



**Fiorano**  
Peer-to-Peer Dataflow Pipelines™

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

# FioranoMQ 9

## High Availability Guide

### AMERICA'S

Fiorano Software, Inc.  
718 University Avenue Suite  
212, Los Gatos,  
CA 95032 USA  
Tel: +1 408 354 3210  
Fax: +1 408 354 0846  
Toll-Free: +1 800 663 3621  
Email: [info@fiorano.com](mailto:info@fiorano.com)

### EMEA

Fiorano Software Ltd.  
3000 Hillswood Drive Hillswood  
Business Park Chertsey Surrey  
KT16 0RS UK  
Tel: +44 (0) 1932 895005  
Fax: +44 (0) 1932 325413  
Email: [info\\_uk@fiorano.com](mailto:info_uk@fiorano.com)

### APAC

Fiorano Software Pte. Ltd.  
Level 42, Suntec Tower Three 8  
Temasek Boulevard 038988  
Singapore  
Tel: +65 68292234  
Fax: +65 68292235  
Email: [info\\_asiapac@fiorano.com](mailto:info_asiapac@fiorano.com)

**Fiorano**

Entire contents © Fiorano Software and Affiliates. All rights reserved. Reproduction of this document in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Fiorano disclaims all warranties as to the accuracy, completeness or adequacy of such information. Fiorano shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The opinions expressed herein are subject to change without prior notice.

## FIORANO END-USER LICENSE AGREEMENT

THIS FIORANO END-USER LICENSE AGREEMENT (THE "AGREEMENT") IS A LEGAL AGREEMENT BETWEEN YOU (HEREINAFTER "CUSTOMER"), EITHER AN INDIVIDUAL OR A CORPORATE ENTITY, AND FIORANO SOFTWARE, INC., HAVING A PLACE OF BUSINESS AT 718 UNIVERSITY AVE, SUITE 212 LOS GATOS, CA 95032, USA, OR ITS AFFILIATED COMPANIES (HEREINAFTER "FIORANO") FOR CERTAIN SOFTWARE DEVELOPED AND MARKETED BY FIORANO AS DEFINED IN GREATER DETAIL BELOW. BY OPENING THIS PACKAGE, INSTALLING, COPYING, DOWNLOADING, EXTRACTING AND/OR OTHERWISE USING THE SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING PARTY TO THIS AGREEMENT ON THE DATE OF INSTALLATION, COPYING, DOWNLOAD OR EXTRACTION OF THE SOFTWARE (THE "EFFECTIVE DATE"). IF YOU DO NOT AGREE WITH ANY OF THE TERMS OF THIS AGREEMENT, PLEASE STOP INSTALLING AND/OR USING THE SOFTWARE AND PROMPTLY RETURN THE UNUSED SOFTWARE TO THE PLACE OF PURCHASE. BY DEFAULT, THE SOFTWARE IS MADE AVAILABLE TO CUSTOMERS IN ONLINE, DOWNLOADABLE FORM. THE TERMS OF THIS AGREEMENT SHALL APPLY TO EACH SOFTWARE LICENSE GRANTED BY FIORANO UNDER THIS AGREEMENT.

### 1. DEFINITIONS.

- a. **"Affiliate"** means, in relation to Fiorano, another person firm or company which directly or indirectly controls, is controlled by or is under common control with Fiorano and the expression 'control' shall mean the power to direct or cause the direction of the general management and policies of the person firm or company in question.
- b. **"Commencement Date"** means the date on which Fiorano delivers the Software to Customer, or if no delivery is necessary, the Effective Date set forth in this Agreement or on the relevant Order Form.
- c. **"Designated Center"** means the computer hardware, operating system, customer-specific application and Customer Geographic Location at which the Software is deployed as designated on the corresponding Order Form.
- d. **"Designated Contact"** shall mean the contact person or group designated by Customer and agreed to by Fiorano who will coordinate all Support requests to Fiorano.
- e. **"Documentation"** means the user guides and manuals for installation and use of the Software. Documentation is provided in CD-ROM or bound form, whichever is generally available.
- f. **"Error"** shall mean a reproducible defect in the Supported Program or Documentation when operated on a Supported Environment which causes the Supported Program not to operate substantially in accordance with the Documentation.
- g. **"Excluded Components"** shall mean such components as are listed in Exhibit B. Such Excluded Components do not constitute Software under this Agreement and

are third party components supplied subject to the corresponding license agreements specified in Exhibit B.

- h. **"Excluded License"** shall mean and include any license that requires any portion of any materials or software supplied under such license to be disclosed or made available to any party either in source code or object code form. In particular, all versions and derivatives of the GNU GPL and LGPL shall be considered Excluded Licenses for the purposes of this Agreement.
- i. **"Resolution"** shall mean a modification or workaround to the Supported Program and/or Documentation and/or other information provided by Fiorano to Customer intended to resolve an Error.
- j. **"Residuals"** shall mean information in non-tangible form which may be retained by persons who have had access to the Confidential Information, including ideas, concepts, know-how or techniques contained therein.
- k. **"Order Form"** means the document in hard copy form by which Customer orders Software licenses and services, and which is agreed to in writing by the parties. The Order Form shall reference the Effective Date and be governed by the terms of this Agreement. Customer understands that any document in the nature of a purchase order originating from Customer shall not constitute a contractual offer and that the terms thereof shall not govern any contract to be entered into between Fiorano and Customer. The Order Form herein shall constitute an offer to purchase made by the Customer under the terms of the said Order Form and this Agreement.
- l. **"Software"** means each of the individual Products, as further outlined in Exhibit-A, in object code form distributed by Fiorano for which Customer is granted a license pursuant to this Agreement, and the media, Documentation and any Updates thereto.
- m. **"Support"** shall mean ongoing support provided by Fiorano pursuant to the terms of this Agreement and Fiorano's current support policies. **"Supported Program" or "Supported Software"** shall mean the then current version of the Software in use at the Designated Center for which the Customer has paid the then-current support fee (**"Support Fee"**).
- n. **"Support Hours"** shall mean 9 AM to 5 PM, Pacific Standard Time, Monday through Friday, for Standard Support.
- o. **"Support Period"** shall mean the period during which Customer is entitled to receive Support on a particular Supported Program, which shall be a period of twelve (12) months beginning from the Commencement Date, or if applicable, twelve (12) months from the expiration of the preceding Support Period. Should Fiorano withdraw support pursuant to section 1 (q), the Support Period shall be automatically reduced to the expiration date of the appropriate Software.
- p. **"Supported Environment"** shall mean any hardware and operating system platform which Fiorano provides Support for use with the Supported Program.
- q. **"Update"** means a subsequent release of the Software that Fiorano generally makes available for Supported Software licensees at no additional license fee

other than shipping and handling charges. Update shall not include any release, option, feature or future product that Fiorano licenses separately. Fiorano will provide Updates for the Supported Programs as and when developed for general release in Fiorano's sole discretion. Fiorano may withdraw support for any particular version of the Software, including without limitation the most current Update and any preceding release with a notice of three (3) months to Customer.

## **2. SOFTWARE LICENSE.**

### **(a) Rights Granted, subject to the receipt by Fiorano of appropriate license fees.**

**(i)** The Software is Licensed to Customer for use under the terms of this Agreement and **NOT SOLD**. Fiorano grants to Customer a limited, non-exclusive, world wide license to use the Software as specified on an Order Form and subject to the licensing restrictions in Exhibit C under this Agreement, as follows:

**(1)** to use the Software solely for Customer's operations at the Designated Center consistent with the use limitations specified or referenced in this Agreement, the Documentation for such Software or any Order Form accepted by Fiorano pursuant to this Agreement. Customer may not relicense, rent or lease the Software or use the Software for third party training, commercial timesharing or service bureau use;

**(2)** to use the Documentation provided with the Software in support of Customer's authorized use of the Software;

**(3)** to make a single copy for back-up or archival purposes and/or temporarily transfer the Software in the event of a computer malfunction. All titles, trademarks and copyright or other restricted rights notices shall be reproduced in any such copies;

**(4)** to allow third parties to use the Software for Customer's operations, so long as Customer ensures that use of the Software is in accordance with the terms of this Agreement.

**(ii)** Customer shall not copy or use the Software (including the Documentation) except as specified in this Agreement and applicable Order Form. Customer shall have no right to use other third party software or Excluded Components that are included within the Software except in connection and within the scope of Customer's use of Fiorano's Software product.

**(iii)** Customer agrees not to cause or permit the reverse engineering, disassembly, decompilation, or any other attempt to derive source code from the Software, except to the extent expressly provided for by applicable law.

**(iv)** Customer hereby warrants that it shall not, by any act or omission, cause or permit the Products or any part thereof to become expressly or impliedly subject to any Excluded License.

(v) Fiorano and its Affiliates shall retain all title, copyright and other proprietary rights in the Software. Customer does not acquire any rights, express or implied, in the Software, other than those specified in this Agreement.

(vi) Customer agrees that it will not publish or cause or permit to be published any results of benchmark tests run on the Software.

(vii) If the Software is licensed for a specific term, as noted on the Order Form, then the license shall expire at the end of the term and the termination conditions in section 4(d) shall automatically become applicable.

**(b) Transfer.** Customer may transfer a Software license within its organization upon notice to Fiorano; transfers are subject to the terms and fees specified in Fiorano's transfer policy in effect at the time of the transfer. If the Software is licensed for a specific term, then it may not be transferred by Customer.

**(c) Verification.** At Fiorano's written request, Customer shall furnish Fiorano with a signed certification verifying that the Software is being used pursuant to the provisions of this Agreement and applicable /Order Form. Fiorano (or Fiorano's designee) may audit Customer's use of the Software. Any such audit shall be conducted during regular business hours at Customer's facilities and shall not unreasonably interfere with Customer's business activities. If an audit reveals that Customer has underpaid fees to Fiorano, Customer shall be invoiced directly for such underpaid fees based on the Fiorano Price List in effect at the time the audit is completed. If the underpaid fees are in excess of five percent (5%) of the aggregate license fees paid to Fiorano pursuant to this Agreement, the Customer shall pay Fiorano's reasonable costs of conducting the audit. Audits shall be conducted no more than once annually.

**(d) Customer Specific Objects.**

(i) The parties agree and acknowledge, subject to Fiorano's underlying proprietary rights, that Customer may create certain software objects applicable to Customer's internal business ("Customer Specific Objects"). Any Customer Specific Object developed solely by Customer shall be the property of Customer. To the extent that Customer desires to have Fiorano incorporate such Customer Specific Objects into Fiorano's Software (and Fiorano agrees, in its sole discretion, to incorporate such Customer Specific Objects), Customer will promptly deliver to Fiorano the source and object code versions (including documentation) of such Customer Specific Objects, and any updates or modifications thereto, and hereby grants Fiorano a perpetual, irrevocable, worldwide, fully-paid, royalty-free, exclusive, transferable license to reproduce, modify, use, perform, display, distribute and sublicense, directly and indirectly, through one or more tiers of sublicensees, such Customer Specific Objects.

(ii) Any objects, including without limitation Customer Specific Objects, developed solely or jointly with Customer by Fiorano shall be the property of Fiorano.

**(e) Additional Restrictions on Use of Source Code.**

Customer acknowledges that the Software, its structure, organization and any human-readable versions of a software program ("Source Code") constitute valuable trade secrets that belong to Fiorano and/or its suppliers Source Code Software, if and when supplied to Customer shall constitute Software licensed under the terms of this Agreement and the Order Form. Customer agrees not to translate the Software into another computer language, in whole or in part.

(i) Customer agrees that it will not disclose all or any portion of the Software's Source Code to any third parties, with the exception of authorized employees ("Authorized Employees") and authorized contractors ("Authorized Contractors") of Customer who (i) require access thereto for a purpose authorized by this Agreement, and (ii) have signed an employee or contractor agreement in which such employee or contractor agrees to protect third party confidential information. Customer agrees that any breach by any Authorized Employees or Authorized Contractors of their obligations under such confidentiality agreements shall also constitute a breach by Customer hereunder.

(ii) Customer shall ensure that the same degree of care is used to prevent the unauthorized use, dissemination, or publication of the Software's Source Code as Customer uses to protect its own confidential information of a like nature, but in no event shall the safeguards for protecting such Source Code be less than a reasonably prudent business would exercise under similar circumstances. Customer shall take prompt and appropriate action to prevent unauthorized use or disclosure of such Source Code, including, without limitation, storing such Source Code only on secure central processing units or networks and requiring passwords and other reasonable physical controls on access to such Source Code.

(iii) Customer shall instruct Authorized Employees and Authorized Contractors not to copy the Software's Source Code on their own, and not to disclose such Source Code to anyone not authorized to receive it.

(iv) Customer shall handle, use and store the Software's Source Code solely at the Customer Designated Center.

#### **(f) Acceptance tested Software**

Customer acknowledges that it has, prior to the date of this Agreement, carried out adequate acceptance tests in respect of the Software. Customer's acceptance of delivery of the Software under this Agreement shall be conclusive evidence that Customer has examined the Software and found it to be complete, and in accordance with the Documentation, in good order and condition and fit for the purpose for which it is required.

### **3. TECHNICAL SERVICES.**

**(a) Maintenance and Support Services.** Maintenance and Support services will be provided under the terms of this Agreement and Fiorano's support policies in effect on the date Support is ordered by Customer. Support services shall be provided from Fiorano's principal place of business or at the Designated Center, as determined in Fiorano's sole discretion. If Fiorano sends personnel to the Designated Center to resolve any Error in the Supported Program, Customer shall pay Fiorano's reasonable travel, meals and lodging expenses.

**(b) Consulting and Training Services.** Fiorano will, upon Customer's request, provide consulting and training services agreed to by the parties pursuant to the terms of a separate written agreement.

**(c) Incidental Expenses.** For any on-site services requested by Customer, Customer shall reimburse Fiorano for actual, reasonable travel and out-of-pocket expenses incurred (separate from then current Support Fees).

**(d) Reinstatement.** Once Support has been terminated by Customer or Fiorano for a particular Supported Program, it can be reinstated only by prior approval from Fiorano and then only upon payment of the reinstatement fee applicable at the time of reinstatement.

**(e) Supervision and Management.** Customer is responsible for undertaking the proper supervision, implementation and management of its use of the Supported Programs, including, but not limited to: (i) assuring proper Supported Environment configuration, Supported Programs installation and operating methods; and (ii) following industry standard procedures for the security of data, accuracy of input and output, and back-up plans, including restart and recovery in the event of hardware or software error or malfunction. Fiorano does not warrant (i) the performance of, or combination of, Software with any third party software, (ii) any implementation of the Software that does not follow Fiorano's delivery methodology, or (iii) any components not supplied by Fiorano.

**(f) Training.** Customer is responsible for proper training of all appropriate personnel in the operation and use of the Supported Programs and associated equipment.

**(g) Access to Personnel and Equipment.** Customer shall provide Fiorano with access to Customer's personnel and its equipment during Support Hours. This access must include the ability to dial-in from Fiorano facilities to the equipment on which the Supported Programs are operating and to obtain the same access to the equipment as those of Customer's employees having the highest privilege or clearance level. Fiorano will inform Customer of the specifications of the modem equipment and associated software needed, and Customer will be responsible for the costs and use of said equipment.

**(h) Support Term.** Upon expiration of an existing Support Period for a particular Supported Program, a new Support Period shall automatically begin for a consecutive twelve (12) month term ("Renewal Period") so long as (i) Customer pays the Support Fee within thirty (30) days of invoice by Fiorano; and (ii) Fiorano is still offering Support on such Supported Program.

**(i) Annual Support Fees.** Annual Support Fees shall be at the rates set forth in the applicable Order Form.

#### **4. TERM AND TERMINATION.**

**(a) Term.** This Agreement and each Software license granted under this Agreement shall continue unless terminated under this **Section 4** ("Term and Termination").

**(b) Termination by Customer.** If the Software is licensed for a specific term as noted on an Order Form, Customer may terminate any Software license at the end of the term; however, any such termination shall not relieve Customer's obligations specified in **Section 4(d)** ("Effect of Termination").

**(c) Termination by Fiorano.** Fiorano may terminate this Agreement or any license upon written notice if Customer breaches this Agreement and fails to correct the breach within thirty (30) days of notice from Fiorano.

**(d) Effect of Termination.** Termination of this Agreement or any license shall not limit Fiorano from pursuing other remedies available to it, including injunctive relief, nor shall such termination relieve Customer's obligation to pay all fees that have accrued or are otherwise owed by Customer under any Order Form. Such rights and obligations of the parties' which, by their nature, are intended to survive the termination of this agreement shall survive such termination. Without limitation to the foregoing, these shall include rights and liabilities arising under Sections **2(a)(iii)**, **2(a)(iv)** ("Rights Granted"), **2(d)** ("Customer Specific Objects"), **4** ("Term and Termination"), **5** ("Indemnity, Warranties, Remedies"), **6** ("Limitation of Liability"), **7** ("Payment Provisions"), **8** ("Confidentiality") and **9** ("Miscellaneous") Upon termination, Customer shall cease using, and shall return or at Fiorano's request destroy, all copies of the Software and Documentation and upon Fiorano's request certify the same to Fiorano in writing within thirty (30) days of termination. In case of termination of this Agreement or any license for any reason by either party, Fiorano shall have no obligation to refund any amounts paid to Fiorano by Customer under this Agreement. Further, if Customer terminates the agreement before the expiry of a term for a term-license, then Customer shall be obliged to pay the entire license fee for the entire licensed term.

## **5. INDEMNITY, WARRANTIES, REMEDIES.**

**(a) Infringement Indemnity.** Fiorano agrees to indemnify Customer against a third party claim that any Product infringes a U.S. copyright or patent and pay any damages finally awarded, provided that: (i) Customer notifies Fiorano in writing within ten (10) days of the claim; (ii) Fiorano has sole control of the defense and all related settlement negotiations; and (iii) Customer provides Fiorano with the assistance, information and authority at no cost to Fiorano, necessary to perform Fiorano's obligations under this **Section 5** ("Indemnities, Warranties, Remedies"). Fiorano shall have no liability for any third party claims of infringement based upon (i) use of a version of a Product other than the most current version made available to the Customer, (ii) the use, operation or combination of any Product with programs, data, equipment or documentation if such infringement would have been avoided but for such use, operation or combination; or (iii) any third party software, except as the same may be integrated, incorporated or bundled by Fiorano, or its third party licensors, in the Product licensed to Customer hereunder.



If any Product is held or claimed to infringe, Fiorano shall have the option, at its expense, to (i) modify the Product to be non-infringing or (ii) obtain for Customer a license to continue using the Software. If it is not commercially reasonable to perform either of the above options, then Fiorano may terminate the license for the infringing Product and refund the pro rated amount of license fees paid for the applicable Product using a twelve (12) month straight-line amortization schedule starting on the Commencement Date. This **Section 5(a)** ("Infringement Indemnity") states Fiorano's entire liability and Customer's sole and exclusive remedy for infringement.

**(B) WARRANTIES AND DISCLAIMERS.**

(i) SOFTWARE WARRANTY. EXCEPT FOR EXCLUDED COMPONENTS WHICH ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, FOR EACH SUPPORTED SOFTWARE LICENSE WHICH CUSTOMER ACQUIRES HEREUNDER, FIORANO WARRANTS THAT FOR A PERIOD OF THIRTY (30) DAYS FROM THE COMMENCEMENT DATE THE SOFTWARE, AS DELIVERED BY FIORANO TO CUSTOMER, WILL SUBSTANTIALLY PERFORM THE FUNCTIONS DESCRIBED IN THE ASSOCIATED DOCUMENTATION IN ALL MATERIAL RESPECTS WHEN OPERATED ON A SYSTEM WHICH MEETS THE REQUIREMENTS SPECIFIED BY FIORANO IN THE DOCUMENTATION. PROVIDED THAT CUSTOMER GIVES FIORANO WRITTEN NOTICE OF A BREACH OF THE FOREGOING WARRANTY DURING THE WARRANTY PERIOD, FIORANO SHALL, AS CUSTOMER'S SOLE AND EXCLUSIVE REMEDY AND FIORANO'S SOLE LIABILITY, USE ITS REASONABLE EFFORTS, DURING THE WARRANTY PERIOD ONLY, TO CORRECT ANY REPRODUCIBLE ERRORS THAT CAUSE THE BREACH OF THE WARRANTY IN ACCORDANCE WITH ITS TECHNICAL SUPPORT POLICIES. IF CUSTOMER DOES NOT OBTAIN A SUPPORTED SOFTWARE LICENSE, THE SOFTWARE IS PROVIDED "AS IS." ANY IMPLIED WARRANTY OR CONDITION APPLICABLE TO THE SOFTWARE, DOCUMENTATION OR ANY PART THEREOF BY OPERATION OF ANY LAW OR REGULATION SHALL OPERATE ONLY FOR DEFECTS DISCOVERED DURING THE ABOVE WARRANTY PERIOD OF THIRTY (30) DAYS UNLESS TEMPORAL LIMITATION ON SUCH WARRANTY OR CONDITION IS EXPRESSLY PROHIBITED BY APPLICABLE LAW. ANY SUPPLEMENTS OR UPDATES TO THE SOFTWARE, INCLUDING WITHOUT LIMITATION, BUG FIXES OR ERROR CORRECTIONS SUPPLIED AFTER THE EXPIRATION OF THE THIRTY-DAY LIMITED WARRANTY PERIOD SHALL NOT BE COVERED BY ANY WARRANTY OR CONDITION, EXPRESS, IMPLIED OR STATUTORY.

(ii) MEDIA WARRANTY. FIORANO WARRANTS THE TAPES, DISKETTES OR ANY OTHER MEDIA ON WHICH THE SOFTWARE IS SUPPLIED TO BE FREE OF DEFECTS IN MATERIALS AND WORKMANSHIP UNDER NORMAL USE FOR THIRTY (30) DAYS FROM THE COMMENCEMENT DATE. CUSTOMER'S SOLE AND EXCLUSIVE REMEDY AND FIORANO'S SOLE LIABILITY FOR BREACH OF THE MEDIA WARRANTY SHALL BE FOR FIORANO TO REPLACE DEFECTIVE MEDIA RETURNED WITHIN THIRTY (30) DAYS OF THE COMMENCEMENT DATE.

(iii) SERVICES WARRANTY. FIORANO WARRANTS ANY SERVICES PROVIDED HEREUNDER SHALL BE PERFORMED IN A PROFESSIONAL AND WORKMANLIKE MANNER IN ACCORDANCE WITH GENERALLY ACCEPTED INDUSTRY PRACTICES. THIS WARRANTY SHALL BE VALID FOR A PERIOD OF THIRTY (30) DAYS FROM PERFORMANCE. FIORANO'S SOLE AND EXCLUSIVE LIABILITY AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY PURSUANT TO THIS WARRANTY SHALL BE USE BY FIORANO OF REASONABLE EFFORTS FOR RE-PERFORMANCE OF ANY SERVICES NOT IN COMPLIANCE WITH THIS WARRANTY WHICH ARE BROUGHT TO FIORANO'S ATTENTION BY WRITTEN NOTICE WITHIN FIFTEEN (15) DAYS AFTER THEY ARE PERFORMED.

(IV)DISCLAIMER OF WARRANTIES. SUBJECT TO LIMITED WARRANTIES PROVIDED FOR HEREINABOVE, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE, DOCUMENTATION AND SERVICES (IF ANY) ARE PROVIDED AS IS AND WITH ALL FAULTS, FIORANO HEREBY DISCLAIMS ALL OTHER WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY (IF ANY) IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF RELIABILITY OR AVAILABILITY, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, AND OF LACK OF NEGLIGENCE, ALL WITH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATION, SOFTWARE, AND RELATED CONTENT THROUGH THE SOFTWARE OR OTHERWISE ARISING OUT OF THE USE OF THE SOFTWARE. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SOFTWARE.

6. LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL FIORANO BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION, FOR PERSONAL INJURY, FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY OF GOOD FAITH OR OF REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SOFTWARE, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATION, SOFTWARE, AND RELATED CONTENT THROUGH THE SOFTWARE, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS EULA, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING NEGLIGENCE), MISREPRESENTATION, STRICT LIABILITY, BREACH OF CONTRACT OR BREACH OF WARRANTY OF FIORANO, AND EVEN IF FIORANO OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

NOTWITHSTANDING ANY DAMAGES THAT MAY BE INCURRED FOR ANY REASON AND UNDER ANY CIRCUMSTANCES (INCLUDING, WITHOUT LIMITATION, ALL DAMAGES AND LIABILITIES REFERENCED HEREIN AND ALL DIRECT OR GENERAL DAMAGES IN LAW, CONTRACT OR ANYTHING ELSE), THE ENTIRE LIABILITY OF FIORANO UNDER ANY PROVISION OF THIS EULA AND THE EXCLUSIVE REMEDY OF THE CUSTOMER HEREUNDER (EXCEPT FOR ANY REMEDY OF REPAIR OR REPLACEMENT IF SO ELECTED BY FIORANO WITH RESPECT TO ANY BREACH OF THE LIMITED WARRANTY) SHALL BE LIMITED TO THE PRO-RATED AMOUNT OF FEES PAID BY CUSTOMER UNDER THIS AGREEMENT FOR THE PRODUCT, USING A TWELVE (12) MONTH STRAIGHT-LINE AMORTIZATION SCHEDULE STARTING ON THE COMMENCEMENT DATE. FURTHER, IF SUCH DAMAGES RESULT FROM CUSTOMER'S USE OF THE SOFTWARE OR SERVICES, SUCH LIABILITY SHALL BE LIMITED TO THE PRORATED AMOUNT OF FEES PAID FOR THE RELEVANT SOFTWARE OR SERVICES GIVING RISE TO THE LIABILITY TILL THE DATE WHEN SUCH LIABILITY AROSE, USING A TWELVE (12) MONTH STRAIGHT-LINE AMORTIZATION SCHEDULE STARTING ON THE COMMENCEMENT DATE. NOTWITHSTANDING ANYTHING IN THIS AGREEMENT, THE FOREGOING LIMITATIONS, EXCLUSIONS AND DISCLAIMERS SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.

The provisions of this Agreement allocate the risks between Fiorano and Customer. Fiorano's pricing reflects this allocation of risk and the limitation of liability specified herein.

## 7. PAYMENT PROVISIONS.

**(a) Invoicing.** All fees shall be due and payable thirty (30) days from receipt of an invoice and shall be made without deductions based on any taxes or withholdings. Any amounts not paid within thirty (30) days will be subject to an immediately due and payable late payment fee equivalent to: the sum of \$50.00 plus an interest equal to the lower of (a) the maximum applicable legal interest rate, or (b) one percent (1%) per month.

**(b) Payments.** All payments made by Customer shall be in United States Dollars for purchases made in all countries except the United Kingdom or the European Union, in which case the payments shall be made in British Pounds Sterling or Euros respectively. Payments shall be directed to:

Fiorano Software, Inc.

718 University Ave.

Suite 212, Los Gatos, CA 95032

Attn: Accounts Receivable.

If the product is purchased outside the United States, payments may have to be made to an Affiliate as directed by Fiorano Software, Inc.

**(c) Taxes.** The fees listed in this Agreement or the applicable Order Form does not include Taxes. In addition to any other payments due under this Agreement, Customer agrees to pay, indemnify and hold Fiorano harmless from, any sales, use, excise, import or export, value added or similar tax or duty, and any other tax not based on Fiorano's net income, including penalties and interest and all government permit fees, license fees, customs fees and similar fees levied upon the delivery of the Software or other deliverables which Fiorano may incur in respect of this Agreement, and any costs associated with the collection or withholding of any of the foregoing items (the "Taxes").

## **8. CONFIDENTIALITY.**

**(a) Confidential Information.** "Confidential Information" shall refer to and include, without limitation, (i) the source and binary code of Products, and (ii) the business and technical information of either party, including but not limited to any information relating to product plans, designs, costs, product prices and names, finances, marketing plans, business opportunities, personnel, research, development or know-how;

**(b) Exclusions of Confidential Information.** Notwithstanding the foregoing, "Confidential Information" shall not include: (i) Information that is not marked confidential or otherwise expressly designated confidential prior to its disclosure, (ii) Information that is or becomes generally known or available by publication, commercial use or otherwise through no fault of the receiving party, (iii) Information that is known to the receiving party at the time of disclosure without violation of any confidentiality restriction and without any restriction on the receiving party's further use or disclosure; (iv) Information that is independently developed by the receiving party without use of the disclosing party's confidential information, or (v) Any Residuals arising out of this Agreement. Notwithstanding, any Residuals belonging to Source Code shall belong exclusively to Fiorano and

Customer shall not have any right whatsoever to any Residuals relating to Source Code hereunder.

- (c) **Use and Disclosure Restrictions.** During the term of this Agreement, each party shall refrain from using the other party's Confidential Information except as specifically permitted herein, and from disclosing such Confidential Information to any third party except to its employees and consultants as is reasonably required in connection with the exercise of its rights and obligations under this Agreement (and only subject to binding use and disclosure restrictions at least as protective as those set forth herein executed in writing by such employees).
- (d) **Continuing Obligation.** The confidentiality obligation described in this section shall survive for three (3) years following any termination of this Agreement. Notwithstanding the foregoing, Fiorano shall have the right to disclose Customer's Confidential Information to the extent that it is required to be disclosed pursuant to any statutory or regulatory provision or court order, provided that Fiorano provides notice thereof to Customer, together with the statutory or regulatory provision, or court order, on which such disclosure is based, as soon as practicable prior to such disclosure so that Customer has the opportunity to obtain a protective order or take other protective measures as it may deem necessary with respect to such information.

## 9. MISCELLANEOUS.

**(a) Export Administration.** Customer agrees to comply fully with all applicable relevant export laws and regulations including without limitation, those of the United States ("Export Laws") to assure that neither the Software nor any direct product thereof are (i) exported, directly or indirectly, in violation of Export Laws; or (ii) are intended to be used for any purposes prohibited by the Export Laws, including, without limitation, nuclear, chemical, or biological weapons proliferation.

**(b) U. S. Government Customers.** The Software is "commercial items," as that term is defined at 48 C.F.R. 2.101 (OCT 1995), consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (SEPT 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (JUNE 1995), all U.S. Government Customers acquire the Software with only those rights set forth herein.

**(c) Notices.** All notices under this Agreement shall be in writing and shall be deemed to have been given when mailed by first class mail five (5) days after deposit in the mail. Notices shall be sent to the addresses set forth at the beginning of this Agreement or such other address as either party may specify in writing.

**(d) Force Majeure.** Neither party shall be liable hereunder by reason of any failure or delay in the performance of its obligations hereunder (except for the payment of money) on account of strikes, shortages, riots, insurrection, fires, flood, storm, explosions, acts of God, war, governmental action, labor conditions, earthquakes, material shortages or any other cause which is beyond the reasonable control of such party.

**(e) Assignment.** Neither this Agreement nor any rights or obligations of Customer hereunder may be assigned by Customer in whole or in part without the prior written approval of Fiorano. For the avoidance of doubt, any reorganization, change in ownership or a sale of all or substantially all of Customer's assets shall be deemed to trigger an assignment. Fiorano's rights and obligations, in whole or in part, under this Agreement may be assigned by Fiorano.

**(f) Waiver.** The failure of either party to require performance by the other party of any provision hereof shall not affect the right to require such performance at any time thereafter; nor shall the waiver by either party of a breach of any provision hereof be taken or held to be a waiver of the provision itself.

**(g) Severability.** In the event that any provision of this Agreement shall be unenforceable or invalid under any applicable law or court decision, such unenforceability or invalidity shall not render this Agreement unenforceable or invalid as a whole and, in such event, any such provision shall be changed and interpreted so as to best accomplish the objectives of such unenforceable or intended provision within the limits of applicable law or applicable court decisions.

**(h) Injunctive Relief.** Notwithstanding any other provisions of this Agreement, a breach by Customer of the provisions of this Agreement regarding proprietary rights will cause Fiorano irreparable damage for which recovery of money damages would be inadequate, and that, in addition to any and all remedies available at law, Fiorano shall be entitled to seek timely injunctive relief to protect Fiorano's rights under this Agreement.

**(i) Controlling Law and Jurisdiction.** If this Software has been acquired in the United States, this Agreement shall be governed in all respects by the laws of the United States of America and the State of California as such laws are applied to agreements entered into and to be performed entirely within California between California residents. All disputes arising under this Agreement may be brought in Superior Court of the State of California in Santa Clara County or the United States District Court for the Northern District of California as permitted by law. If this Software has been acquired in any other jurisdiction, the laws of the Republic of Singapore shall apply and any disputes arising hereunder shall be subject to the jurisdiction of the courts of Singapore, Singapore. Customer hereby consents to personal jurisdiction of the above courts. The parties agree that the United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from application to this Agreement.

**(j) No Agency.** Nothing contained herein shall be construed as creating any agency, partnership or other form of joint enterprise or liability between the parties.

**(k) Headings.** The section headings appearing in this Agreement are inserted only as a matter of convenience and in no way define, limit, construe or describe the scope or extent of such section or in any way affect such section.

**(l) Counterparts.** This Agreement may be executed simultaneously in two or more counterparts, each of which will be considered an original, but all of which together will constitute one and the same instrument.

**(m) DISCLAIMER.** THE SOFTWARE IS NOT SPECIFICALLY DEVELOPED OR LICENSED FOR USE IN ANY NUCLEAR, AVIATION, MASS TRANSIT OR MEDICAL APPLICATION OR IN ANY OTHER INHERENTLY DANGEROUS APPLICATIONS. CUSTOMER AGREES THAT FIORANO AND ITS SUPPLIERS SHALL NOT BE LIABLE FOR ANY CLAIMS OR DAMAGES ARISING FROM CUSTOMER'S USE OF THE SOFTWARE FOR SUCH APPLICATIONS. CUSTOMER AGREES TO INDEMNIFY AND HOLD FIORANO HARMLESS FROM ANY CLAIMS FOR LOSSES, COSTS, DAMAGES OR LIABILITY ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SOFTWARE IN SUCH APPLICATIONS.

**(n)Customer Reference.** Fiorano may refer to Customer as a customer in sales presentations, marketing vehicles and activities. Such activities may include, but are not limited to; a press release, a Customer user story completed by Fiorano upon implementation of the Software, use by Fiorano of Customer's name, logo and other marks, together with a reasonable number of technical or executive level Customer reference calls for Fiorano.

**(o)Entire Agreement.** This Agreement, together with any exhibits, completely and exclusively states the agreement of the parties. In the event of any conflict between the terms of this Agreement and any exhibit hereto, the terms of this Agreement shall control. In the event of any conflict between the terms of this Agreement and any purchase order or Order Form, this Agreement will control, and any pre-printed terms on Customer's purchase order or equivalent document will be of no effect. This Agreement supersedes, and its terms govern, all prior proposals, agreements or other communications between the parties, oral or written, regarding the subject matter of this Agreement. This Agreement shall not be modified except by a subsequently dated written amendment signed by the parties, and shall prevail over any conflicting "pre-printed" terms on a Customer purchase order or other document purporting to supplement the provisions hereof.

---

---

## Exhibit A

### Fiorano Product List

Each of the individual items below is a separate Fiorano product (the "Product"). The Products in this list collectively constitute the Software. Fiorano reserves the right to modify this list at any time in its sole discretion. In particular, Product versions might change from time to time without notice.

1. Fiorano SOA Enterprise Server
2. Fiorano ESB Server
3. FioranoMQ Server Peer / FioranoMQ (standalone version)
4. Fiorano Peer Server
5. Fiorano SOA Tools
6. Fiorano Mapper Tool
7. Fiorano Database Business Component
8. Fiorano HTTP Business Component
9. Fiorano SMTP Business Component
10. Fiorano FTP Business Component
11. Fiorano File Business Component
12. Fiorano MOM Business Components (MQSeries, MSMQ, JMS)

---

NOTE: Other business components may be added to or removed from this list from time to time at Fiorano's sole discretion.

---

## Exhibit B

### EXCLUDED COMPONENTS

---

(a) Any third party or open source library included within the Software

---



---

## Exhibit C

**Licensing Restrictions.** The Software licensed hereunder is subject to the following licensing restrictions.

The parties understand that the modules of the Software are licensed as noted in this section. The term "Target System" means any computer system containing one or more Processors based upon any architecture, running any operating system, excluding computers running IBM MV-S, OS/390 and related "mainframe" operating systems. The Term "Processor" means a computation hardware unit such as a Microprocessor that serves as the main arithmetic and logic unit of a computer. A Processor might consist of multiple "Cores", in which case licenses shall have to be purchased on a per-Core basis. A Target System may have one or more Processors, each of which may have one or more Cores. In the sections below, Cores may replace Processors as applicable.

- (a) If the Software is Fiorano ESB Enterprise Server, FioranoMQ Peer, Fiorano SOA server or FioranoMQ Server (JMS), then the Software is licensed on a per Processor basis on a single Target System, where the total number of Processors on the Target System may not exceed the total number of Processors licensed, with the additional restriction that only a single instance of the Fiorano ESB Enterprise Server may run on a single Target System and that a separate license must be purchased for each instance of the Fiorano ESB Enterprise Server, Fiorano ESB Peer Server or FioranoMQ Server (JMS) Server for each Processor;
- (b) If the Software is Fiorano SOA Tools or Fiorano Mapper Tool , or any Fiorano Test and/or Development license, then the Software is licensed on a per-named-user basis, where the total number of named users may not exceed the total number of named users licensed;
- (c) If the Software is a Fiorano Business Component of any kind (including but not limited to Fiorano HTTP, File, SMTP, File, Database, and other Business Components, etc.), then the Software is licensed on the basis of the number of CPUs of the Target System on which the FioranoMQ Peer (to which the Business Component connects runs). A separate license needs to be purchased for each CPU of each Target System of each FioranoMQ Peer instance to which any Business Component connects.

**Evaluations.** Licenses used for evaluation cannot be used for any purposes other than an evaluation of the product. Existing customers must purchase new licenses to use additional copies of any Product and may not use evaluation keys in any form. All evaluation keys are restricted to 45-days and extensions need to be applied for explicitly. Any misuse of evaluation keys shall be subject to a charge of 125% (one hundred and twenty-five percent) of the license fee plus 20% support.

**Non-Production Environments.** For all non-production environments referenced on the Order Form (including all HA (high-availability), QA, Staging and Development environments), the following is understood: each non-production environment is an exact replica of the Production Environment from the standpoint of the number of copies of the Fiorano ESB Enterprise Server, FioranoMQ Peer, Fiorano SOA server and/or FioranoMQ Server (JMS) licensed. Each non-production environment is licensed on the exact same number and configuration of CPUs and/or Cores as the corresponding Production Environment.

**Run-Time Libraries.** The Fiorano ESB Enterprise Server, FioranoMQ Peer, Fiorano SOA server and FioranoMQ Server (JMS) products are “server” products, each of which has a runtime library associated with it. The runtime library may be freely bundled with and/or used for internal development purposes by all Users who have licensed at least one production copy of the corresponding Server Software.

Copyright (c) 1999-2010, Fiorano Software Private Limited. and Affiliates

# Contents

---

## FioranoMQ 9 High Availability ..... 21

1.1 Introduction to HA .....	21
1.1.1 Known HA implementations / solutions .....	21
2.1 Introduction to Fiorano HA .....	21
2.1.1 Terminology.....	22
2.1.2 Types of Fiorano HA .....	23
2.1.2.1 Shared HA.....	23
2.1.2.2 Replicated HA .....	23
2.1.3 Requirements.....	23
2.1.3.1 Hardware .....	23
2.1.3.2 Software .....	23
3.1 Fiorano HA Concept.....	24
3.1.1 Fiorano Replicated High Availability Working .....	24
3.1.1.1 States and state transition .....	25
3.1.1.3. How do Server State Changes?.....	26
3.1.1.4. What are the objects that are replicated? .....	27
3.1.2 Fiorano Shared High Availability Working.....	28
3.1.2.1 States and State Transition .....	28
3.1.2.3. How do failovers happens?.....	30
3.1.2.4. Database details? .....	30
3.1.3 Advantages or disadvantages / how to determine which HA to use.....	30
4.1 Configurations of HA.....	31
4.1.1 Replicated HA.....	35
4.1.1.1 Basic configurations .....	35
4.1.1.2 Advanced configurations.....	39
4.1.2 Shared HA .....	44
4.1.2.1 Basic configurations .....	44
4.1.2.2 Advanced configurations.....	50
5.1 Managing of HA server.....	55
5.1.1 Replicated HA.....	55
5.1.1.1 Launching the servers .....	55
5.1.1.2 Stopping the servers .....	57
5.1.1.3 Determining status of servers.....	58
5.1.1.4 Log details.....	59
5.1.1.5 Verifying HA Setup .....	59
5.1.2 Shared HA .....	60
5.1.2.1 Launching the servers .....	60
5.1.2.2 Stopping the servers .....	62
5.1.2.3. Determining status of servers.....	63

5.1.2.4. Log details.....	63
5.1.2.5 Verify HA Setup.....	64
6.1 Troubleshooting.....	65
7.1 Best practices.....	68
8.1 FAQ .....	68

# FioranoMQ 9 High Availability

---

## 1.1 Introduction to HA

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 down within seconds. Hence, it is absolutely imperative for the messaging backbone 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 backup should be automatic and transparent to the client application.

### 1.1.1 Known HA implementations / solutions

To enable any HA implementation one needs to provide certain guarantees. In case active server is down, second server needs to take all the existing connections and undelivered messages must be delivered. To achieve this task, the state information of servers must be exchanged between the two servers and database used by servers must be consistent. In order to make database consistent there are two methods one can use. First one is replicated database, where both servers have their own copy of database but on any changes to active server's database those changes are propagated to passive server's database. In this way consistency of these two servers is maintained. In the second implementation one can use a common shared database for both the servers. In this case only the active server makes changes to the common database. As this database is also available to passive server on failure of active server (say primary), when passive server (say secondary) takes over, it has all the changes made by Primary server.

## 2.1 Introduction to Fiorano HA

FioranoMQ Server (FioranoMQ Server) is designated with backup server when it runs on High Availability mode, which is started along with the primary FioranoMQ Server. In case the primary server is unavailable 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.

The Enterprise Server represents a Highly Available entity, which appears as a single FioranoMQ Server to its applications. JMS application connects to the primary FioranoMQ server when initializing, 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; the client application should not concern about reconnecting logic in its code as it is handled by FioranoMQ's runtime internally.

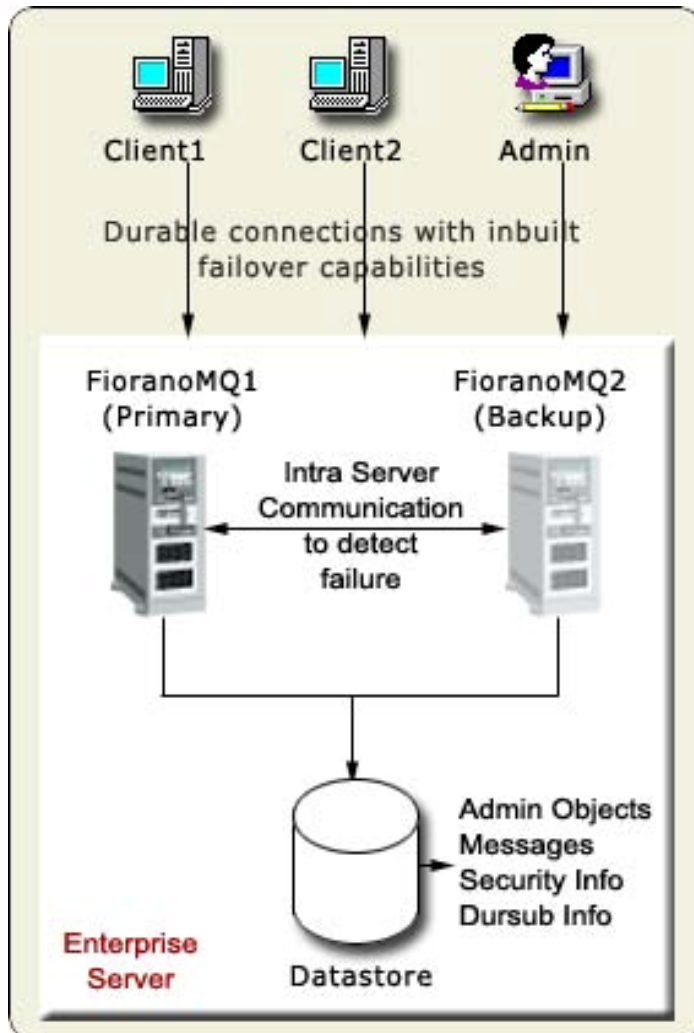


Figure 1: Enterprise Server

### 2.1.1 Terminology

**Primary Server:** The server running with Primary Profile.

**Secondary Server:** The server running with Secondary Profile.

**Failover:** The process of shifting from the primary server to the backup server or vice versa is termed as Failover.

**HA Layer configuration:** The HA Layer configuration deals with configuring the peer server details. The important configuration parameters defined in this layer are Primary (true or false), Lockfile, BackupHAIPort, BackupHAIPAddress, GatewayServerIPAddress, and GatewayServerPort. This configuration is required to be done in both replication and shared HA modes.

## 2.1.2 Types of Fiorano HA

FioranoMQ provides complete flexibility to administrators. There are two modes in which HA can be run:

1. Shared HA
2. Replicated HA

### 2.1.2.1 Shared HA

In this mode the database is shared between the active and passive servers. Whenever the active server goes down, the passive server becomes active and uses the database that is being shared.

### 2.1.2.2 Replicated HA

In this mode, the database is replicated from active server to the passive server. Also, whenever a server is disconnected from the network and reconnects, the database of the active server is replicated into the passive server. This data include message data, admin/security data as well as status information whatever changes are made when the passive server was down.

## 2.1.3 Requirements

Fiorano's HA solution is purely implemented in software and is not dependent on expensive hardware solutions.

### 2.1.3.1 Hardware

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. This hardware can also be avoided by using either replication support or using a central RDBMS server as the message store in the Enterprise Server.

### 2.1.3.2 Software

It can run on any java supported platform.

## 3.1 Fiorano HA Concept

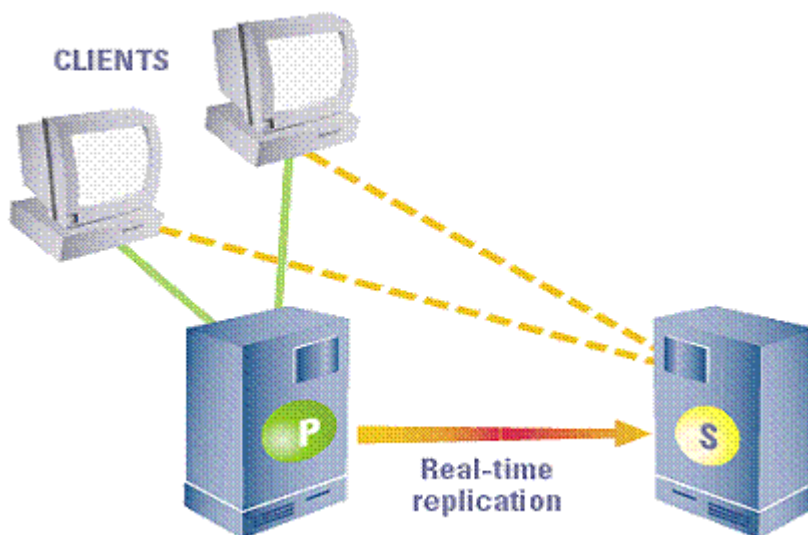
### HA Locking Mechanism

HA locking mechanism is employed by the servers in replicated mode to determine the server state in case, a server of the pair is unavailable or if the network fails. A read and write permissions file is shared on a machine this file is referred as the **LockFile**. The machine hosting the LockFile is referred to as the gateway machine. A server can switch to Active only if it holds a lock over the **LockFile**.

In HA implementation prior to the locking mechanism, a network link failure between the servers could have led to both servers switching to Standalone state. Since the lock can be held by only one server at a time, it prevents both servers from switching to Active/Standalone state.

The locking mechanism makes the state switching of a HA server more deterministic.

### 3.1.1 Fiorano Replicated High Availability Working



The central concept of backchannel replication is that the Active Server (the server which is in the Active State) replicates its data store and state to the Passive Server, thus keeping both servers in sync. This replication channel is supported on a private network dedicated to the synchronization of the broker state and messaging data.



In Replicated HA, the database is replicated from the ACTIVE server to the PASSIVE server. This data include messages, admin/security data as well as status information whatever changes are made when the passive server was down. Whenever a server is disconnected from the network and reconnects back, the database of the active server is replicated into the passive server's database. In this case, both servers have their own copy of database but on any changes to active server's database those changes are propagated to passive server's database. In this way consistency of these two servers is maintained.

The passive server accepts no client connection while in its hot-standby (passive) role, but is prepared to immediately transition to the Active role as soon as it detects that the Active Server is unavailable. If the primary fails, all Fiorano applications fail over from the primary and reconnect to the designated secondary backup broker.

The primary and secondary broker-pair use the replication channel to routinely seek the heartbeat of the other and watch for any interruption in the data flow or connection to switch states. A locking mechanism (as already explained) is also employed to determine the state of the servers.

This Hot-failover process is immediate and is completely transparent to all client applications. The Secondary Server in the active role is sensitive to re-establishment of the replication channel. This reconnection may come from a recovery of the Primary Server or from a replacement Primary Server. Once the primary comes up again, it assumes the role of the Passive Server since the Secondary Server switched to Active Server).

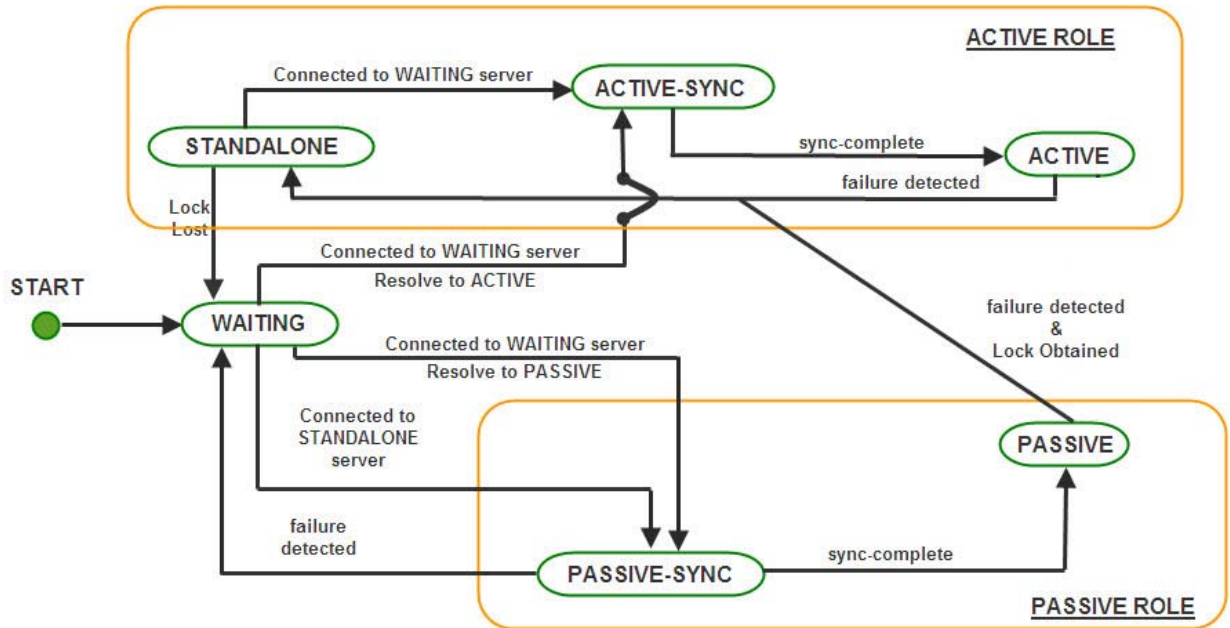
**Important:** If servers are running in Replicated HA mode, it becomes essential to have a reliable network for database synchronization calls.

### 3.1.1.1 States and state transition

The following states occur in different phases of the servers in replication mode.

- **ACTIVE:** refers to normal working state. In this state the server accepts client connections.
- **PASSIVE:** in this state the server monitors its active peer server and is in standby mode.
- **ACTIVE\_TRANSITION\_STATE:** this occurs while the server is synchronizing with the standby server and at the same time serving client applications.
- **PASSIVE\_TRANSITION\_STATE:** this occurs on standby server side while the active server is synchronizing with the standby server.
- **WAITING:** this occurs when the server is waiting for the state of other server/ trying to acquire lock on the lock file to become active. The server does not accept client connections in this state.
- **STANDALONE:** this occurs in a server when it is actively servicing client and the other server is disconnected.
- **DEAD** - Indicates that the server is down/not present in network.

The following diagram explains the transition to various states:



**Note:**

- failure detected – refers to the link between the servers being broken
- sync-complete – database synchronization complete
- Lock Lost – lock over the LockFile is lost
- Lock Obtained – lock obtained over the LockFile
- Resolve to Active/Passive – based on which server obtains the lock

**3.1.1.3. How do Server State Changes?**

1. On startup, the Server enters into **WAITING** state. In this state, the server is waiting for its backup server to connect to it. This is the initial synchronization state, which is required to sync up the primary server with the secondary to avoid any message loss. This server will change state if one of the following occurs.
  - **Switch to PASSIVE SYNC state:** If the HA channel is established and the other server is in STANDALONE state.
  - **Switch to PASSIVE(STANDBY) SYNC or ACTIVE SYNC state:** If the HA channel is established and the other server is also in WAITING state, then the servers assumes themselves as being in Active or Passive roles depending on the Repository Timestamps and remote server status.

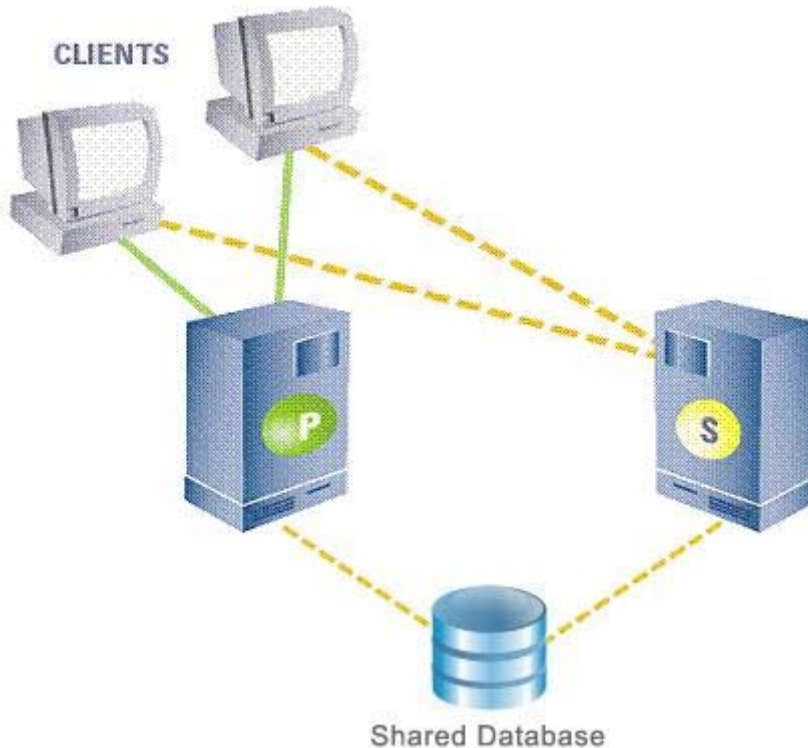
2. When the Server is actively serving clients and its backup server is not running or if the HA transport channel is broken and it has the lock over the **Lockfile**, then the state of the active server is **STANDALONE**. If the server in **STANDALONE** state establishes the HA channel and the other server is in WAITING state, then the **STANDALONE** server shifts to **ACTIVE SYNC** state and further to ACTIVE state. However, a passive (standby) server can switch to STANDALONE if the other server is not running or if the transport channel is broken and passive server acquires the lock over the LockFile.
3. When the Server is in **ACTIVE SYNC** state, the server starts synchronizing its data with the backup server which is in PASSIVE SYNC state. The Server in **ACTIVE SYNC** continues to serve its clients. Completion of the Runtime Synchronization Protocol causes a transition of the **ACTIVE SYNC** server to the **ACTIVE** state and the server in **PASSIVE SYNC** state to PASSIVE state.
4. Once the **ACTIVE SYNC** Server completes the synchronization, it enters into the ACTIVE state and resumes actively transmitting state information and all replication data onto the **PASSIVE** server. At this point, if there is a failure of the **ACTIVE** server, the Hot Standby **PASSIVE** server is ready to move into the **STANDLONE** state and starts accepting requests from the clients.
5. An ACTIVE server can switch to **WAITING** if the transport channel is broken and it loses the lock over the **LockFile**. Similarly, a **STANDALONE** server can switch to **WAITING** if it loses the lock over the LockFile.
6. Whenever there is a change in the server state, it broadcasts the present and previous state to the Backup Server. The Servers transition is a function of its own state, the present and previous state of the Backup Server and whether or not it holds the lock over the LockFile.

#### 3.1.1.4. What are the objects that are replicated?

In the database following objects need to be replicated on Primary and Secondary server's machine:

- PTP objects
- ACL objects
- Principal objects
- PubSub objects
- Admin objects

### 3.1.2 Fiorano Shared High Availability Working



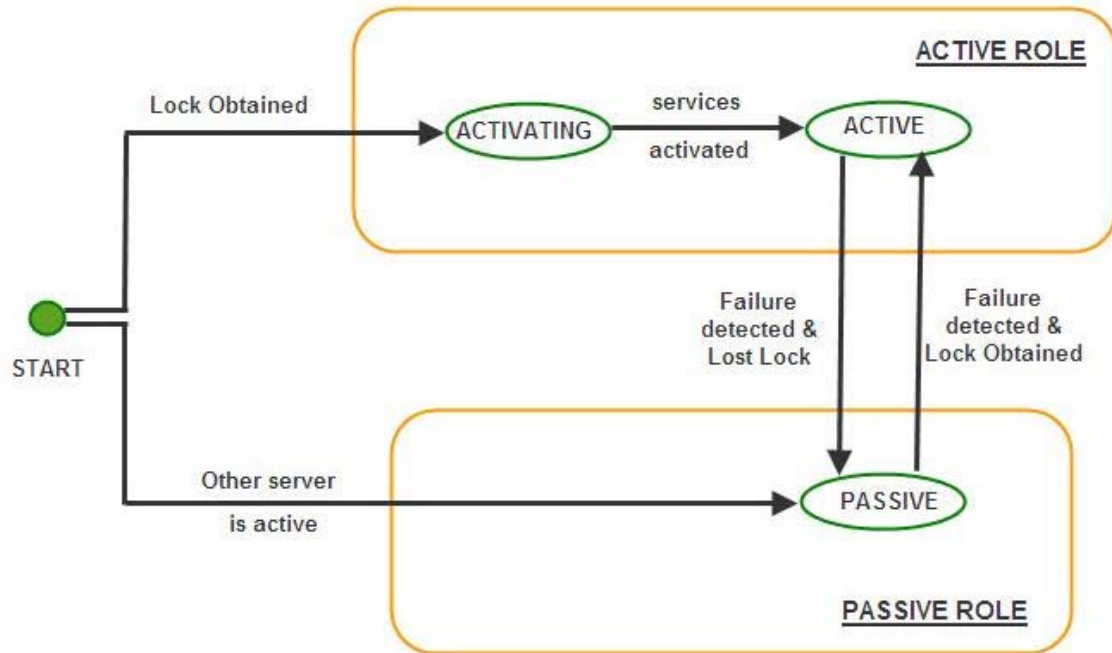
In this mode of High availability, the database is shared between the active and passive servers and do not replicate data over the network. In this, only the active server makes changes to the common database. If the Active fails, all Fiorano applications fail over from the Active and reconnect to the designated Passive backup broker. The Active and Passive broker-pair use the network channel between them to routinely seek the heartbeat of the other and watch for any break in connection to switch states. A locking mechanism (as explained in section 3.1) is also employed to determine the state of the servers. The database which is common to both the servers is referred to as the **shared database**.

**Note:** The shared database connectivity is critical for the servers to function, as the servers store all data in it. It is mandatory for the **Shared Database** to be always accessible to the servers. Unavailability of the shared database could lead to data loss and data corruption.

#### 3.1.2.1 States and State Transition

- **ACTIVE:** refers to normal working state. In this state the server accepts connections
- **PASSIVE:** in this state the server monitors its active peer server and is in standby mode.
- **ACTIVATING:** this occurs while the server is in transition to become active and synchronizing with the database.

The following diagram explains the transition to various states:



**Note:**

- failure detected – refers to the link between the servers being broken
- Lock Lost – lock over the LockFile is lost
- Lock Obtained – lock obtained over the LockFile

When the server starts up, the server tries to acquire a lock on the lock file. If it acquires the lock successfully, it switches to the ACTIVATING state. It then switches to ACTIVE state once all its services have been activated. Unlike in replicated HA, where the servers wait for each other to come up (that is, in WAITING state), a server in shared mode does not need to wait for its backup server to come up because they share a common database and no database synchronization is required which is the case for servers working in replicated mode.

After switching to ACTIVE state, the server keeps trying to connect to its backup server. If the backup server starts up, the backup server switches to PASSIVE state.

At this point, if there is a failure of the ACTIVE server, the Hot Standby PASSIVE Server is ready to move into the ACTIVE state and starts accepting requests from the clients.

### 3.1.2.3. How do failovers happens?

In case the primary server becomes unavailable, all the client applications connected to it are automatically reconnected to the secondary server. The process of shifting from the primary server to the backup server or vice versa is transparent to the application. The client application should not be concern about writing reconnect logic in its 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.

### 3.1.2.4. Database details?

In shared HA mode database is shared between active and passive servers.

## 3.1.3 Advantages or disadvantages / how to determine which HA to use

Both the HA implementations have their own advantages and disadvantages over each-other. These are pointed out here:

In Replicated HA implementation there are two copies of database available, so in case of any database failure on one system, up-to-date database is available on second system. But for Shared HA implementation there is just one database available. So any corruption to this database leads to degradation of system performance.

- In Shared HA implementation, for each database store operation it needs just one disk-write. But for Replicated HA implementation for each operation on database it needs two disk-writes (one each for both the databases on Primary and secondary). Thus performance will be better with shared HA implementation.
- In Shared HA implementation, just one datastore is used contrary to replicated HA, where two datastores are used. So, Cost of storage is less in shared HA implementation. But, for the normal operation of FioranoMQ it needs not much storage space. Thus, storage factor cannot be considered as major one, while deciding which HA implementation to use.

From above discussion we can infer that, when reliability is preferred over performance use Replicated HA implementation and use Shared HA implementation when performance is preferred over reliability.

## 4.1 Configurations of HA

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 and 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 and 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

To launch the server on one of these profiles, use the following command:

```
fmq.bat -profile %SELECTED_PROFILE%
```

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

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

**Note:** This document would 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.

Following steps need to be followed for configuring any HA profiles

1. Start Fiorano Studio. Select Tools >Configure FioranoMQ from the menu bar.
2. Select the directory pointing to %SELECTED\_HA\_PROFILE% from the dialog box and click the Open button.
3. Right-click on the selected profile, select FioranoMQ Replicated HA or FioranoMQ Shared HA option. User now can see a new window appeared with all the required configurable parameters. Now, configure all the required parameters.

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

**Configure BackupHAIPAddress & BackupHAPort of remote Server:** HA Layer of an FioranoMQ server connects to the backup server's HA Layer. In order to configure the back-up server's IP and Port, navigate to the node specified below and set the value of BackupHAIPAddress and BackupHAPort displayed in the Properties of FioranoHAKRPCProvider pane.

```
%SELECT_HA_PROFILE%>Fiorano -> HAKRPCProvider -> FioranoHAKRPCObject
```

The default profiles have Backup HA IP address configured to localhost and BackupHA Port 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.

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

**Configure self - Port:** This port refers to the 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 configure Self port, navigate to the node specified below and modify the value of Port in the Properties of FioranoHAConnectionManager pane

```
%SELECTED_HA_PROFILE% >Fiorano -> FioranoHAConnectionManager
```

**Note:** A change in this parameter would require modifying BackupHAPort Configuration in the peer server.

**Configure Lock file:** Full path of the file which will be used for determining the HA states. This file should be present in the third machine and always be available to the HA servers with all permissions. The LockFile machine must be same as the machine whose IPAddress is given as GatewayServerIPAddress. In order to configure Lock file, navigate to the node specified below and modify the value of LockFile in the Properties of FioranoHAManager pane. Path to Lock file:

```
%SELECTED_HA_PROFILE% >Fiorano -> FioranoHAManager
```

A file is created and the directory containing this file is shared with read/write permissions. The lock file, if present on a machine having UNIX/Solaris operating system should be shared by using the NFS protocol - version 4. If the lock file is present on Windows it should be shared using the Samba Protocol. (The lock file can be shared on Windows using NFS – version 4, if it supports it.)

**Setup:**

OS hosting the Lock File & Protocol Used	OS hosting the servers
Windows - Samba	Windows / Linux
Linux - NFSv4	Linux / Solaris 8,9,10
Solaris - NFSv4	Linux / Solaris 8,9,10

**Note:** The user has to make sure that, the operating system hosting the server supports the protocol used for sharing the lock file. The LockFile and the directory containing the LockFile should have read/write permissions set. On Operating Systems other than Windows, one can verify the permissions using the ls -l command.

On Windows Operating System, the directory on the gateway machine containing the LockFile should be mapped to a network drive.



**Example:**

If the directory containing the lock file on Windows, is shared using the samba protocol, this directory should be mapped to a network drive on the Windows machine hosting the server. Let us say the shared directory is mapped to a drive letter 'Z:/' and the lock file is lock.lck , the lock file path now becomes 'Z:/lock.lck'. This path should be used to set the LockFile parameter while configuring the profile via Fiorano Studio.

On non Windows Operating System, the directory on the gateway machine containing the LockFile should be mounted on the machine hosting the server.

**Examples:**

If the lock file is present in a windows samba share & is mounted at /home/user/db on the machine hosting the server then the lock file path on the machine hosting the server would be /home/user/db/lock.lck. This absolute path should be used to set the LockFile parameter while configuring the profile via Fiorano Studio.

**Example Mount Command:**

'mount -t cifs -o rw //<gatewayIP>/<sharename> <path on local machine where the directory has to be mounted.>' If the lock file 'lock.lck' is shared using NFS v4 and the directory db is mounted at /home/user/db on the machine hosting the server then the lock file path on the machine hosting the server would be /home/user/db/lock.lck. This absolute path should be used to set the LockFile parameter while configuring the profile via Fiorano Studio.

**Example Mount Command:**

'mount -t nfs4 -o rw <gatewayIP>:/ <path on local machine where the directory has to be mounted.>'

**Note:** The newly added mount point will disappear after system reboot. The user has to make sure that the mount is automated on system reboot. One can refer to the URL '[http://www.brennan.id.au/19-Network\\_File\\_System.html#nfs4](http://www.brennan.id.au/19-Network_File_System.html#nfs4)' on how to share a directory using NFS v4.

**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 (preferably to the physical router machine on the network). In order to configure Gateway Server, navigate to the node specified below and modify the value of GatewayServerIP Address and GatewayServerPort in the Properties of FioranoHAManager pane

```
%SELECTED_HA_PROFILE% >Fiorano -> FioranoHAManager
```

**Note:** Gateway machine server should be configured as the machine which hosts the lock file. This is essential for the functionality of the HA server to work.

**Configure isPrimary:** Set Primary to true for primary FioranoMQ Server and false for backup Server by modifying its value as shown in the Properties panel by clicking the node specified below.

```
%SELECTED_HA_PROFILE%->Fiorano->HA->HAManager•FioranoHAManager
```

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

**Common Configuration:** HA requires enabling Durable Connections, Pinging and Auto Revalidation support in FioranoMQ Server. This can be done by modifying the parameters specified below in the Properties panel.

`%SELECTED_HA_PROFILE%->Fiorano-->etc -> FmqConfigLoader`

- Set AllowDurableConnections Property to true.
- Set PingEnabled to true.
- Set EnableAutoRevalidation to true

**Note:** The pre-created profiles already come with this configuration and hence no change would be required when working on them.

Right-click on the `%SELECTED_HA_PROFILE%` root node present under the Explorer tab on the left hand side of the Studio and select the Save option to save the configuration.

**Example Configuration:**

Parameter	Primary Server	Backup Server
url HAKRPCProvider	http://164.164.128.128:1856	http://164.164.128.108:1956
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 DurableConnections	True	true
PingEnabled	True	true
HAManager isPrimary	True	false
Gateway IP Address	164.164.128.108	164.164.128.128
LockFile	/home/fmq/Lock.dat	/home/fmq/Lock.dat

After changing the HA Layer's configuration, the server's existing database should be cleared before restarting. 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, in Windows:

- `clearDB.bat FioranoMQ_HA_rpl\HAPrimary`
- `clearDB.bat FioranoMQ_HA_rpl\HASsecondary`

On the client-side the primary server URL has to be added as provider URL and secondary server URL has to be added to the backup URLs'.

## 4.1.1 Replicated HA

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).

### 4.1.1.1 Basic configurations

One can configure replicated HA profile through changing Configs.cfg file (text based configuration) or through Studio. Following section describe these options.

#### Text based

Profiles can now be configured through a text based properties file. Profiles can be found under the %FIORANO\_HOME%\fmq\profiles\FioranoMQ\_HA\_rpl directory (say, PROFILE\_HOME). In order to configure replicated HA profile, edit Configs.cfg file under %PROFILE\_HOME%\HAPrimary\conf and %PROFILE\_HOME%\HASsecondary\conf.

1. The default entries in Configs.cfg file looks like:

```
ObjectName=Container
# HA Common Properties.
#
#FMQGroup.BackupServerIp=localhost
#FMQGroup.BackupServerPort=1956
#FMQGroup.BackupHAPort=3000
#FMQGroup.BackupHAIPAddress=localhost
#FMQGroup.Port=2000
#FMQGroup.PingInterval=30000
#FMQGroup.LockFile=null
#FMQGroup.SocketCreationTimeout=10000
#FMQGroup.GatewayServerIPAddress=localhost
#FMQGroup.GatewayServerPort=7
```

In front of every parameter, its default values are given.

2. Take the scenario of running HAPrimary on a machine with IPAddress: 192.169.1.22 and HASsecondary on a machine with IPAddress: 192.169.1.200. GatewayServer address is given as 192.169.1.16 which is present in the network. If LockFile, say Lock.dat, has been shared from third system and mounted to /home/fmq1 on 192.168.1.22 system and to /home/fmq2 on 192.168.1.200 system, then on Primary profile these parameters should takes following value:

```
ObjectName=Container
# HA Common Properties.
#
FMQGroup.BackupServerIp=192.168.1.200
#FMQGroup.BackupServerPort=1956
#FMQGroup.BackupHAPort=3000
FMQGroup.BackupHAIPAddress=192.168.1.200
#FMQGroup.Port=2000
#FMQGroup.PingInterval=30000
FMQGroup.LockFile=/home/fmq1/Lock.dat
#FMQGroup.SocketCreationTimeout=10000
```

```
FMQGroup.GatewayServerIPAddress=192.168.1.16
#FMQGroup.GatewayServerPort=7
```

On Secondary profile these parameters should takes following value:

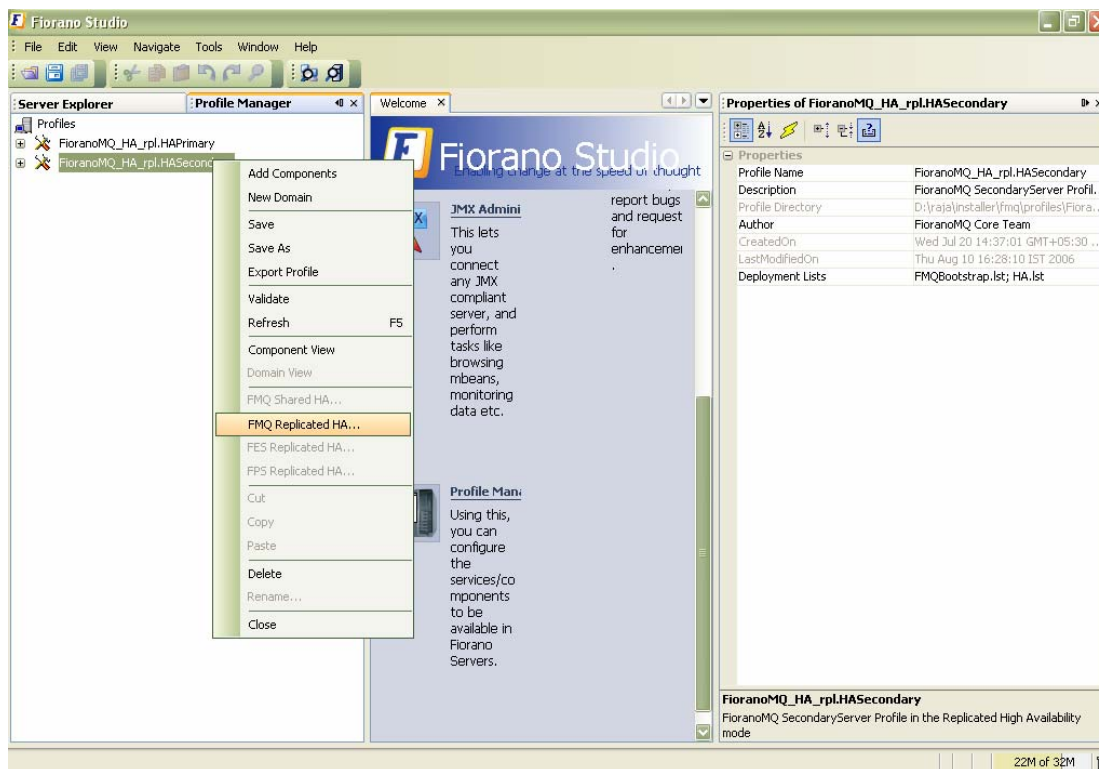
```
ObjectName=Container
# HA Common Properties
#
FMQGroup.BackupServerIp=192.168.1.22
#FMQGroup.BackupServerPort=1856
FMQGroup.BackupHAIPAddress=192.168.1.22
#FMQGroup.BackupHAPort==2000
#FMQGroup.Port=3000
#FMQGroup.PingInterval=30000
FMQGroup.GatewayServerIPAddress=192.168.1.16
#FMQGroup.GatewayServerPort=7
FMQGroup.LockFile=/home/fmq2/Lock.dat
#FMQGroup.SocketCreationTimeout=10000
```

Modified Configs.cfg file need to be saved for both Primary and secondary profiles.

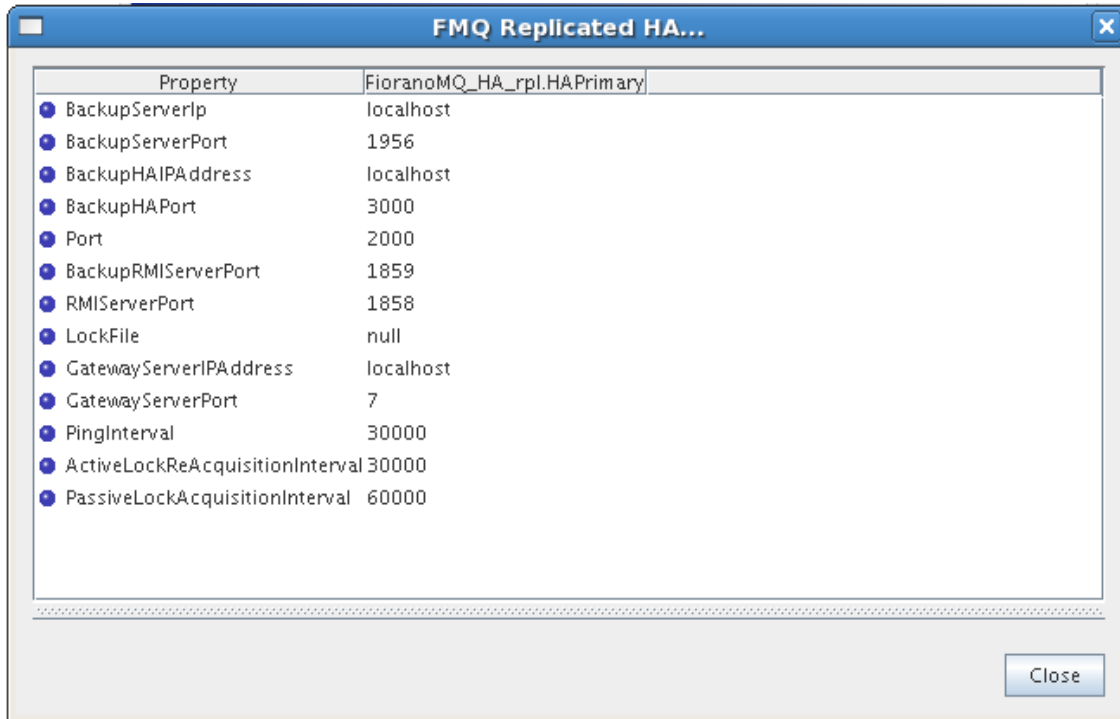
## Studio

**Note:** Here, %SELECTED\_HA\_PROFILE% refers to FioranoMQ\_HA\_rpl\HAPrimary or FioranoMQ\_HA\_rpl\HASsecondary.

If replicated HA profile needs to be used, then open FioranoMQ\_HA\_rpl\HAPrimary and FioranoMQ\_HA\_rpl\HASsecondary profile in Studio Profile Manager. Right-click on the profile and click on FMQ Replicated HA. The FMQ Replicated HA wizard appears.



You can configure all the required values from the wizard. After configuring the wizard, right-click the node and select Save option from the pop-up menu



The following properties are available in the FMQ HA replicated wizard.

### **BackupHAIPAddress**

This IP represents the URL where another FioranoMQ HA server runs. In case this HA server fails then all the required synchronized data is present in backup HA server and HA layer connects to backup HA server.

### **BackupHAPort**

This port represents the backup machine HA port on which another FioranoMQ Server runs.

### **Port**

This port represents the FMQ Server HA port on which this HA Server runs.

### **LockFile**

Full path of the file which will be used for determining the HA states. This file should be present in the third machine and always available to the HA servers with all permissions. The LockFile machine must be same as the machine whose IPAddress is given as GatewayServerIPAddress.

### **GatewayServerIPAddress**

This value represents the IP address of third machine which is always present in the network. This value must not be the machine where one of the FioranoMQ servers runs.

### GatewayServerPort

This value represents the port to which any of the HA Servers pings. This port must not be used for some other purpose.

### ActiveLockAcquisitionInterval:

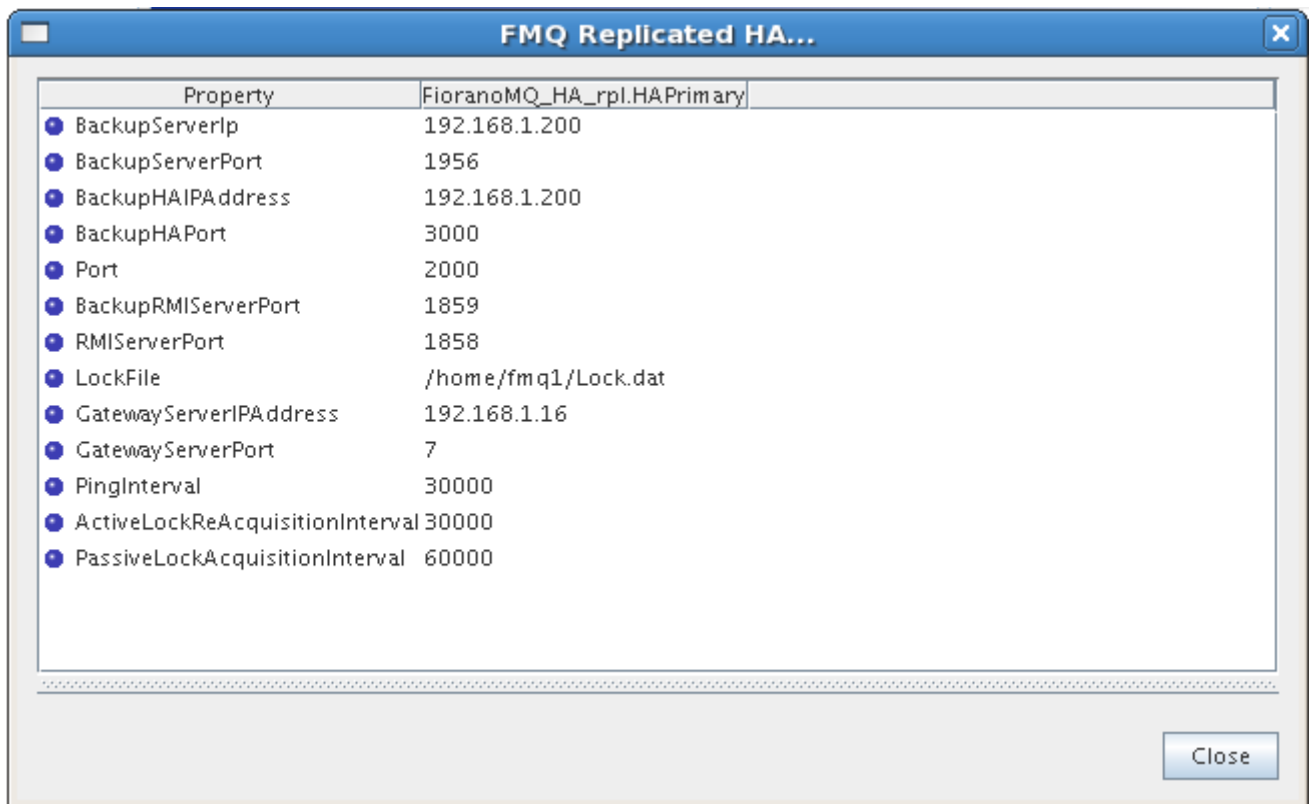
This parameter indicates the wait Interval between each attempt to acquire the lock for active server. This value should be in multiples of pingInterval/2 otherwise server may try to acquire the lock on the next multiple of pingInterval/2.

### PassiveLockAcquisitionInterval:

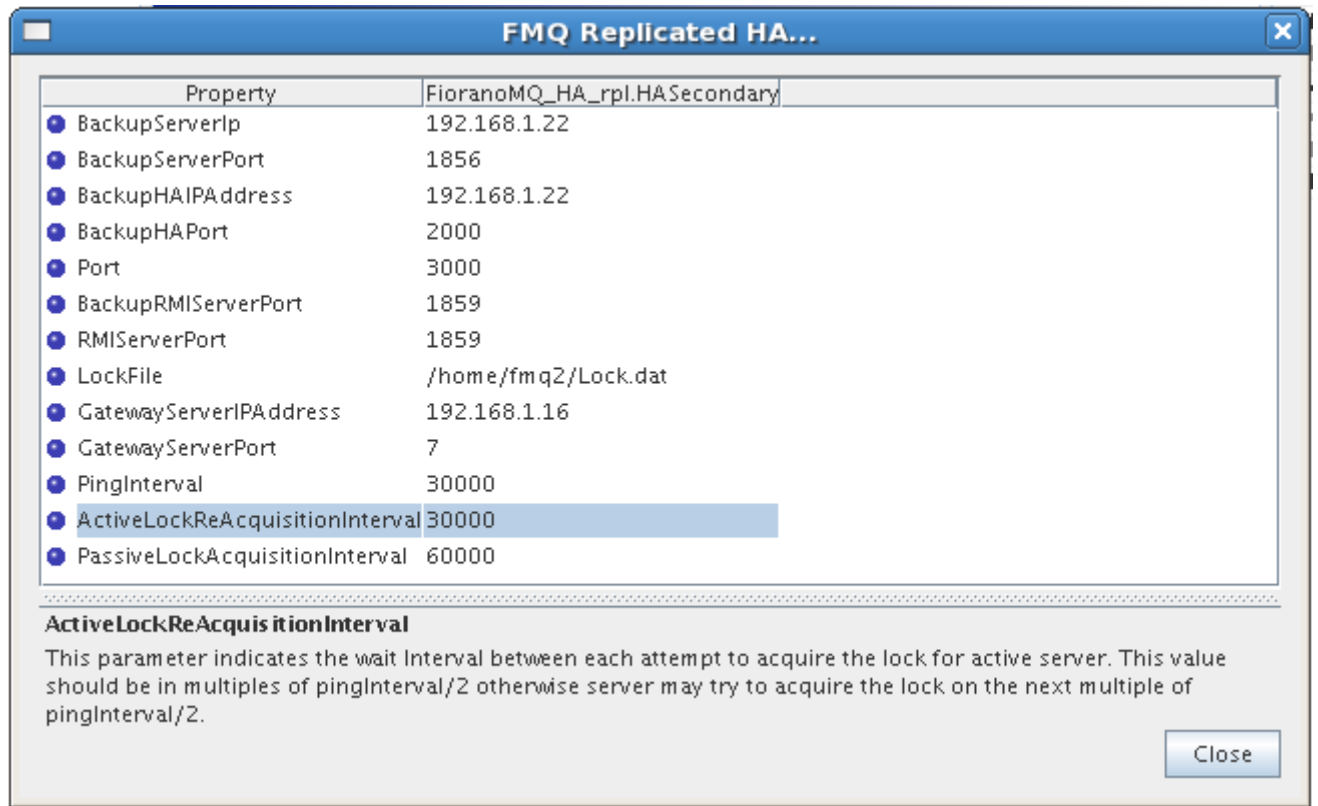
This parameter indicates the wait Interval for the passive server to acquire the lock when the link between active and passive server is down. This value should be greater than '2\*ActiveLockReacquisitionInterval', otherwise an exception will be thrown. Server won't start if this value is not set properly.

All IP addresses, ports should represent the correct values. Without modifying any of these values HAPrimary and HASecondary can run on a single machine.

Take the scenario of running HAPrimary on a machine with IPAddress: 192.169.1.22 and HASecondary on a machine with IPAddress: 192.169.1.200. Gatewayserver address is given as 192.169.1.16 which is present in the network. If LockFile, say Lock.dat, has been shared from third system and mounted to /home/fmq1 on 192.168.1.22 system and to /home/fmq2 on 192.168.1.200 system, then HAPrimary wizard looks like:



And HASecondary wizard looks like



#### 4.1.1.2 Advanced configurations

##### Text based

Edit Configs.cfg file under:

`%FIORANO_HOME%\fmq\profiles\FioranoMQ_HA_rpl\HAPrimary\conf` for HAPrimary and

`%FIORANO_HOME%\fmq\profiles\FioranoMQ_HA_rpl\HASecondary\conf` for HASecondary

##### 1. Port

Specify the Port on which HA Manager would listen for connection from the peer HA Server. Apart from the port that the HA Server uses to listen for connections from its clients, it must also listen for a connection from its HA Peer Server, in order to know the status of its peer.

This parameter is used to specify that port.

##### Valid values:

Default value is 2000 for Primary and 3000 for secondary profile.

**Note:** The port must be not in use by any other services.

##### Configuration:

In Configs.cfg file this property can be found under HA common Properties. For example: `#FMQGroup.Port=2000`

User can change this value and for taking changes into effect uncomment the value, like- `FMQGroup.Port=2500`

## 2. **NagleAlgo**

Boolean determining whether Nagle's Algorithm is enabled or not in socket creation with peer HA Server. Nagle's algorithm is used in TCP/IP networks for congestion control. It works by coalescing a number of small outgoing messages, and sending them all at once. Specifically, as long as there is a sent packet for which the sender has received no acknowledgment, the sender should keep buffering its output until it has a full packet's worth of output, so that output can be sent all at once.

### **Valid values:**

Default value is no

- yes - NagleAlgo is enabled
- no - NagleAlgo is not enabled

### **Configuration:**

In `Configs.cfg` file this property can be found under HA Connection Manager Settings. For example:

```
#NagleAlgo=false
```

User can change this value and for taking changes into effect uncomment the value, like-

```
NagleAlgo=true
```

## 3. **SocketInitializationTimeout**

Time (in milliseconds) within any client socket should identify itself and able to exchange the version number with server. The version number exchange is an authentication mechanism that FioranoMQ employs.

### **Valid values:**

Default value is 60000 (which is equal to a minute).

### **Configuration:**

In `Configs.cfg` file this property can be found under HA Connection Manager Settings. For example:

```
#SocketInitializationTimeout=60000
```

User can change this value and for taking changes into effect uncomment the value, like- `SocketInitializationTimeout=65000`

## 4. **ActiveLockReAcquisitionInterval**

This parameter indicates the wait Interval between each attempt to acquire the lock for active server. This value should be in multiples of `pingInterval/2` otherwise server may try to acquire the lock on the next multiple of `pingInterval/2`. Default value for this parameter is 30000.

### **Configuration:**

In `Configs.cfg` file this property can be found under HA Manager Settings. For example:

```
#ActiveLockReAcquisitionInterval=30000
```



User can change this value and for taking changes into effect uncomment the value, like-

```
ActiveLockReAcquisitionInterval=35000
```

#### 5. **PassiveLockReAcquisitionInterval**

This parameter indicates the wait Interval for the passive server to acquire the lock when the link between active and passive server is down. This value should be greater than '2\*ActiveLockReacquisitionInterval', otherwise an exception will be thrown. Server won't start if this value is not set properly. Default value for this parameter is 60000.

##### **Configuration:**

In Configs.cfg file this property can be found under HA Manager Settings. For example:

```
#PassiveLockAcquisitionInterval=60000
```

User can change this value and for taking changes into effect uncomment the value, like-

```
PassiveLockAcquisitionInterval=65000
```

#### 6. **PingInterval**

This interval represents the duration after which HAServer pings the other HABackup server to find whether the server is running properly or not.

##### **Configuration:**

In Configs.cfg file this property can be found under HA Common Properties. For example:

```
#FMQGroup.PingInterval=30000
```

User can change this value and for taking changes into effect uncomment the value, like-

```
FMQGroup.PingInterval=35000
```

#### 7. **BackupRMI ServerPort**

Port of backup server on which Mx4j RMIConnector server will bind.

##### **Configuration:**

In Configs.cfg file this property can be found under HA KRPC Provider Settings. For example:

```
#BackupRMIServerPort=1858
```

User can change this value and for taking changes into effect uncomment the value, like-

```
BackupRMIServerPort=1958
```

#### 8. **RMI ServerPort**

Port on which Mx4j RMIConnector server will bind.

##### **Configuration:**

In Configs.cfg file this property can be found under RMI Connector Settings. For example:

```
#RMIServerPort=1859
```

User can change this value and for taking changes into effect uncomment the value, like-

```
RMIServerPort=1959
```

#### 9. **MaxWaitTimeout**

Whenever an owning server realizes that there has been a change in its peer server's state, it must handle the change in its peer state by changing its state appropriately. While the owning server is changing state it starts a timer. If for some reason the state change is unable to complete after a reasonable amount of time then the owning server must realize that.

This parameter specifies the maximum waiting timeout in milliseconds for which the local server should wait for state transition to complete.

##### **Configuration:**

In Configs.cfg file this property can be found under HA Manager Settings. For example:

```
#MaxWaitTimeout=30000
```

User can change this value and for taking changes into effect uncomment the value, like-

```
MaxWaitTimeout=35000
```

### **Studio**

All the properties listed in this section are described in text based configuration.

#### 1. **Port**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

#### 2. **NagleAlgo**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

#### 3. **SocketInitializationTimeout**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

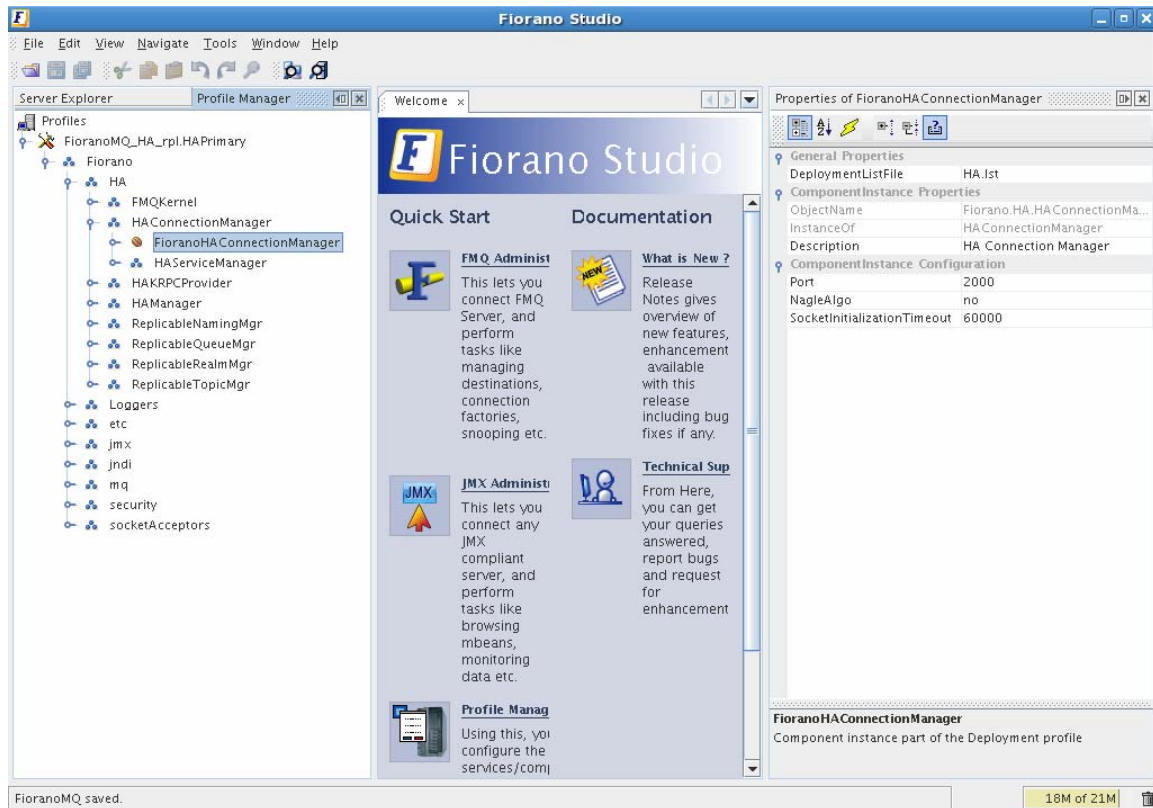


Figure: Configuring Port, NagleAlgo and SocketInitializationTimeout parameters from studio

#### 4. ActiveLockReAcquisitionInterval

This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.

#### 5. PassiveLockReAcquisitionInterval

This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.

#### 6. PingInterval

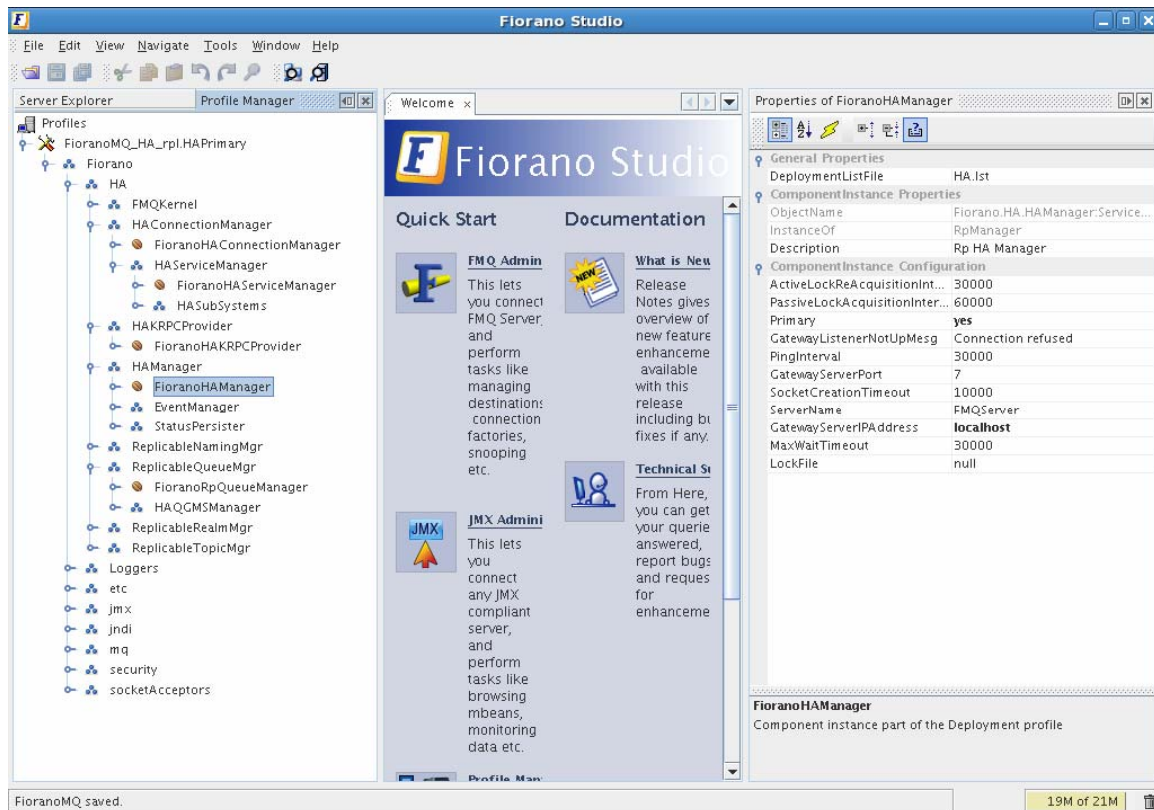
This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.

#### 7. MaxWaitTimeout

This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.



## 8. BackupRMI ServerPort

This property can be found under:

Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config through studio.

## 9. RMI ServerPort

This property can be found under:

Fiorano->jmx->RMIBasedJMXConnector->config through studio.

## 4.1.2 Shared HA

### 4.1.2.1 Basic configurations

One can configure Shared HA profile through changing Configs.cfg file (text based configuration) or through Studio. Following section describe these options.

#### Text based

Profiles can now be configured through a text based properties file. Profiles can be found under the %FIORANO\_HOME%\fmq\profiles\FioranoMQ\_HA\_shared directory (say, PROFILE\_HOME). In order to configure shared HA profile, edit Configs.cfg file under %PROFILE\_HOME%\HAPrimary\conf and %PROFILE\_HOME%\HASsecondary\conf.

1. The default entries in Configs.cfg file looks like:

```

ObjectName=Container
# HA Common Properties.
#
#FMQGroup.BackupServerIp=localhost
#FMQGroup.BackupServerPort=1956
#FMQGroup.BackupHAIPAddress=localhost
#FMQGroup.BackupHAPort=3000
#FMQGroup.Port=2000
#FMQGroup.PingInterval=30000
#FMQGroup.GatewayServerIPAddress=localhost
#FMQGroup.GatewayServerPort=7
#FMQGroup.LockFile=null
#FMQGroup.SocketCreationTimeout=10000
    
```

In front of every parameter, its default values are given.

2. Take the scenario of running HAPrimary on a machine with IPAddress: 192.169.1.22 and HASecondary on a machine with IPAddress: 192.169.1.200. GatewayServer address is given as 192.169.1.16 which is present in the network. If LockFile, say Lock.dat, has been shared from third system and mounted to /home/fmq1 on 192.168.1.22 system and to /home/fmq2 on 192.168.1.200 system, then on Primary profile these parameters should takes following value:

```

ObjectName=Container
# HA Common Properties.
#
FMQGroup.BackupServerIp=192.168.1.200
#FMQGroup.BackupServerPort=1956
FMQGroup.BackupHAIPAddress=192.168.1.200
#FMQGroup.BackupHAPort=3000
#FMQGroup.Port=2000
#FMQGroup.PingInterval=30000
FMQGroup.GatewayServerIPAddress=192.168.1.16
#FMQGroup.GatewayServerPort=7
FMQGroup.LockFile=/home/fmq1/Lock.dat
#FMQGroup.SocketCreationTimeout=10000
    
```

On Secondary profile these parameters should takes following value:

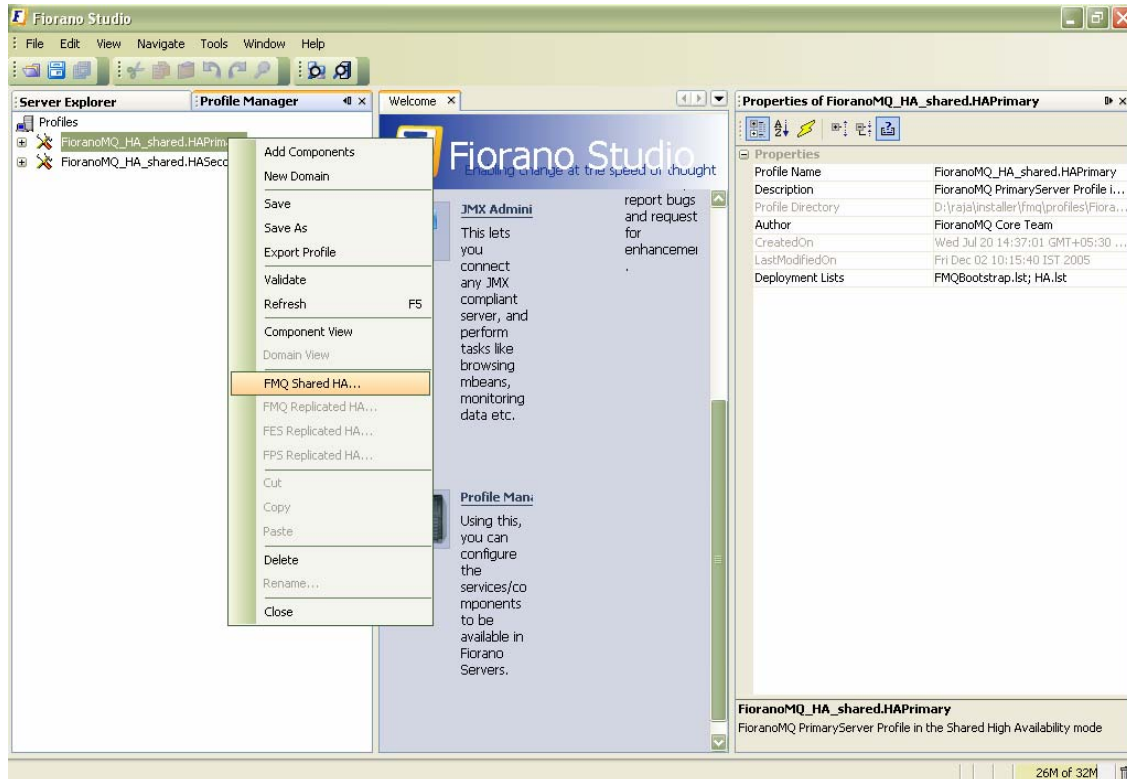
```

ObjectName=Container
# HA Common Properties
#
FMQGroup.BackupServerIp=192.168.1.22
#FMQGroup.BackupServerPort=1856
FMQGroup.BackupHAIPAddress=192.168.1.22
#FMQGroup.BackupHAPort==2000
#FMQGroup.Port=3000
#FMQGroup.PingInterval=30000
FMQGroup.GatewayServerIPAddress=192.168.1.16
#FMQGroup.GatewayServerPort=7
FMQGroup.LockFile=/home/fmq2/Lock.dat
#FMQGroup.SocketCreationTimeout=10000
    
```

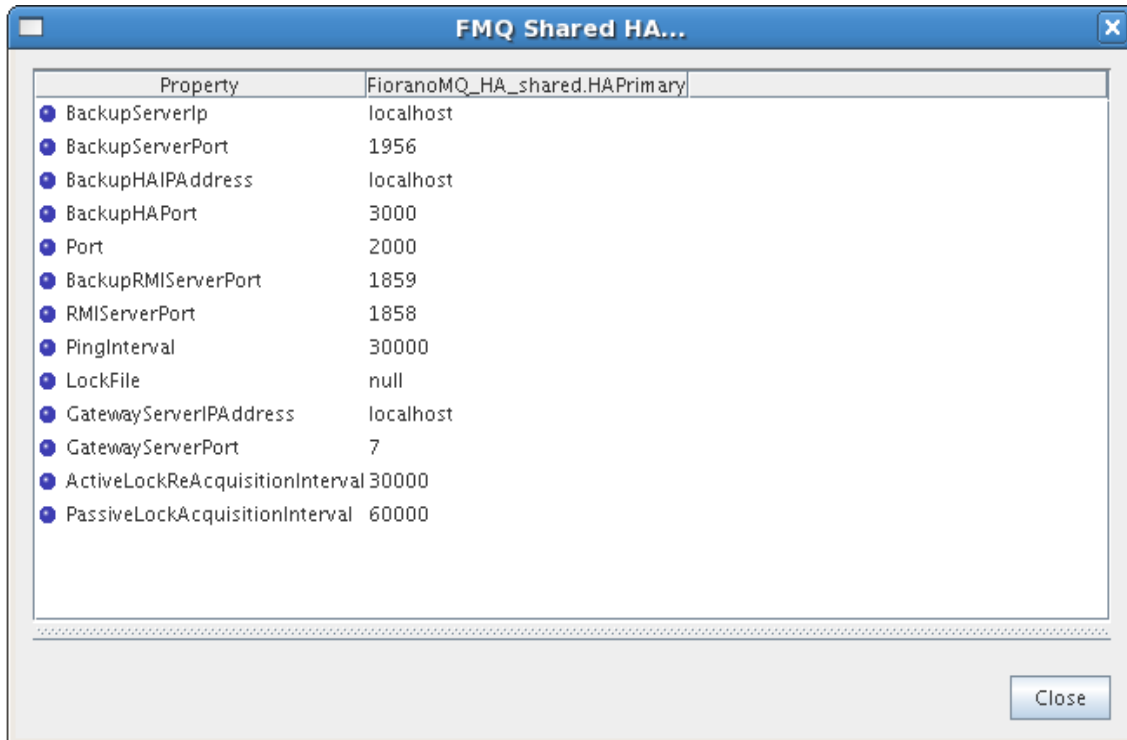
Modified Configs.cfg file need to be saved for both Primary and secondary profiles.

## Studio

If shared HA needs to be used, then open FioranoMQ\_HA\_shared/HAPrimary and FioranoMQ\_HA\_shared/HASecondary profile in Studio Profile Manager. Right-click on the profile and click on FMQ Shared HA. The FMQ Shared HA wizard appears.



You can configure all the required values from the wizard. After configuring the wizard, right-click the node and select Save option from the pop-up menu



The following properties are available in the FMQ HA replicated wizard.

#### **BackupHAIPAddress**

This IP represents the URL where another FioranoMQ HA server runs. In case this HA server fails then all the required synchronized data is present in backup HA server and HA layer connects to backup HA server.

#### **BackupHAPort**

This port represents the backup machine HA port on which another FioranoMQ Server runs.

#### **Port**

This port represents the FMQ Server HA port on which this HA Server runs.

#### **BackupRMI ServerPort**

Port of backup server on which Mx4j RMIConnector server will bind.

#### **RMI ServerPort**

Port on which Mx4j RMIConnector server will bind.

### **PingInterval**

This interval represents the duration after which HAServer pings the other HABackup server to find whether the server is running properly or not.

### **LockFile**

Full path of the file which will be used for determining the HA states. This file should be present in the third machine and always available to the HA servers with all permissions. The LockFile machine must be same as the machine whose IPAddress is given as GatewayServerIPAddress.

### **GatewayServerIPAddress**

This value represents the IP address of third machine which is always present in the network. This value must not be the machine where one of the FioranoMQ servers runs.

### **GatewayServerPort**

This value represents the port to which any of the HA Servers pings. This port must not be used for some other purpose.

### **ActiveLockReAcquisitionInterval**

This parameter indicates the wait Interval between each attempt to acquire the lock for active server. This value should be in multiples of pingInterval/2 otherwise server may try to acquire the lock on the next multiple of pingInterval/2. Default value for this parameter is 30000.

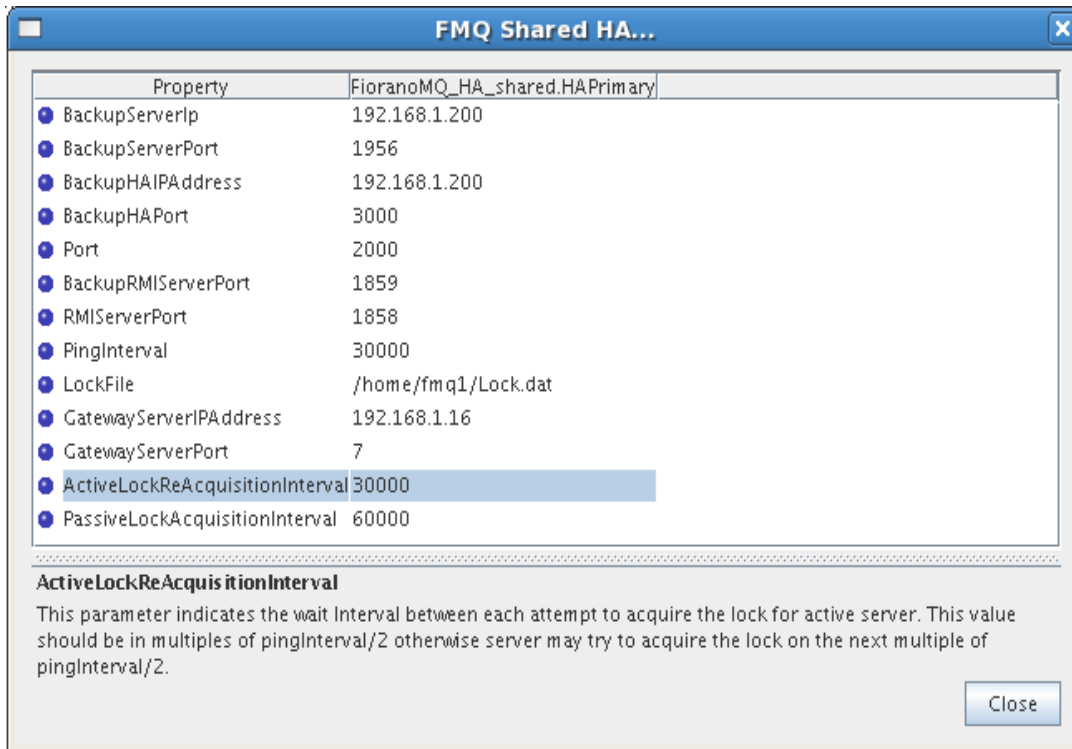
### **PassiveLockReAcquisitionInterval**

This parameter indicates the wait Interval for the passive server to acquire the lock when the link between active and passive server is down. This value should be greater than '2\*ActiveLockReacquisitionInterval', otherwise an exception will be thrown. Server won't start if this value is not set properly. Default value for this parameter is 60000.

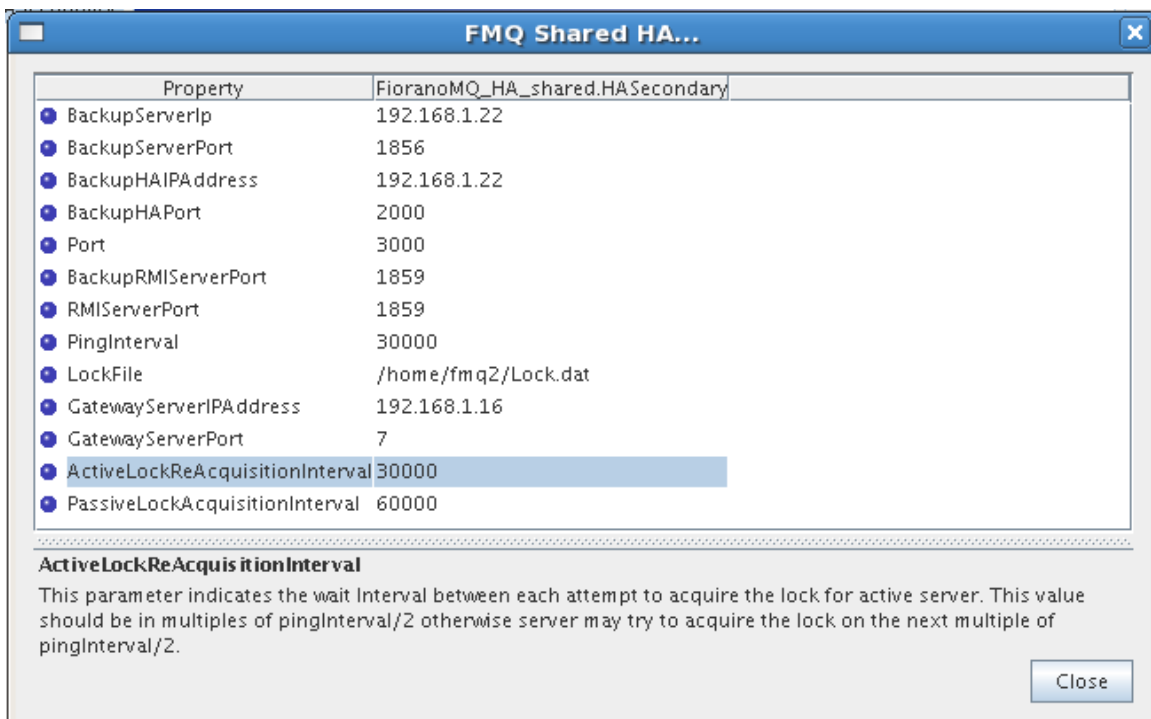
All IP addresses, ports should represent the correct values. Without modifying any of these values HAPrimary and HASecondary can run on a single machine.

Take the scenario of running HAPrimary on a machine with IPAddress: 192.169.1.22 and HASecondary on a machine with IPAddress: 192.169.1.200. Gatewayserver address is given as 192.169.1.16 which is present in the network. If LockFile, say Lock.dat, has been shared from third system and mounted to /home/fmq1 on 192.168.1.22 system and to /home/fmq2 on 192.168.1.200 system, then HAPrimary wizard looks like:





And HASecondary wizard looks like:



### 4.1.2.2 Advanced configurations

#### Text based

Edit Configs.cfg file under

%FIORANO\_HOME%\fmq\profiles\FioranoMQ\_HA\_shared\HAPrimary\conf for HAPrimary and

%FIORANO\_HOME%\fmq\profiles\FioranoMQ\_HA\_shared\HASsecondary\conf for HASsecondary.

#### 1. Port

Specify the Port on which HA Manager would listen for connection from the peer HA Server. Apart from the port that the HA Server uses to listen for connections from its clients, it must also listen for a connection from its HA Peer Server, in order to know the status of its peer.

This parameter is used to specify that port.

##### Valid values:

Default value is 2000 for Primary and 3000 for secondary profile.

**Note:** The port must be not in use by any other services.

##### Configuration:

In Configs.cfg file this property can be found under HA common Properties. For example: #FMQGroup.Port=2000

User can change this value and for taking changes into effect uncomment the value, like- FMQGroup.Port=2500

#### 2. NagleAlgo

Boolean determining whether Nagle's Algorithm is enabled or not in socket creation with peer HA Server. Nagle's algorithm is used in TCP/IP networks for congestion control. It works by coalescing a number of small outgoing messages, and sending them all at once. Specifically, as long as there is a sent packet for which the sender has received no acknowledgment, the sender should keep buffering its output until it has a full packet's worth of output, so that output can be sent all at once.

##### Valid values:

Default value is no

- yes - NagleAlgo is enabled
- no - NagleAlgo is not enabled

##### Configuration:

In Configs.cfg file this property can be found under HA Connection Manager Settings. For example: #NagleAlgo=false

User can change this value and for taking changes into effect uncomment the value, like- NagleAlgo=true

#### 3. SocketInitializationTimeout

Time (in milliseconds) within any client socket should identify itself and able to exchange the version number with server. The version number exchange is an authentication mechanism that FioranoMQ employs.

**Valid values:**

Default value is 60000 (which is equal to a minute).

**Configuration:**

In Configs.cfg file this property can be found under HA Connection Manager Settings. For example: `#SocketInitializationTimeout=60000`

User can change this value and for taking changes into effect uncomment the value, like- `SocketInitializationTimeout=65000`

4. **ActiveLockReAcquisitionInterval**

This parameter indicates the wait Interval between each attempt to acquire the lock for active server. This value should be in multiples of `pingInterval/2` otherwise server may try to acquire the lock on the next multiple of `pingInterval/2`. Default value for this parameter is 30000.

**Configuration:**

In Configs.cfg file this property can be found under HA Manager Settings. For example: `#ActiveLockReAcquisitionInterval=30000`

User can change this value and for taking changes into effect uncomment the value, like- `ActiveLockReAcquisitionInterval=35000`

5. **PassiveLockReAcquisitionInterval**

This parameter indicates the wait Interval for the passive server to acquire the lock when the link between active and passive server is down. This value should be greater than `'2*ActiveLockReacquisitionInterval'`, otherwise an exception will be thrown. Server won't start if this value is not set properly. Default value for this parameter is 60000.

**Configuration:**

In Configs.cfg file this property can be found under HA Manager Settings. For example: `#PassiveLockAcquisitionInterval=60000`

User can change this value and for taking changes into effect uncomment the value, like- `PassiveLockAcquisitionInterval=65000`

6. **PingInterval**

This interval represents the duration after which HAServer pings the other HABackup server to find whether the server is running properly or not.

**Configuration:**

In Configs.cfg file this property can be found under HA Common Properties. For example: `#FMQGroup.PingInterval=30000`

User can change this value and for taking changes into effect uncomment the value, like- `FMQGroup.PingInterval=35000`

7. **BackupRMI ServerPort**

Port of backup server on which Mx4j RMIConnector server will bind.

**Configuration:**

In Configs.cfg file this property can be found under HA KRPC Provider Settings. For example: `#BackupRMI ServerPort=1858`

User can change this value and for taking changes into effect uncomment the value, like- BackupRMIServerPort=1958

#### 8. **RMI ServerPort**

Port on which Mx4j RMIConnector server will bind.

##### **Configuration:**

In Configs.cfg file this property can be found under RMI Connector Settings. For example: #RMIServerPort=1859

User can change this value and for taking changes into effect uncomment the value, like- RMIServerPort=1959

### **Studio**

All the properties listed in this section are described in text based configuration.

#### 1. **Port**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

#### 2. **NagleAlgo**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

#### 3. **SocketInitializationTimeout**

This property can be found under:

Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config through studio.

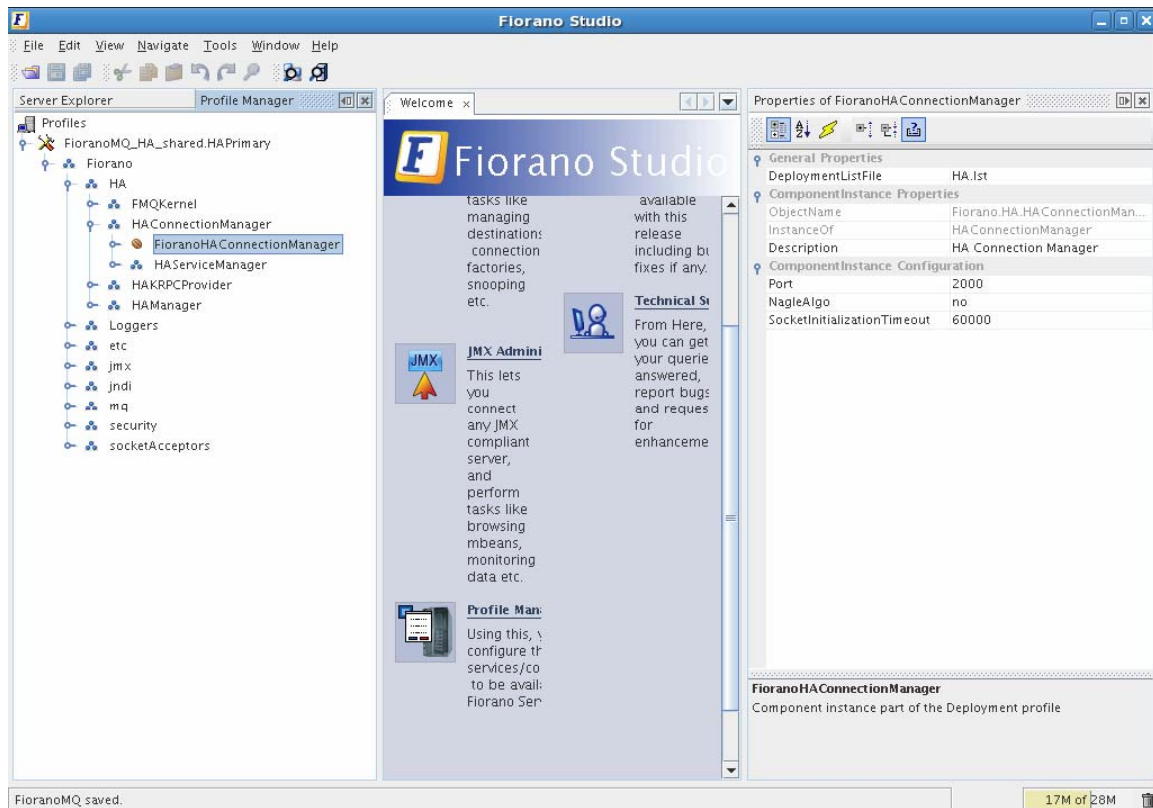


Figure: Configuring Port, NagleAlgo and SocketInitializationTimeout parameters from studio

#### 4. ActiveLockReAcquisitionInterval

This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.

#### 5. PassiveLockReAcquisitionInterval

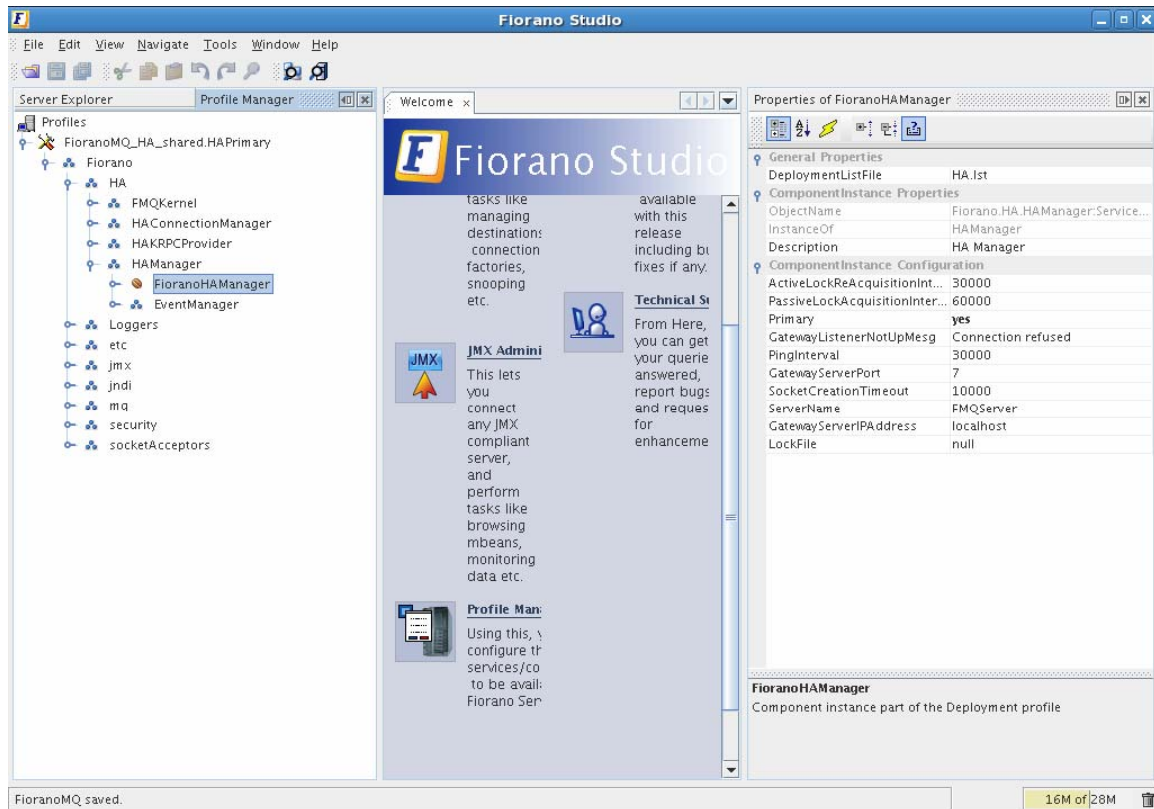
This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.

#### 6. PingInterval

This property can be found under:

Fiorano->HA->HAManager->FioranoHAManager->config through studio.



### 7. BackupRMI ServerPort

This property can be found under:

Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config through studio.

### 8. RMI ServerPort

This property can be found under:

Fiorano->jmx->RMIBasedJMXConnector->config through studio.

## 5.1 Managing of HA server

### 5.1.1 Replicated HA

#### 5.1.1.1 Launching the servers

While launching the Enterprise Server, the launch sequence within the two servers is not important. Either server (primary or backup) can be started first. On start up, the servers establish a communication path between each other in order to exchange information regarding proper functioning of all servers. The servers can be launched by using the `fmq` (bat/sh) script (available in the `fmq/bin` directory of the installation package), and supplying the `%SELECTED_HA_PROFILE%` as an argument. Different methods to run HA profiles are described here.

##### 5.1.1.1.1 Console

###### Windows:

When launching the Primary HA Server in Replicated Mode, the command line would be:

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

For Secondary HA server in Replicated Mode, the command line would be:

```
fmq.bat -profile FioranoMQ_HA_rpl/HASecondary
```

###### Unix:

When launching the Primary HA Server in Replicated Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_rpl/HAPrimary -nobackground
```

For Secondary HA server in Replicated Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_rpl/HASecondary -nobackground
```

##### 5.1.1.1.2 Background

###### Unix:

When launching the Primary HA Server in Replicated Mode, the command line would be:

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

For Secondary HA server in Replicated Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_rpl/HASecondary
```

##### 5.1.1.1.3 Service

HA replicated profiles can be made to run as service. Detailed procedure based on platform is provided here.

## Windows:

Any profile present under %FMQ\_DIR%/profiles can be made to run as NT service.

Steps for installing HA profile as NT Service:

1. Open a command prompt
2. Goto to %FMQ\_DIR%\bin\service directory
3. To install FioranoMQ Kernel as NT service, run the command  
install-fmq-service.bat -profile FioranoMQ\_HA\_rpl/HAPrimary or  
install-fmq-service.bat -profile FioranoMQ\_HA\_rpl/HASecondary
4. After installation you should be able to see "FioranoMQ <%PROFILE\_NAME%>" NT service installed.
5. Start the NT Service. Logs for NT service run can be viewed in %FMQ-DIR%\bin\service\logs\%PROFILE\_NAME%.log.

## Linux:

For enabling Replicated HA profiles as service, there is a script "fmq-linux" available in fmq/bin/service directory of FioranoMQ installation directory. Follow the procedures listed here:

1. Set JAVA\_HOME at line 8.
2. Set the FIORANO\_HOME at line 10.
3. fmq-linux by default runs FioranoMQ Profile, to run Primary profile change the name of the profile on line 20 to FioranoMQ\_HA\_shared/HAPrimary. To run Secondary profile, change the name of the profile on line 20 to FioranoMQ\_HA\_shared/HASecondary.
4. Give Executable Permission, "chmod +x fmq-linux".
5. To start as a service, type "./fmq-linux start".
6. To stop the server, type "./fmq-linux stop".
7. To run this as a service with system startup, type "chkconfig --add fmq-linux". This will launch the service whenever the system is started.

## SOLARIS:

For enabling Replicated HA profiles as service, there is a script "fmq-solaris" available in fmq/bin/service directory of FioranoMQ installation directory. Follow the procedures listed here:

1. Set the FIORANO\_HOME at line 9 of fmq-solaris file.
2. fmq-solaris by default runs FioranoMQ Profile, to run Primary profile change the name of the profile on line 19 to FioranoMQ\_HA\_rpl/HAPrimary. To run Secondary profile, change the name of the profile on line 19 to FioranoMQ\_HA\_rpl/HASecondary.
3. Copy the file fmq-solaris in "/etc/init.d".
4. Give Executable Permission, "chmod +x fmq-solaris".
5. To start as a service, type "./fmq-solaris start".
6. To stop the server, type "./fmq-solaris stop".



7. To run this as a service with system startup create the links from "/etc/init.d" using the following commands

To start the service at startup:

```
In -s fmq-solaris /etc/rc3.d/S86fmq-solaris
```

**Note:** If the priority level '86' which is specified in the link already exists in the run level (rc3.d), then select a number which doesn't exist there.

**Example:** If you want to set the priority 71 then create the link using the following command in "/etc/init.d"

```
In -s fmq-solaris /etc/rc3.d/S71fmq-solaris
```

8. Set JAVA\_HOME in <FIORANO\_HOME>/fiorano\_vars.sh

**Note:** Even if you set JAVA\_HOME in your .bashrc or .bash\_profile, it won't be set at startup.

So JAVA\_HOME must be set in "fiorano\_vars.sh" script file.

### 5.1.1.2 Stopping the servers

#### 5.1.1.2.1 Console

##### Windows:

When stopping the Primary HA Server in Replicated Mode with RMIPort 1858, the command line would be:

```
shutdown-fmq.bat -rmiPort 1858
```

For Secondary HA server in Replicated Mode with RMIPort 1859, the command line would be:

```
shutdown-fmq.bat -rmiPort 1859
```

**Note:** For shutting down both the Primary and Secondary HA servers in a single request, the command line would be: shutdown-fmq.bat -rmiPort 1858 -ha

This will send a "Shutdown" request to the backup server first and then shuts down the primary server. Any errors during the execution will be logged on to the console.

##### Unix:

When stopping the Primary HA Server in Replicated Mode with RMIPort 1858, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1858
```

For Secondary HA server in Replicated Mode with RMIPort 1859, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1859
```

**Note:** For shutting down both the Primary and Secondary HA servers in a single request, the command line would be: ./shutdown-fmq.sh -rmiPort 1858 -ha

This will send a "Shutdown" request to the backup server first and then shuts down the primary server. Any errors during the execution will be logged on to the console.

#### 5.1.1.2.2 Background

##### Unix:

When stopping the Primary HA Server in Replicated Mode with RMI Port 1858, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1858
```

For Secondary HA server in Replicated Mode with RMI Port 1859, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1859
```

**Note:** For shutting down both the Primary and Secondary HA servers in a single request, the command line would be: `./shutdown-fmq.sh -rmiPort 1858 -ha`

This will send a "Shutdown" request to the backup server first and then shuts down the primary server. Any errors during the execution will be logged on to the console.

#### 5.1.1.2.3 Service

##### Windows:

1. To remove FioranoMQ Kernel NT service, run the command

```
uninstall-fmq-service.bat -profile %PROFILE_NAME%
```

2. Un-Install should stop and remove NT service for given profile from the services list.

##### Linux:

To remove initially created service type "chkconfig --del fmq-linux". This will remove the previously added service

##### SOLARIS:

To stop the service at shutdown/reboot type:

```
ln -s fmq-solaris /etc/rc3.d/K45fmq-solaris
```

**Note:** It is recommended to choose low priority level for shutdown.

#### 5.1.1.3 Determining status of servers

It is easy to determine the status of server while running through console, as status of servers are printed on console. It can also be determined by running `ServerStatus.java` file available in `fmq/samples/JMX` directory of FioranoMQ installation directory. Details about running this file are available in `readme` file in the same directory.

### 5.1.1.4 Log details

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

### 5.1.1.5 Verifying HA Setup

On starting the Fiorano MQ Server that is part of an HA pair, the server prints debug information about its own state (ACTIVE, PASSIVE, and WAITING). It also prints information about its backup server's state whenever it detects a change.

Example Statements on console:

```
[Sat Feb 07 14:03:50 IST 2009] Old status of remote server = DEAD
[Sat Feb 07 14:03:50 IST 2009] New status of remote server = WAITING
[Sat Feb 07 14:03:50 IST 2009] Old status of remote server = DEAD
[Sat Feb 07 14:03:50 IST 2009] New status of remote server = WAITING
[Sat Feb 07 14:03:50 IST 2009] Old status of remote server = WAITING
[Sat Feb 07 14:03:50 IST 2009] New status of remote server = PASSIVE TRANSITION STATE
```

The Console includes statements as shown below:

'Primary Server switched to ACTIVE' or 'Secondary Server switched to PASSIVE', which indicate that the pair has successfully connected. Also, a statement gets printed when the lock is successfully acquired over the LockFile on the active servers console.

**Example:** Successfully acquired lock on: C:\lockFile.dat.

The figure below illustrates a successfully started Fiorano HA MQ Server:

```
E:\Fiorano\FioranoMQ9.1.0\fmq\bin>fmq.bat -profile FioranoMQ_HA_rpl\HAPrimary
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home        : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile          : E:\Fiorano\FioranoMQ9.1.0\fmq\profiles\FioranoMQ_HA_rpl\HAPrimary
Server's Process ID : 416
Created MBeanServer with ID: 57b5c33:125e25d250b:-8000:sreenivas-lap:1
[30/Dec/2009 17:32:53] license INFO The fiorano-mq9.lic license for the product
also, Node locked = false
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1958
[Wed Dec 30 17:33:04 PST 2009] Primary Server switched to WAITING
Backup Server not Connected. Check configuration or Start backup Server ...
Profile ..\profiles\FioranoMQ_HA_rpl\HAPrimary successfully deployed on Wed Dec 30 17:33:04 P
[Wed Dec 30 17:33:19 PST 2009] Old status of remote server = DEAD
[Wed Dec 30 17:33:19 PST 2009] New status of remote server = WAITING
Successfully acquired lock on ::C:\lockFile.dat
Max Memory allocated to the Server : 254 MB.
Dashboard Listening Port: 1780
Fiorano Server accepting connections at http://0.0.0.0:1856
Server Protocol:: TCP, Default:: true
Maximum Client Connections : 1,024
FioranoMQ 9.1.0, Build # 9195
[Wed Dec 30 17:33:33 PST 2009] Primary Server switched to ACTIVE
[Wed Dec 30 17:33:34 PST 2009] Old status of remote server = WAITING
[Wed Dec 30 17:33:34 PST 2009] New status of remote server = PASSIVE
```

## 5.1.2 Shared HA

### 5.1.2.1 Launching the servers

While launching the Enterprise Server, the launch sequence within the two servers is not important. Either server (primary or backup) can be started first. On start up, the servers establish a communication path between each other in order to exchange information regarding proper functioning of all servers. The servers can be launched by using the runContainer script (available in the fmq/bin directory of the installation package), and supplying the %SELECTED\_HA\_PROFILE% as an argument.

In order to run shared HA profiles we need to specify dbPath as extra command line argument. This is the path where run folder will be created and this must be pointing to the same directory for both the profiles (Primary and Secondary). It would be better if this shared db is on third machine (other than machines where Primary and Secondary profiles are running). For sharing db from third machine, use same method as specified for sharing LockFile. Different methods to run HA profiles are described here. It is supposed that shared db is mounted, for Primary server, on /home/fmq1/shared for Linux machine ( C:\fmq1\share on Windows machine) and, for Secondary server, /home/fmq2/shared on Linux machine ( C:\fmq2\share on Windows machine).

#### 5.1.2.1.1 Console

##### Windows:

When launching the Primary HA Server in shared Mode, the command line would be:

```
fmq.bat -profile FioranoMQ_HA_shared/HAPrimary -dbPath C:\fmq1\share
```

For Secondary HA server in shared Mode, the command line would be:

```
fmq.bat -profile FioranoMQ_HA_shared/HASecondary -dbPath C:\fmq2\share
```

##### Unix:

When launching the Primary HA Server in shared Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_shared/HAPrimary -nobackground -dbPath /home/fmq1/share
```

For Secondary HA server in shared Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_shared/HASecondary -nobackground -dbPath /home/fmq2/share
```

#### 5.1.2.1.2 Background

##### Unix:

When launching the Primary HA Server in shared Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_shared/HAPrimary -dbPath /home/fmq1/share
```

For Secondary HA server in shared Mode, the command line would be:

```
./fmq.sh -profile FioranoMQ_HA_shared/HASecondary -dbPath /home/fmq2/share
```

### 5.1.2.1.3 Service

#### Windows

Any profile present under %FMQ\_DIR%/profiles can be made to run as NT service.

Steps for installing HA profile as NT Service:

1. Open a command prompt
2. Goto to %FMQ\_DIR%\bin\service directory
3. To install FioranoMQ Kernel as NT service, run the command  

```
install-fmq-service.bat -profile FioranoMQ_HA_shared/HAPrimary Or  
install-fmq-service.bat -profile FioranoMQ_HA_shared/HASecondary
```
4. After installation you should be able to see "FioranoMQ <%PROFILE\_NAME%>" NT service installed.
5. Start the NT Service. Logs for NT service run can be viewed in %FMQ-DIR%\bin\service\logs\%PROFILE\_NAME%.log.

#### Linux:

For enabling Shared HA profiles as service, there is a script "fmq-linux" available in fmq/bin/service directory of FioranoMQ installation directory. Follow the procedures listed here:

1. Set JAVA\_HOME at line 8.
2. Set the FIORANO\_HOME at line 10.
3. fmq-linux by default runs FioranoMQ Profile, to run Primary profile change the name of the profile on line 20 to FioranoMQ\_HA\_shared/HAPrimary. To run Secondary profile change the name of the profile on line 20 to FioranoMQ\_HA\_shared/HASecondary..
4. Give Executable Permission, "chmod +x fmq-linux".
5. To start as a service, type "./fmq-linux start".
6. To stop the server, type "./fmq-linux stop".
7. To run this as a service with system startup, type "chkconfig --add fmq-linux". This will launch the service whenever the system is started.

#### SOLARIS:

For enabling Shared HA profiles as service, there is a script "fmq-solaris" available in fmq/bin/service directory of FioranoMQ installation directory. Follow the procedures listed here:

1. Set the FIORANO\_HOME at line 9 of fmq-solaris file.
2. fmq-solaris by default runs FioranoMQ Profile, to run Primary profile change the name of the profile on line 19 to FioranoMQ\_HA\_rpl/HAPrimary. To run Secondary profile, change the name of the profile on line 19 to FioranoMQ\_HA\_rpl/HASecondary.
3. Copy the file fmq-solaris in "/etc/init.d".

4. Give Executable Permission, "chmod +x fmq-solaris".
5. To start as a service, type "./fmq-solaris start".
6. To stop the server, type "./fmq-solaris stop".
7. To run this as a service with system startup create the links from "/etc/init.d" using the following commands

To start the service at startup:

```
ln -s fmq-solaris /etc/rc3.d/S86fmq-solaris
```

**Note:** If the priority level '86' which is specified in the link already exists in the run level (rc3.d), then select a number which doesn't exist there.

**Example:** If you want to set the priority 71 then create the link using the following command in "/etc/init.d"

```
ln -s fmq-solaris /etc/rc3.d/S71fmq-solaris
```

To stop the service at shutdown/reboot:

```
ln -s fmq-solaris /etc/rc3.d/K45fmq-solaris
```

**Note:** It is recommended to choose low priority level for shutdown

8. Set JAVA\_HOME in <FIORANO\_HOME>/fiorano\_vars.sh

**Note:** Even if you set JAVA\_HOME in your .bashrc or .bash\_profile, it won't be set at startup.

So JAVA\_HOME must be set in "fiorano\_vars.sh" script file.

### 5.1.2.2 Stopping the servers

#### 5.1.2.2.1 Console

##### Windows:

When stopping the Primary HA Server in shared Mode with RMI Port 1858, the command line would be:

```
shutdown-fmq.bat -rmiPort 1858
```

For Secondary HA server in shared Mode with RMI Port 1859, the command line would be:

```
shutdown-fmq.bat -rmiPort 1859
```

##### Unix:

When stopping the Primary HA Server in shared Mode with RMI Port 1858, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1858
```

For Secondary HA server in shared Mode with RMI Port 1859, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1859
```

### 5.1.2.2.2 Background

#### Unix:

When stopping the Primary HA Server in shared Mode with RMI Port 1858, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1858
```

For Secondary HA server in shared Mode with RMI Port 1859, the command line would be:

```
./shutdown-fmq.sh -rmiPort 1859
```

### 5.1.2.2.3 Service

#### Windows:

1. To remove FioranoMQ Kernel NT service, run the command

```
uninstall-fmq-service.bat -profile %PROFILE_NAME%
```

2. Un-Install should stop and remove NT service for given profile from the services list.

#### Linux:

To remove initially created service type "chkconfig --del fmq-linux". This will remove the previously added service

#### SOLARIS:

To stop the service at shutdown/reboot type:

```
ln -s fmq-solaris /etc/rc3.d/K45fmq-solaris
```

**Note:** It is recommended to choose low priority level for shutdown.

### 5.1.2.3. Determining status of servers

It is easy to determine the status of server while running through console, as status of servers are printed on console. It can also be determined by running ServerStatus.java file available in fmq/samples/JMX directory of FioranoMQ installation directory. Details about running this file are available in readme file in the same directory.

### 5.1.2.4. Log details

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

### 5.1.2.5 Verify HA Setup

On starting the Fiorano MQ Server that is part of an HA pair, the server prints debug information about its own state (ACTIVE, PASSIVE, and ACTIVATING). It also prints information about its backup server's state whenever it detects a change.

Example Statements on console:

```
[Sat Feb 07 14:03:50 IST 2009] Old status of remote server = DEAD
[Sat Feb 07 14:03:50 IST 2009] New status of remote server = WAITING
[Sat Feb 07 14:03:50 IST 2009] Old status of remote server = WAITING
[Sat Feb 07 14:03:50 IST 2009] New status of remote server = PASSIVE
```

The Console includes statements as shown below:

'Primary Server switched to ACTIVE' or 'Secondary Server switched to PASSIVE', which indicate that the pair has successfully connected. Also, a statement gets printed when the lock is successfully acquired over the lockfile on the active servers console.

**Example:** Successfully acquired lock on: C:\lockFile.dat.

The figure below illustrates a successfully started Fiorano HA MQ Server in shared mode:

```
E:\Fiorano\FioranoMQ9.1.0\fmq\bin>fmq.bat -profile FioranoMQ_HA_shared\HAPrimary -dbPath C:\DB
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home         : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile           : E:\Fiorano\FioranoMQ9.1.0\fmq\profiles\FioranoMQ_HA_shared\HAPrimary

Server's Process ID : 5920
Created MBeanServer with ID: 16574816:125e26d5ab3:-8000:sreenivas-lap:1
[30/Dec/2009 17:50:35] license INFO The fiorano-mq9.lic license for the product MQ ver 9x is valid
alse, Mode locked = false
[Wed Dec 30 17:50:37 PST 2009] Primary Server switched to PASSIVE
Profile ..\profiles\FioranoMQ_HA_shared\HAPrimary successfully deployed on Wed Dec 30 17:50:37 PST 2009
Successfully acquired lock on :: C:\lockFile.dat
[Wed Dec 30 17:50:39 PST 2009] Trying to establish connection with localhost:3000 ...
[Wed Dec 30 17:50:40 PST 2009] Primary Server switched to ACTIVATING
[Wed Dec 30 17:50:41 PST 2009] Trying to establish connection with localhost:3000 ...
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1958
Max Memory Allocated to the Server : 254 MB.
Dashboard Listening Port: 1780
[Wed Dec 30 17:50:58 PST 2009] New status of remote server = PASSIVE
Fiorano Server accepting connections at http://0.0.0.0:1856
Server Protocol:: TCP, Default:: true
Maximum Client Connections : 1,024
FioranoMQ 9.1.0, Build # 9195
[Wed Dec 30 17:50:59 PST 2009] Primary Server switched to ACTIVE
```



## 6.1 Troubleshooting

### 1. What if HA servers are not able to run?

```
E:\Fiorano\FioranoMQ9.1.0\Fmq\bin>fmq.bat -profile FioranoMQ_HA_rpl\HAPrimary
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home         : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile           : E:\Fiorano\FioranoMQ9.1.0\Fmq\profiles\FioranoMQ_HA_rpl\HAPrimary
Server's Process ID : 4840
Created MBeanServer with ID: 39fce974:125e299410b:-8000:sreenivas-lap:1
[30/Dec/2009 18:38:32] license INFO The fiorano-mq9.lic license for the product MQ ver 9x is va
also, Node locked = false
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1958
[Wed Dec 30 18:38:43 PST 2009] Primary Server switched to WAITING
Backup Server not Connected. Check configuration or Start backup Server ...
Profile ..\profiles\FioranoMQ_HA_rpl\HAPrimary successfully deployed on Wed Dec 30 18:38:43 PST 2009
-----
HA properties are not set properly in the backup server. Please verify whether the following properties are c
1.ActiveLockReAcquisitionInterval
2.PassiveLockAcquisitionInterval
3.PingInterval
4.GatewayServerIPAddress
5.GatewayServerPort
6.LockFile
-----
Container shutdown initiated
```

This might be related to HA properties. Please verify whether the following properties are configured to use same values in the backup server.

1. ActiveLockReAcquisitionInterval
2. PassiveLockAcquisitionInterval
3. PingInterval
4. GatewayServerIPAddress
5. GatewayServerPort
6. LockFile

### 2. While running HA Profile following error message is thrown to console:

```
E:\Fiorano\FioranoMQ9.1.0\Fmq\bin>fmq.bat -profile FioranoMQ_HA_rpl\HAPrimary
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home         : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile           : E:\Fiorano\FioranoMQ9.1.0\Fmq\profiles\FioranoMQ_HA_rpl\HAPrimary
Server's Process ID : 1148
Created MBeanServer with ID: -58b5613d:125e2937c3c:-8000:sreenivas-lap:1
[30/Dec/2009 18:32:14] license INFO The fiorano-mq9.lic license for the product MQ ver 9x is valid
also, Node locked = false
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1958
Lock file::null is not specified properly.
=====
Unable to validate permissions on lock file::null.
Either file is not valid or lock file machine is not allowing to verify permissions on lock,
Unable to boot the server
=====
Container shutdown initiated
Container shutdown successful at Wed Dec 30 18:32:25 PST 2009
```

Lock file::null is not specified properly.

=====

Unable to validate permissions on lock file::null.

Either file is not valid or lock file machine is not allowing to verify permissions on lock,

Unable to boot the server

=====  
 =====

This will occurs when Lock file is not specified. So, please make sure to specify Lock file.

**3. What if HA servers are not able to run, while launched as a service?**

Look in the logs runContainer\_output\_log and runContainer\_error\_log. These logs have all the error and messages thrown at run time.

**4. SocketBindException saying that the HA Port is already bound.**

This exception means that some other program running on the HA port or the last instance of the server is not properly killed. You can choose to stop or kill the application which is holding up the port and start the server again or choose a different HA port. But changing this means that there needs to be a change in the Backup Servers' configuration for its Backup Server port.

**5. None of the servers connect to each other.**

One Server is in STANDALONE state and the other server in WAITING state and both the servers are unable to communicate.

This behavior means that the Backup Server IP and port numbers are wrong for both the server configurations.

**Example:** Server consoles when it cannot connect to each other.

The Server console below illustrates a situation where the server is not able to connect to the Backup Server. If it the backup server is running as shown, then there is a problem with the configuration. The message prints the IP address and the port to which it is trying to connect to establish the HA channel. You can check if the Backup Server is running in the printed IP address and port.

**Primary Server Console**

```
E:\Fiorano\FioranoMQ9.1.0\fmq\bin>fmq.bat -profile FioranoMQ_HA_rpl\HAPrimary
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home         : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile           : E:\Fiorano\FioranoMQ9.1.0\fmq\profiles\FioranoMQ_HA_rpl\HAPrimary

Server's Process ID : 1724
Created MBeanServer with ID: 7cb7c1ee:125e28660e3-8000:sreenivas-lap:1
[30/Dec/2009 18:17:55] license INFO The fiorano-mq9.lic license for the prod
also, Node locked = false
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1958
[Wed Dec 30 18:18:13 PST 2009] Primary Server switched to WAITING
Backup Server not Connected. Check configuration or Start backup Server ...
Profile ..\profiles\FioranoMQ_HA_rpl\HAPrimary successfully deployed on Wed Dec 30 18:18:1
[Wed Dec 30 18:18:30 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
Successfully acquired lock on ::C:\lockFile.dat
Max Memory Allocated to the Server : 254 MB.
Dashboard Listening Port: 1780
Fiorano Server accepting connections at http://0.0.0.0:1856
Server Protocol:: TCP, Default:: true
Maximum Client Connections : 1,024
FioranoMQ 9.1.0. Build # 9195
[Wed Dec 30 18:18:40 PST 2009] Primary Server switched to STANDALONE
[Wed Dec 30 18:18:58 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:19:17 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:19:35 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:19:54 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:20:12 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:20:31 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:20:49 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
[Wed Dec 30 18:21:08 PST 2009] Trying to establish connection with 192.168.1.177:3000 ...
```

## Secondary Server Console

```

E:\Fiorano\FioranoMQ9.1.0\fmq\bin>fmq.bat -profile FioranoMQ_HA_rpl\HASsecondary
Fiorano Home      : E:\Fiorano\FioranoMQ9.1.0
Java Home         : E:\Fiorano\FioranoMQ9.1.0\jre1.5.0_16
Profile           : E:\Fiorano\FioranoMQ9.1.0\fmq\profiles\FioranoMQ_HA_rpl\HASsecondary
Server's Process ID : 5688
Created MBeanServer with ID: -4901c7c5:125e286adfa:-8000:sreenivas-lap:1
[30/Dec/2009 18:18:15] license INFO The fiorano-mq9.lic license for the product
left = 47, Locale specific = false, CPU based = false, Node locked = false
RMIConnectorServer started at: service:jmx:rmi://localhost/jndi/fmq
RmiConnectorServer Listening Port: 1859
[Wed Dec 30 18:18:32 PST 2009] Secondary Server switched to WAITING
Backup Server not Connected. Check configuration or Start backup Server ...
Profile ..\profiles\FioranoMQ_HA_rpl\HASsecondary successfully deployed on Wed Dec 30 18:18:32 PST 2009
[Wed Dec 30 18:18:49 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:19:08 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:19:27 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:19:46 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:20:04 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:20:24 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:20:42 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:21:01 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:21:20 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:21:39 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:21:58 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:22:17 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...
[Wed Dec 30 18:22:36 PST 2009] Trying to establish connection with 192.168.1.24:2000 ...

```

## 6. Both servers go to Standalone/Active state

Both servers go to Standalone/Active state in replicated/shared mode respectively if the network link between them is broken. This can happen if the servers do not refer to the same **LockFile**.

## 7. If Both servers print the message 'Both servers are in STANDALONE state. Please refer troubleshooting section of Fiorano HA guide or take backup of both server databases and contact fiorano support team.'

If this happened after proper configuration of servers, please take backup of both server databases, server logs and console logs. Please call Fiorano support team to inform the situation and take help from them to follow rest of the steps. The following steps can cause message ordering issues.

To avoid downtime in production servers, user can move secondary server database and delete log file. Now run both the servers. Servers should work now properly and process all the data. If primary server switched to ACTIVE, then user need to investigate secondary server data for avoiding message loss. If primary server switched to PASSIVE then user need to investigate both primary and secondary server data for avoiding message loss.

Now, server database (contents of run directory) needs to be investigated to see whether it contains any latest messages which may not be present in other server database. Now move this database to a StandAlone FioranoMQ server run directory and run FioranoMQ server. Using fmq/samples/PTP/QueueBrowser and fmq/samples/PubSub/MessageBrowsers all pending messages can be seen. Either the messages which are not present in the other server data need to be transferred to the HA server using a custom application or a StandAloneBridge and StandAloneRepeater can be used to transfer all messages to the HA server.

## 7.1 Best practices.

- For database store use SAN disks. This is not necessary for running HA profiles, but they are more reliable and data can be retrieved in case of failure.
- Make sure to have Lockfile and Gateway server on same machine.
- Use nfs-4 for mounting files.
- In case of shared HA profiles give Separate addresses for storing server logs. This will make debugging easy.

## 8.1 FAQ

### **What is high availability?**

High availability is a system design protocol and associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period.

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 down within seconds. Hence, it is absolutely imperative for the messaging backbone 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 backup should be automatic and transparent to the client application.