message store, Primary and Backup Server in an enterprise server would be required to share the admin objects (Destinations and Connection factories) and Security Information (ACLs and User Information) amongst themselves. This is achieved by using a common naming and realm storage (RDBMS, LDAP etc.) or setting up replication on these databases between the two servers.

Gateway Machine

Consider a scenario in which enterprise server consists of FMQ server 1 and FMQ server 2. Both these servers are constantly monitoring each other's status without any problems. Now, let's say FMQ server 2 due to some network failure goes out of network. Though FMQ server 2 is still running but it is no longer connected to the network (and hence not accessible to FMQ Server 2 and to client applications).

In this scenario, a third gateway machine is used to detect the HA server which is no longer available on the network. It becomes imperative to choose the gateway machine which itself is least expected to be out of network. It makes sense to use the actual gateway server of the network in which enterprise server is deployed as the Gateway machine for HA.

In case the Gateway machine itself goes out of network, then HA will continue to function properly as long as the two HA servers are present on the network. If one of the HA servers also goes out of network, then it is not possible to reach the expected state. In this case, both the servers will switch to the passive mode and enterprise server will not be in a position to process any client request. However it will again be available for client requests when either HA server 2 or gateway machine comes back in the network

FioranoMQ HA Salient Features

Shared and Replication database

FioranoMQ provides complete flexibility to administrators giving them an option to either use shared database (between active and passive server) or use database replication (from active to passive server). So in scenarios where it is not possible to share the database, administrators can still use FioranoMQ's High Availability using the inbuilt replication support.

Application Failover

In case the primary server becomes unavailable, all the client applications connected to it are automatically re-connected to the secondary server. The process of shifting from the primary server to the backup server or vice-versa is transparent to the application. It need not bother about writing reconnect logic in it's code. This is achieved by connecting to the server through a Durable Connection. In case a backup server is available, the Durable Connection would connect to the backup server else it waits for the server to restart. Further, it stores all the data sent during the disconnected period in a local repository and transfers this data as soon as the connection is re-established, thus making the system highly reliable and robust even in the case of network failures.

For more information on Durable Connections refer to the chapter named 'Durable Connections' of the FioranoMQ Developer's Guide.

Durable connections is a proprietary feature of FioranoMQ (though it does not require any proprietary APIs) and should not be confused with Durable Subscribers.

Data Store Consistency maintained between server switches

When the primary server becomes un-available, its backend database state is conserved. This state is picked up by the secondary server when it becomes ready for action. This avoids loss of persistent information between server switches and at the same time provides access to the information that was stored through one server from its backup server. For example, all the messages that were published on to various destinations residing on the primary server before it went down are available to valid consumers coming through the secondary backup server without any message getting lost.

Expensive HA Hardware Not Required

Fiorano's HA solution is purely implemented in software and is not dependent on expensive hardware solutions. It can be run on any java supported platform. With Shared database option, one might want to use a RAID or SAN disks if using HA over Fiorano's proprietary file based data store for enhanced speed and stability, but these hardware are not an essential component of Fiorano's HA solution. These hardware can also be avoided by using either replication support or using a central RDBMS server as the message store in the Enterprise Server.

Implementing Cluster

The Enterprise server can be clustered with other Enterprise Servers or even stand alone FioranoMQ servers. This cluster together can share destinations (using a common naming store) and provide load balancing facilities.

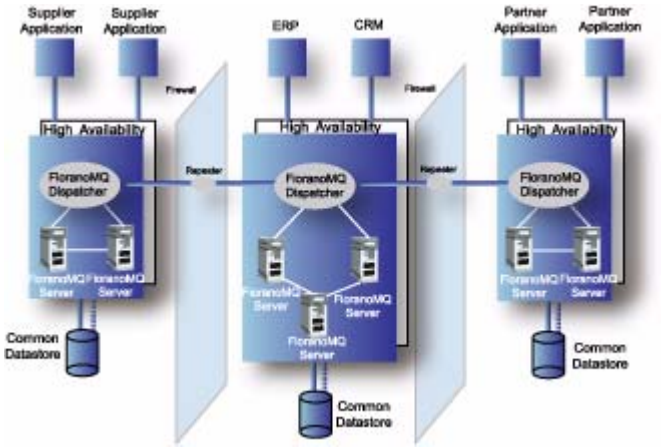


FIGURE 1-2 Clustering of Enterprise Servers with repeaters and bridges

HA Example Scenario

This section of the document provides a description of events that occur in case one of the servers in the Enterprise Server becomes un-available.

State - 1 (Normal Operation State)

All client applications are connected to one of the servers in the Enterprise Server, let's assume that they are connected to the primary server at this instant. The backup server is up and running but is in passive mode. This server would not accept any client connections at this point of time. Primary and Backup Server are continuously exchanging health information over a dedicated channel. All persistent information is being stored in the backend datastore through the primary server. This is pictorially shown below.

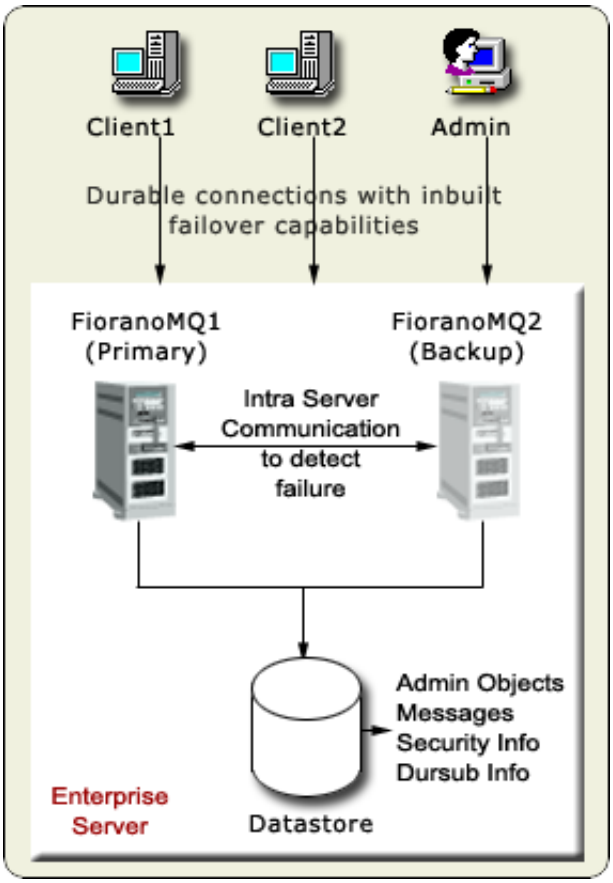


FIGURE 1-3 This is the normal mode of operations

State - 2 (Active Server goes down)

The backup server detects the primary server's failure and initiates its start up sequence. All client applications connected to the primary server also detect the problem and Fiorano's runtime library internally attempts to re-connect back to the secondary server. New messages that are published in this down-time are stored in a local repository at all client machines.

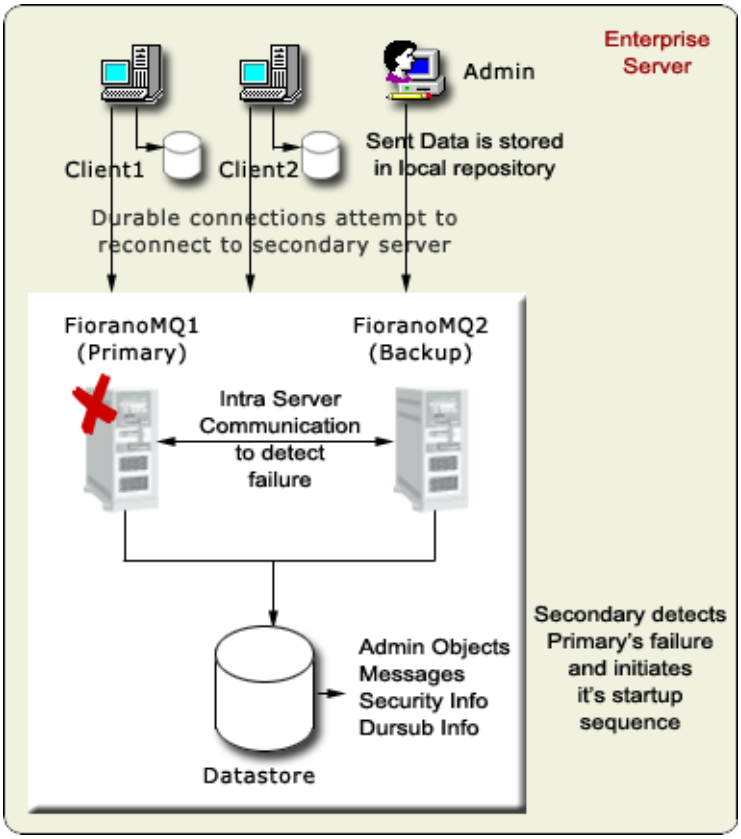


FIGURE 1-4 Primary Server goes down

State - 3 (Backup Server resumes operations)

Backup Server starts up completely and is ready to take up client connections. All client applications re-connect with the secondary server. The messages that were stored in the local repository (that were sent during the down-time) are sent to the secondary server. All durable consumers pick up messages from where they had left.

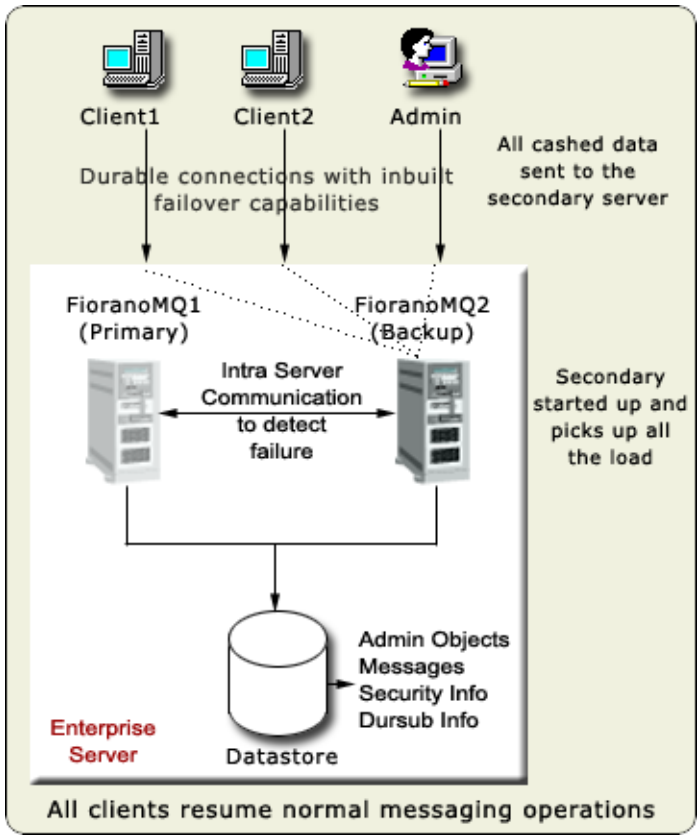


FIGURE 1-5 The secondary server starts up completely

Upon restarting the primary server, it would detect that its backup is alive and enter into passive mode. It would continuously ping the backup server and will initiate its startup sequence if the backup server goes down due to any reason.

Configuring FioranoMQ in HA Mode

As mentioned earlier, Fiorano MQ servers within an Enterprise Server are required to have a similar backend database. This can be achieved by using a common or shared database or setting up replication between the database instances. The first decision to be taken while setting up HA would be to select one of these two options. FioranoMQ installer comes with pre-built profiles for both shared & replication mode that are pre-configured to demonstrate HA on a single machine.

Default HA Profiles

This section of the document provides details about the pre-configured HA profiles. These profiles can be found in %FIORANO_HOME%\fmq\profiles\FioranoMQ_HA_rpl & FioranoMQ_HA_shared directories respectively.

Primary & Secondary Servers can be started simply by booting the container with these profiles. The table below lists the profiles to be used for Primary & Secondary Servers when setting up HA in Replication or Shared mode.

Mode	Profile Directory	Meant for
Replication	fmq/profiles/FioranoMQ_HA_rpl/HAPrimary	Primary Server
Replication	fmq/profiles/FioranoMQ_HA_rpl/HASecondary	Secondary Server
Shared	fmq/profiles/FioranoMQ_HA_shared/HAPrimary	Primary Server
Shared	fmq/profiles/FioranoMQ_HA_shared/HASecondary	Secondary Server

In order to launch the server on one of these profiles, use the following command:

runContainer -fmq.profile %SELECTED_PROFILE%

For example, to launch the primary server in replication mode, use the command

runContainer -fmq.profile FioranoMQ_HA_rpl/HAPrimary

This document will refer the selected profile (in FioranoMQ_HA_rpl or FioranoMQ_HA_shared directories) as %SELECTED_HA_PROFILE%. The possible values for %SELECTED_HA_PROFILE% are as shown in the table above

Configuration Steps

Configuring HA in FMQ Server primary involves configuring the following modules:

- HA Layer

- Admin Objects Store
- Security Store
- Message Store

The HA Layer configuration deals with configuring the peer server details. The important configuration parameters defined in this layer are "isPrimary (true or false), HA Port, Backup Server's IP Address, Backup Server's Client Port, Backup Server's HA Port. This configuration is required to be done in both replication & shared HA modes. Additionaly Admin Objects Store, Security Store & Message Store are required to be configured for setting up Shared HA. In other words when configuring HA, one of the following set of instructions would be followed depending on the mode (replication or shared).

Replication Mode

Step 1: Install FMQ Servers

Step 2: HA Configuration

HA Shared Mode

Step 1: Install FMQ Servers

Step 2: HA Configuration

Step 3: Admin Store Configuration

Step 4: Security Store Configuration

Step 5: Message Store Configuration

The sections to follow provides step-by-step instructions for the above steps.

Configuring FioranoMQ HA in "replication" mode is much easier, as the pre-created profiles are mostly ready to be used (only the gateway IP needs to be modified). These profiles are pre-configured to demonstrate HA on a single machine. However one can easily setup HA on two different machines as well by properly configuring the HA Layer (as described below).

When setting up the server in shared mode, besides setting up the HA Layer one will be required to modify the server's configuration to point it to shared admin store, security store and message store. Instructions for doing this can be found in Steps 3-5 in sections below.

Step 1: FMQ Server(s) Installation

FioranoMQ Servers that make up an Enterprise Server can be installed on a single machine or two separate machines that might vary in hardware or software configuration. When setting up HA on one machine no further change is required - The default configuration (for primary & secondary server) is ready to be used. However, when setting up on different machines, the IP addresses of the peer servers will have to be configured in HA Layer. The following section provides step by step instructions on doing this.

The pre-created profile are configured to run HA on one machine for demonstration purpose. Fiorano does not recommend setting up HA on one machine.

Step 2: HA Configuration

HA Layer within an FMQ Server is responsible for creating a dedicated connection with the peer backup server. This dedicated connection is used to exchange health & state information amongst the peers. The following section provides details on configuring HA Layer in FMQ Server and would be applicable for both primary & secondary server.

1. Invoke Fiorano Studio from installer.
2. Select Tools->Configure FioranoMQ from the Menu Bar, select the directory pointing to %SELECTED_HA_PROFILE% from the dialog box and click the open button.

When configuring the server (in offline mode) through Studio, please ensure that FMQ Server is not running.

3. Configure self - HA Port. This port refers to tbe port number on which HA Layer would accept connection from its peer server. The default HA profiles have their HA Port configured for primary server as 2000 and secondary server as 3000. In order to modify this Navigate to the node mentioned below and change the port number in "properties" panel.

   %SELECTED_HA_PROFILE%->Core->HA->HAConnection Manager

A change in this parameter would require modifying HAKRPCProvider Configuration in the peer server.

4. Configure IP & HA Port of remote Server. HA Layer of an FMQ server connects to the backup server's HA Layer. In order to configure the back-up server's IP & Port navigate to the node specified below & set the values of Backup Address & Port displayed in the properties panel at the top right corner.

   %SELECTED_HA_PROFILE%->Core->HA->HAKRPCProvider

The default profiles have Backup HAIP address configured to "localhost" and Backup HAPort to 3000 in primary server and 2000 in secondary server. In other words, Primary Server's HA layer in default profiles would try to connect at localhost on port 3000 while the Secondary's HA Layer would try to connect to localhost on port 2000 by default.

The BackupHA Port here does not refer to the port on which backup server accepts client connections but the port on which its HA layer is listening. This should be same as the value of HAConnectionManager's port configured in the peer server.

5. Configure Gateway Server. HA Layer pings the gateway machine to determine the status of network connectivity. The default profiles have gateway configured to "localhost" and should be changed to any third independent reliable machine (preferrably to the physical router machine on the network). In order to configure Gateway Server, Navigate to the node specified below & modify the value of Gateway I.P. Address & Gateway Server Port in the properties panel.

   %SELECTED_HA_PROFILE%->Core->HA->HAManager & Specify

6. Configure 'isPrimary'. Set 'isPrimary' to true for primary FMQ Server & false for back-up Server by modifying its value as shown in the properties panel on clicking the node specified below.

   %SELECTED_HA_PROFILE%->Core->HA->HAManager & Specify

The default profiles are already configured properly. Hence no change would be required when using them.

7. Common Configuration. HA requires enabling Durable Connections, Pinging & Auto Revalidation support in FMQ Server. This can be done by navigating to & modifying the parameters specified below in the properties panel.

   %SELECTED_HA_PROFILE%->Core-->Common -> CommonConfig

   a. Set AllowDurableConnections Property to true.

   b. Set PingEnabled to true.

   c. Set EnableAutoRevalidation to true

The pre-created profiles already have this configuration and hence no change would be required when working on them.

8. Configure Backup Server IP & Port. Specify the BackUpServerIP and BackupServerPort which will be used by clients for connection Failure. This IP:Port should refer to those of the backup FMQ Server's IP Address and the port on which it accepts client connections. This can be done by navigating to the node specifeid below and modifying the Backup Server IP & Backup Server Port parameters in the properties panel.

   %SELECTED_HA_PROFILE%->Core-->Common -> CommonConfig

FMQ Server would use these parameters to set backup server url in default connection factories. For client connections to fail-over, it is important to note that the backup server url is properly configured in the connection factories. If the configuration is changed once the default connection factories have been created, one

should either re-create the database or use self created connection factories with correct urls.

The default profiles have the IP address configured to localhost for both primary and secondary server and the Backup Port to 1956 in primary server and 1856 in secondary server. NOTE - In default profile the secondary server accepts client connections on port 1956 and the primary server accepts client connections on port 1856.

9. Right click on the "%SELECTED_HA_PROFILE%" root node present under the Explorer tab on the left hand side of the Studio window and select the "Save" option to save the configuration done so far.

10. Example Configuration

Parameter	Primary Server	Backup Server
url	http://164.164.128.128:1856	http://164.164.128.108:1956
HAKRPCProvider		
BackupHAIP Address	164.164.128.108	164.164.128.128
Back-upHAPort	3000	2000
HAConnectionManager		
Port	2000	3000
Common Config		
Backup IP Address	164.164.128.108	164.164.128.128
Port	1956	1856
Enable-DurableCon-nections	True	True
PingEnabled	True	True
HAManager		
isPrimary	True	False
Gateway IP Address	164.164.128.225	164.164.128.225

After changing the HA Layer's configuration, the server's existing database should be cleared before re-starting. The existing datastore can be cleared by executing the "clearDB" script in fmq/bin folder of your installation with profile name as an argument. For example:

clearDB FioranoMQ_HA_rpl/HAPrimary
clearDB FioranoMQ_HA_rpl/HASecondary

For windows, please use '\' instead of '/'.

This is required so as to re-create the connection factories to include the backup IP address of the peer server. If clearing the entire datastore is not possible, one should use newly created connection factories for HA enabled applications.

The rest of this section (Steps 3-5) are valid for setting up HA in shared mode only.

Step 3: Configuring Admin Store

In order to configure Admin Store, decide on the type of Shared Admin Object Store that you would like to use for FMQ Server. The possible options are given below:

a. File Based (Fiorano's proprietary file format) - Default option

b. RDBMS Server

c. XML File

d. LDAP Server

It is recommended to use an RDBMS or an LDAP Server as the store for admin objects when setting up the Enterprise Server.

Configure both FMQ Servers to point to a common storage as per instructions provided in Chapter on "Admin Store" in Admin Guide.

If using default implementation or xml based implementation in shared HA mode, the Path attribute for Naming Manager Server should be made to point to same physical location. This might require mapping a drive locally on a machine if both servers that are part of Enterprise Server are running on separate machines.

Step 4: Configuring Common Security Store

FioranoMQ Security Realms can be categorized into two domains:

a. Principal- user management

b. ACL- Access Control List

FioranoMQ Server provides a pluggable component for both the above domains. These components namely Principal Manager and ACL Manager provide the user management and ACL Management services respectively to the server. Now both

these components require a data store to persist security information that they manage. The possible options for storage media are as follows:

- File Based (Fiorano's proprietary file format) - Default option
- RDBMS Server
- XML File
- LDAP Server

When configuring an FMQ Server in HA shared mode, the data store used by both the servers (in an Enterprise Server) should point to the same physical location. It is recommended to use an RDBMS or an LDAP Server as the store for security managers when setting up the Enterprise Server.

☞ Instructions for configuring Security Realms to use the above types of storage medias can be found in Chapter titled - "FioranoMQ Security" in Admin Guide.

If using default implementation or xml based implementation in shared HA mode, the Path attribute for Principal or ACL Manager should be made to point to same physical location. This might require mapping a drive locally on a machine if both servers that are part of Enterprise Server are running on separate machines.

Step 5: Configuring Database

FioranoMQ Server provides options to use an RDBMS Server or file based data base for storing messages. Within an Enterprise Server both the servers are required to point to the same database (RDBMS or File based). Instructions for doing this for both types of databases are listed below.

RDBMS Server

☞ Please refer Chapter titled "Message Store" in Admin Guide for instructions on setting up a single instance of FioranoMQ to use an RDBMS Server as it's back-end message store.

Instructions for several commonly used databases are provided. When setting up an Enterprise Server, one should configure both the server instances with exactly same set of database parameters.

When using an RDBMS Server, please make sure to specify the data store of destination being used as RDBMS instead of the default File Based.

File Based Datastore

In Default Mode, Fiorano MQ creates a proprietary file based data base for storing messages. This database by default is created in the run folder of the profile being used. For example with default profile, the database directory would be **%FMQ_DIR%\fmq\profiles\FioranoMQ\run**.

Within an enterprise server both the servers should have their databases pointing to the same physical directory. Database directory can be specified through a command line option -fmq.db.path when running the server. For example assuming the shared database is to be created in **c:\fmq-db** directory, the command line to launch both the servers would look like that shown below.

```
%FMQ_DIR%\bin> runContainer -fmq.db.path c:\fmq-db
```

In case both the servers are to run on separate machines, the database directory will have to be made accessible to the other machine (by mapping the drive locally for example).

Launching

When launching the Enterprise Server, the launch Sequence within the two servers is not important. Any server (primary or backup) can be started first. On start up, both the servers establish communication with each other for exchanging health information. The servers can be launched by using the runContainer script (available in fmq/bin of installation) and supplying the **%SELECTED_HA_PROFILE%** as an argument. For example, when launching the Primary HA server over Replicated Mode, the command line would be

```
runContainer -fmq.profile FioranoMQ_HA_rpl/HAPrimary
```

Verifying the HA Setup

The following can be used to verify the correctness of HA Setup:

Server Side Console Messages

On starting an FMQ Server that is part of an HA Server, the server prints debug information about its own state (ACTIVE, PASSIVE, WAITING). It also prints information about its peer server's state whenever it detects a change in its state. Console would have statements like that shown below:

Local Server switched to ACTIVE

Or

Local Server switched to PASSIVE

Shutdown the active server

Once started up, one of the servers in the Enterprise Server would be in ACTIVE state while the other one would be in PASSIVE state. Upon shutting down the active server, its peer should detect this and should switch to active state. This change would be indicated with debug statements similar to that shown below on the server console.

Old status of remote server = ACTIVE
New status of remote server = DEAD
License key issue date 4/21/05 1:03 PM
License key is valid for 45 days
License key warning date 6/1/05 11:59 PM
License key expiry date 6/5/05 11:59 PM
Product Name: {FioranoMQ} Major Version: {8.0} Minor Version: {3669}
FioranoMQ Server accepting connections at http://164.164.128.57:1856
Local Server switched to STANDALONE

Sample

Once both the primary and the secondary servers are running in HA mode, run a sample using the following steps to verify the installation.

1. Open fmq\samples\pubsub\DurableSubscribers\Publisher_d.java and DurableSubscriber.java in the same directory in your favourite Java IDE.
2. Include the parameter "BackupConnectURLs" in the environment being passed to InitialContext. This variable should point to the backup server's url. For example, if the backup server is running on host called "backup-server" on port 1856, the following tag should be added to environment:

```
env.put ("BackupConnectURLs", "http://backup-server:1856");
```

Make this modification in both Publisher.java as well as DurableSubscriber.java

3. Start off the durable subscriber and publisher applications by the following command line:

```
runclient DurableSubscriber [-topicName ......]
runclient Publisher   [-topicName .....]
```

When using a central RDBMS Database as the backend data store for FMQ, please ensure to use an RDBMS based destination. FioranoMQ by default creates "primaryRDBMSTopic" during startup if RDBMS Database is enabled which can be used for this purpose. Alternatively, another RDBMS based topic can be created through the Admin GUI.

Please ensure that the connection factory being used by the sample (primaryTCF by default) has primary and secondary URL pointing to the two servers. This would automatically be done if FMQ Server is started fresh after configuring HA correctly, else please create a new connection factory with desired primary and secondary urls and use the same in the sample.

4. Publish some message by typing a String through the console and pressing enter. These messages would be received by the Subscriber which would print the text on the console.
5. Now shutdown the primary server by using shutdown script.
6. The secondary server would initiate its startup sequence within the ha.ping_interval as set in the Fiorano Studio.
7. Publish some more messages before the backup server starts up completely. These messages would not be delivered to the subscriber immediately.
8. As soon as the secondary server starts up and the durable subscriber's connection is re-established, all the pending messages would be delivered.

Logging And Tracing

Logging

Just like the base FioranoMQ server, HA server can log into files, console or use any other custom made logger. The type of logging is controlled through the Loggers module in the Fiorano Studio. Options to log all information on console or save all the log into a log or error files is provided through Console-based and File-based logging respectively.

Tracing

The amount of information that is logged by the HA server can be controlled through the trace variable for HA. The server upon startup initializes the value of various trace components. Integers between 0-6 are valid acceptable values. The greater the trace for HA in this file, greater would be the information logged.

```
HA=0  /** No Log                                  **/
HA=1  /** Default Log                             **/
HA=2  /** Displays State Info of HA server        **/
HA=4  /** Displays intra server communication info **/
HA=6  /** Maximum HA log                          **/
```

☞ For more information on logging and tracing, refer to the section "Working with LoggerFactory" in the Administration Guide.

Limitations of HA

1. Client level transactions do not span across servers in the Enterprise Server when running on shared mode. Transacted sessions involving Receivers will be rolled back in case the primary server crashes. Hence, the messages delivered in that transaction will be redelivered to the receivers when connected to the backup server.
2. Distributed transactions, which are in execution during transition phase, will become "in-doubt transactions" when the primary server goes down. These transactions will get rolled back and can be recovered after the client reconnects to the secondary server.
3. JMS Topic Requestor may not receive its intended reply if failover occurs after a request is sent. This occurs because JMS Topic Requestor creates a non-durable subscriber which can miss a message during failover. However if a topic requestor creates a durable subscriber to listen for replies, then it will work fine even during failover.
4. In case both ha servers (primary as well as backup) go down, Requestor will receive a duplicate reply (with reDelivered Flag = true) for the first request made after failover.