Fiorano
Peer-to-Peer Dataflow Pipelines™

www.fiorano.com

FioranoMQ 9

Reference 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

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

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

Fiorano

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 is 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 agreement of the parties.

(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 is 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 perpetually 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 THEPRODUCT, 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 is subject to interest of the lower of the legal interest rate or one percent (1%) per month, which interest is immediately due and payable.

(b) Payments. All payments made by Customer shall be in United States Dollars for purchases made in all countries except the United Kingdom, in which case the payments shall be made in British Pounds Sterling. 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 codeof 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 Union of India shall apply and any disputes arising hereunder shall be subject to the jurisdiction of the Hon'ble City Civil Court, Bangalore, India. 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 is 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 is 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
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 2007 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 2007 Tools or Fiorano Mapper Tool 2007, 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.

Contents

Chapter 10: HA Shared Configuration 132

Chapter 15: Advanced Configuration 166

Chapter 1: Connection Factory Configuration

AddShutdownHook

Boolean used to enable/disable shut down hook in the client.

Enabling this parameter ensures that the program exits normally, when the last non-daemon thread exits or when the exit method is invoked. This also makes sure that connections and sockets are closed properly when the virtual machine is terminated in response to a user interrupt, such as typing ^C, or a system-wide event, such as user log off or system shutdown.

Valid values:

- **yes** - enables shutdown hook
- **no** - disables shutdown hook (Default value)
- **none** - takes default value set in the Server

Default Value:

No - Disables shutdown hook

Shut down hook is disabled by default in the client applications.

AllowDurableConnections

Boolean used to disable or enable durable connections for a particular client when durable connections are enabled on the server. When **AllowDurableConnections** is enabled, messages get stored into CSP (Client side persistence) storage when connection with the server is down.

This parameter is used to set the durable connections to true/false in a client.

This value set in the Connection Factory is overridden by the value set in the client application. This parameter can also be set in the client application as shown below:

```
env.put(FioranoJNDIContext.ALLOW_DURABLE_CONNECTIONS, "true");
```

Here **env** is a hashtable used to create the InitialContext which stores all the JNDI parameters.

Valid values:

- **yes** - enables durable connections in the client. (Default value)
- **no** - disables durable connections in the client.

- **none** - takes default value set in the Server.

Default Value:

yes - enables durable connections in the client.

Dependencies:

This parameter is used for client side configuration.

If durable connections are disabled in Server and enabled in client, effectively it is disabled overall. And if durable connections are enabled in Server, its effect is dependent on the client-side configuration.

AutoDispatchConnections

Boolean used to specify whether dispatcher server should be used for creating client connections or not. By default this flag is enabled.

This parameter is used in the case of a dispatcher server. if it is set to true, the incoming client connections are automatically redirected to the least loaded Server that is registered to the dispatcher. if it is set to false connections are created on the dispatcher itself

Valid values:

- **yes -** enables automatic dispatching of the connection
- **no** - disables automatic dispatching of the connection

Dependencies:

This parameter affects only in the case of the Dispatcher Server.

Backup ConnectURLs

URLs of the backup servers used in case primary server is down. Multiple backup URLs may be specified as a semicolon-separated string of URLs.

Example: Backup Connect URLs can be specified as follows

http://backupServer1:1856;http://backupServer2:1856

BatchTimeoutInterval

Time interval after which batch of non-persistent messages will be sent irrespective of the size of buffer.

Non-persistent messages are sent in batches to the Server, these batches are sent to the server when the batch size exceeds a particular size [default 32 bytes] or for the time specified by batchTimeoutInterval.

Valid values:

Time is specified in milliseconds. Default is 1000 msecs.

BatchingEnabled

Enables message batching algorithm in the client runtimes.

If batching is enabled, non-persistent messages are sent in groups to the server which results in higher message rates. Batching has no effect on persistent messages.

Valid values:

- **true**: enables batching (Default value)
- **false**: disables batching

Default value: true

ClientProxyURL

This parameter holds the configuration information of the client proxy provider in case of http tunneling.

Valid value:

String - value that contains information of the proxy provider.

For Example, http://ClientProxyServer:80

CompressionManager

CompressionManager implementation is used for compressing messages. The Compression Manager specified should be an implementation of the **fiorano.jms.services.msg.compression.ICompressionManager** interface provided by FioranoMQ.

Valid value:

class path of the implemented Compression Manager. Default value is fiorano.jms.services.msg.compression.def.CompressionManagerImpl

Connection ClientID

clientID used for all the connections that are created using this connection factory.

Valid value:

String value

Dependencies:

This parameter depends on **EnsureUniquenessOfClientID** of FioranoMQ->Fiorano->etc->FMQConfigLoader. If EnsureUniquenessOfClientID is set to **true**, no two connections can have same clientID. Hence, setting ConnectionClientID can throw exceptions when more than one connection is created with the Connection Factory.

CreateLocalSocket

This parameter is used when both the client and server are on same machine.

If it is set to true, then a socket is created, when client connects to the Server. If it is set to false, no socket is created for local connections.

Valid values:

- **yes** - creates local socket.
- **no** - does not create a local socket (default value)

By default it is set to **no**

Disable Ping

Boolean used to disable ping for a particular client when pinging is enabled on the server.

If ping is enabled, Client tries to ping the Server for every **PING_INTERVAL** seconds (default 60 seconds). If reply not received from the Server, client re-validates the connection.

Valid values:

- **yes** - pinging is disabled on the client-side
- **no** - pinging is enabled on the client-side (default value)
- **none** - use the default value that is, pinging is enabled by default

DisableSendingCSPStoredMessages

Boolean used to disable sending pending messages stored in CSP to the server.

When a connection with particular clientID is interrupted and if Durable Connections are enabled, messages are stored in local cache under the clientID of the connection. If this flag is enabled, these messages are not sent in the setClientID () call.

Valid values:

- **true - yes** - d disables sending messages stored in CSP when client ID is set.
- **False - no** - enables sending CSP stored messages.

Note: By default DisableSendingCSPStoredMessages is false or no. In the Studio possible values are yes/no. While setting through samples the values are true/false.

JNDI name

`FioranoJNDIContext.DONT_SEND_PREVIOUSLY_STORED_MESSAGE`

For example, env.put(FioranoJNDIContext.DONT_SEND_PREVIOUSLY_STORED_MESSAGES, "true");

Usage

This parameter is set when the messages stored in CSP are NOT to be sent after re-establishing connection with the server.

Scenario

The server is started with durable connection enabled. A publisher and a subscriber are started. Publisher has the DONT_SEND_PREVIOUSLY_STORED_MESSAGES flag set to true. The server is shutdown. Publisher publishes a few messages and is closed too. The server and publisher are restarted. The subscriber doesn't receive the messages sent by the publisher while the server was down.

DurableConnectionReconnectInterval

Time interval between two reconnect attempts.

When a client is disconnected from the Server and if auto revalidation is enabled, the client repeatedly tries to reconnect to the Server in a loop. The time between the successive tries is set by this flag.

Valid value:

The value is set in Milliseconds. By default the interval is set to 1000 msecs.

This parameter is used only when auto revalidation is enabled.

DurableConnectionsBaseDir

When durable connections are enabled and the Server goes down, on the client-side messages that are being published while revalidating the connection, are stored on local disk.

This parameter specifies the name of the base directory to which messages should be persisted.

Valid value:

Any string value that is legal to be a directory name.

The default value is set to **cspCache**. Messages are persisted in the directory cspCache if this variable is not set.

EnableAutoRevalidation

Boolean used to disable auto-revalidation for a particular client when auto-revalidation is enabled on the server.

By enabling auto revalidation, client attempts to reconnect to the server when the Server is down.

Valid value:

- **yes** - enables auto revalidation for the client
- **no** - disables auto revalidation for the client

Default value is enabled.

The client revalidates only when the Auto revalidation is enabled on the Server, that is, client does not revalidate if Auto Revalidation disabled in Server and enabled in Connection Factory parameters.

Server side value takes precedence over CF parameter or client-side set parameter

UseFioranoCbr

Enables/Disables Content Based Routing. CBR allows for XPath based selection on messages with XML content. For more information, refer to chapter 18 (FioranoMQ Content Based Routing) of FioranoMQ concepts Guide.

Valid value:

- **yes** - enables Fiorano Content Based Routing
- **no** - disables Fiorano Content Based Routing

EnableLMS

Boolean used to enable/disable Large Message Support.

Valid values:

- **yes** - enables LMS on client-side (Default)
- **no** - disables LMS on client-side

Dependencies:

UseFioranoCbr flag should be set to **no** for LMS to be working. LMS does not work if AllowDurableConnections is set to TRUE. LMS does not work in case of HA

HTTPProxyURL

This parameter holds the configuration information of the HTTP proxy provider.

Valid value:

A String value containing information about proxy provider.

For example, http://ProxyServer:80

Use Local Procedure Calls

Boolean used to enable/disable Local Procedure Calls. If this is enabled then clients, within the server process, using this Connection Factory connect to the server directly without making any socket connections.

Valid values:

- **yes** - enables LPC
- **no** - disables LPC. By default LPC is disabled.

by default LPC is disabled.

LookupPreferredServer

Boolean used to specify that the client should connect only to the preferred server. When this parameter is enabled, Clients connecting to the Dispatcher redirected to the Server set as preferred Server. If it is disabled clients connect to the least loaded Server.

Valid values:

- **yes** - enables to lookup preferred Server
- **no** - disables looking up preferred Server

Dependencies:

This parameter is used only in the case of a Dispatcher Server.

MaxAdminConnectionReconnectAttempts

Maximum number of reconnect attempts that will be made by the Admin connection to connect to the server when server is down.

Valid value:

Any positive integer value. By default this parameter is set to -1 (infinite).

Note:

- If a positive value is set, admin connection tries to reconnect for that number of times.
- If it is not connected within these attempts revalidation is stopped and an exception is thrown.

MaxDurableConnectionReconnectAttempts

Maximum number of reconnect attempts that the client will make if it is unable to connect to server when server is down.

Valid value:

Any positive integer value. By default this parameter is set to -1 (infinite).

Note:

- This flag will be used only if durable connections are enabled on the server.
- If a positive value is set, the connection tries to reconnect for that number of times.
- If it is not connected within these attempts, revalidation is stopped and an exception is thrown.

MaxSocketCreationTries

Maximum number of attempts that the client will make to create a socket connection on the server when it is unable to connect to the server. If it fails to connect in the specified number of attempts exception is thrown on Client-Side.

Valid value:

Any positive integer value. Default value is 1.

ProxyAuthenticationRealm

It holds the address of the Authentication Realm being used on the Proxy through which HTTP based Connections created using this ConnectionFactory will be routed.

Valid value:

String representing the address of the proxy server behind which the Authentication Realm exists.

Dependencies:

This property is used only when the Transport Protocol is set as HTTP.

ProxyCredentials

Password of the proxy principal used in case proxy is used.

ProxyPrincipal

Name of the proxy principal used in case proxy is used.

ProxyType

Sets the type (name) of the proxy being used for the HTTP based connections created using this ConnectionFactory. This property is used only when the Transport Protocol is set as HTTP.

Valid values:

- MS_ISA_PROXY
- WIN_PROXY
- NETSCAPE_PROXY
- WINGATE_PROXY
- DEFAULT_PROXY

PublishBehaviourInAutoRevalidation

This parameter defines the behavior of publisher when auto re-validation is enabled and durable connections are disabled in the server. A new publish call can either throw an exception or block or ignore the message. Default value is - throw exception.

Valid values:

- **0 – Exception** - Throws an exception on publishing messages (When the server is down, auto re-validation is enabled, but durable connections are disabled).
- **1 - Ignore** - I Ignores further publish message calls and all the messages sent when the server is not up will be lost. (Same conditions as above).
- **2 – Block** - Blocks further messages from being published till the auto re-validation is done. (Same conditions as above).

JNDI name:

`FioranoJNDIContext.PUBLISH_BEHAVIOUR_IN_AUTO_REVALIDATION`

For example, env.put(FioranoJNDIContext.PUBLISH_BEHAVIOUR_IN_AUTO_REVALIDATION, "2");

Usage:

This parameter is used to prevent exceptions from being thrown when messages are sent during the auto re-validation with durable connections disabled. The user may either choose to ignore the messages sent till the auto re-validation is done or could prevent further sending of messages till auto re-validation is complete.

Example:

If FioranoMQ server is used for live trading then old messages cannot be used. So if server is down then publisher should not send any messages as they will be sent afterwards. So this flag should be set to 2 which will block the publisher from sending messages when connection with the server is down.

Dependencies:

If these parameters are also present in the client then EnableAutoRevalidation must be set to true and AllowDurableConnections must be set to false.

PublishWaitDuringCSPSyncup

When Server is disconnected, messages are stored in CSP cache on local machine. When the Server is reconnected, messages stored in CSP are sent to the Server while the current messages are stored in CSP with some wait time. The publish call waits for **CSPSyncup** time specified by this parameter until the CSP messages are sent to the Server. After all CSP messages are sent, publisher sends normally without any delay time.

Valid value:

Time is specified in Milliseconds. Default value is 1000 msecs.

SecurityManager

The Security Manager implementation used to create secure connections [HTTPS or SSL] with the MQ server. This manager should be an implementation of the f**iorano.jms.runtime.IFMQSecurityManager** interface provided by FioranoMQ.

Valid value:

Class path to the security Manager class.

ServerProxyURL

This parameters holds URL for the Server side proxy through which TCP based Connections will be routed using HTTP Tunneling.

Valid value:

String value that contains info. of the proxy provider.

For example, http://ClientProxyServer:80

Dependencies:

This URL will be used only when the Transport Protocol is set as TCP

SleepBetweenSocketCreationTries

Time interval between two socket creation attempts. While creating a connection to the server, client RTL creates sockets on Server. If the server is unavailable at the time, RTL tries to create socket for number of times specified in the parameter - **MaxSocketCreationTries** between successive tries the RTL waits for the time mentioned by this parameter.

Valid value:

Time is specified in milliseconds. The default sleep time is 200 msecs.

SocketKeepAliveEnabled

Once the property is set to true, packets are sent to the remote system when no data is being exchanged to keep the connection active. This is handled by the TCP layer itself.

By default the value is turned off.

SocketTimeout

Time interval after which the socket call will timeout. The flow of execution stops, if it is waiting for reply from the socket. If the connection to the Server is lost, the applications hang on the socket for maximum time set by this parameter.

Valid value:

Time in Milliseconds. The default timeout value is set to four minutes.

SocksProxyURL

Sets the URL for the SOCKS Proxy through which HTTP based Connections will be routed.

Dependencies:

This URL will be used only when the Transport Protocol is set as HTTP.

TCPBatchSize

Non-persistent messages are sent to the Server in batches. This parameter determines the size of the batch. If the message batch size exceeds this size, batch is sent to the Server or else messages are stored in a buffer.

Valid value:

The size is specified in bytes. The default value is 32KB

Dependencies:

Batching should be enabled for this parameter is to be used. If BatchTimeoutInterval completes before the message batch size, then messages are sent to the server irrespective of the Batch size.

TransportProtocol

Protocol used for communicating with server.

Valid value:

Transport protocol can be set to either **TCP** or **HTTP**.

java.naming.security.protocol

Name of the security protocol used to create secure connections with the MQ server.

Valid value:

The possible values that this variable can take are **PHAOS_SSL and SUN_SSL**.

UpdateConnectURL

Boolean used to specify whether connect URL of the connection factory should be updated or not whenever the server restarts.

Valid values:

- **yes** - updates the connectURL
- **no** - does not update the connectURL

By default this is set to false.

UseSingleSocketForSendReceive

By default, FioranoMQ uses two sockets per connection. One socket is used for sending requests and another for receiving messages. This property allows the user to use the same socket for both send and receive operations.

Valid values:

- **yes** - uses single socket for send receive
- **no** - uses two socket for send and receive

By default this is set to true.

XASocketTimeout

Time interval after which socket call will timeout in case of XA enabled Server.

Valid value:

Time in milliseconds. The default timeout value is set to ten minutes.

isThreadContextClassLoaderUsed

Sets the value of USE_THREAD_CONTEXT_CLASS_LOADER.

Valid values: yes/no/none

Chapter 2: Common FioranoMQ Configuration

FMQConfigLoader

The following section explains the parameters present in **FioranoMQ->Fiorano->etc->FMQConfigLoader**

LazyRSCreation

This value determines if thread creation is optimized on the Client side. Although the connection is started with the start method call, a thread for listening in on the data on the connection has to be started. When thread creation is optimized, a thread is created only when the first consumer is created on the session created by the connection.

Valid values:

- **yes** - Thread creation is optimized on the client side that is a thread is created only when the first consumer is created on the session created by the connection.
- **no** - Not optimized. Thread is created as soon as the connection is started. (Default)

Example

In a situation where you want to economize the number of threads running in an application, this must be set to **yes** so that there are no unnecessary threads created until a consumer is created

AllowDurableConnections

A Durable Connection maintains connectivity with FioranoMQ at all times. The applications do not have to take care of storing, re-connecting and then forwarding the stored messages to the server. These activities are not visible to the client application and are automatically performed by FioranoMQ's runtime library. When the connection is restored, messages stored in the local store are automatically sent to the server.

Valid values:

Default value is no.

- **yes** - Durable connections are enabled in this server. All that a client application has to do to make its connection durable is to AllowDurableConnections in its application code (Refer FioranoMQ Handbook section 5.1.1).
- **no** - Durable connections are not enabled in this server. This nullifies the AllowDurationConnections property set in the client.

Example:

Consider a process computer monitoring a steel mill. Real time steel production information is sent each second to a main hub. The main hub uses this information to generate the desired results. If the connection between the Process machine and the Hub breaks, the send mechanism fails and an exception would be raised. Since this data is generated only once, the application is required to store this data at its end upon encountering the exception and then spend resources to connect back to the server. This adds considerable load to the application. In such cases, a Durable Connection comes to the rescue as it does all the hard work for the application. It automatically tries to re-establish the connections, stores the data in transit and sends it to the server as soon as the connection is restored.

Dependencies:

AutoRevalidationEnabled (No. 13) - This value depends on this. If AllowDurableConnections is set to yes, then AutoRevalidationEnabled is automatically set to yes

UseSingleSocket

Each JMS Connection internally results in creation of two sockets with the server. By turning on this flag each Fiorano client is instructed to use a single socket instead.

Valid values:

Default value is no

- **yes** - Each JMSConnection that a client makes results in only socket.
- **no** - Each JMSConnection that a client makes results in two sockets.

Example

In a situation where you want to economize the number of sockets that you create, set this value to **yes**.

UseSingleSocketForAdmin

Turning on this flag, will result in starting off a thread for admin connection that would constantly be waiting for data on socket. This will detect loss of connection immediately.

Valid values:

Default value is **yes**

- **yes** - A thread for an admin connection is created on the socket and it is constantly waiting for data on socket. This will detect loss of connection immediately.
- **no** - Loss of connection cannot be detected immediately.

Example:

In most situations it is better to set UseSingleSocketForAdmin to yes because it provides an easy way to detect loss of connection immediately.

UseFioranoCbr

In group based systems messages are classified as belonging to a certain group, referred to as Topic. Publishers are required to label each message with a topic name, while consumers subscribe to all messages on a particular topic.

When set to **yes** this offers a better alternative to group-based systems, known as content-based routing. These systems route messages to subscribers, based on content instead of message properties contained in the headers. There is no overhead imposed on the publisher and prior knowledge of the domain is not required. Subscribers have the added flexibility of choosing filtering criterion along multiple dimensions, without requiring pre-definition of groups.

Valid values:

Default value is **no**

- **yes** - Enables content based routing at the server side. Now the server has CBR enabled, and clients can send and receive XML messages with XPath selectors.FioranoMQ CBR utilizes a subset of XPath notation and SQL92 syntax to specify XPath message selectors. Essentially, you must use only absolute paths. You can combine several XPath string with AND/OR conditions. Refer samples in samples/PubSub/ContentBasedRouting for running clients with CBR.
- **no** - Content based routing is not enabled in the server.

Example:

Stock Trade Example

Consider a brokerage firm that may have thousands of subscribers interested in information on stock trades. Each subscriber has its own selection criterion based on its requirement. One subscriber would like to be alerted when two stocks fall below a certain price.

MSFT stock falls below 55 AND ORCL stock falls below 15

CreateDefaultACL

This indicates whether or not to create default ACLs for different MQ objects. When set to **yes** different MQ objects are attached with them their respective ACLs that decides which set of users/groups have access to that MQ object.

Valid values:

Default value is **yes**

- **yes** - MQ objects have their respective default ACL. Even if this option is set to yes, for the default ACLs to have any effect, AclBasedDestinationSecurity must be set to true.
- **no** - No Default ACLs are created for each MQ object.

Dependencies:

See also AclBasedDestinationSecurity (No. 7).

AclBasedDestinationSecurity

This indicates whether ACL based security is enabled or not.

Valid values:

Default value is **no**

- **yes** - If Enabled, user operations will first be checked against the associated ACL.
- **no** - User operations will not be checked against the ACL.

Example

In a multi user environment in which each user must have different access privileges with respect to the MQ Server, this value must be set to **yes**.

AllPermissions

This indicates whether all default permissions for a new ACL should be negative or positive. For this option to have any effect AclBasedDestinationSecurity (No. 7) must be set to true.

Valid values:

Default value is **yes**

- **yes** - When set to yes, all default permissions for a new ACL would be positive.
- **no** - When set to no, all default permissions for a new ACL would be negative.

Example

This parameter can be set to **no** when you want to add users conservatively (that is with negative permissions) and then add their respective permissions later.

AllowOnTheFlyAclCheck

This flag indicates whether modifications in an ACL, while the server is still running, would be reflected in currently connected clients or not. Although AclBasedDestinationSecurity would be set to **yes** this will not affect the currently connected clients unless AllowOnTheFlyAclCheck is set to **yes**.

Valid values:

Default value is **no**

- **yes** - Currently connected clients behave in the same way as future clients that will connect after the modifications in the ACL.
- **no** - Currently connected clients behave in the way they behaved before the modifications in the ACL.

Dependencies

The AllowOnFlyAclCheck flag works fine for all permissions except in the following scenario,

1. A publisher is publishing non-persistent messages on a topic.
2. The permission to create publisher should now be revoked on this topic.
3. No exception will be thrown even though user is not allowed to publish because messages are sent in batch mode. For getting an exception immediately,

Work Around 1:

For NP messages, batching is by default enabled which leads to the above explained behavior. To get the Exception at the send API, set the BatchingEnabled parameter in the ConnectionFactory as FALSE.

Work Around 2:

Add this following line in the client code environment while performing the lookup,

```
env.put("BatchingEnabled", "false")
```

Where **env** is the environment passed while performing a JNDI lookup. This will disable batching for this particular client.

See also AclBasedDestinationSecurity (No. 7).

AllowOnTheFlyCreationOfDestinations

This flag controls the behavior of createTopic and createQueue apis in a JMS Session. It only resist the normal user from creating the destination using the session.createTopic or session.createQueue apis. It does not prevent the admin user to create the destinations. You can still create the destinations using admin connection and admin services.

Valid values:

Default value is **yes**

- **yes** -A new destination would be automatically created when invoked with a destination name that does not exist.
- **no** - When invoked with a destination name that does not exist an exception would be thrown.

Example:

In a situation where clients should not be allowed to create any destinations of their own, the MQ Server administrator can make sure only the required destinations are created and clients do not add to the already existing set, by setting this value to **no**

EnsureUniquenessOfClientId

Every FioranoMQ JMS client is supplied with a clientID. This flag indicates whether uniqueness of clientID is required or not.

Valid values:

Default value is **yes**

- **yes** - A connection that sets a client Id that is already set on another connection will get an exception on doing so.
- **no** - A connection that sets a client Id that is already set on another connection will proceed normally.

Example:

In a situation where each client's clientID carries some significance that requires them to be unique this value must be set to **yes**.

MaxTransactionBufferSize

Maximum amount of data (in bytes) that a transacted Session would store in uncommitted state. Attempt to publish more data in uncommitted transaction will result in an exception.

Valid values:

Default value is **10240000**

Range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

EnableAutoRevalidation

This controls if auto-revalidation is enabled by default. Auto-revalidation refers to Fiorano's runtime being able to detect loss of connectivity and its attempts to automatically re-connect back to the server.

Valid values:

Default value is **no**

- **yes** - Fiorano's runtime upon detecting loss of connectivity will automatically try to re-connect back to the server.
- **no** - Client will not automatically reconnect if connection is lost.

Dependencies

This depends on AllowDurableConnections. If that is set to yes then EnableAutoRevalidation is also set to **yes**.

HttpPollingInterval

The polling interval for HTTP requests, the amount of time (in milliseconds) the request would wait for messages before returning response from the server.

Valid values:

Default value is **10000**

Range of integer values in java. (-2^31 to 2^31 -1)

All values less than or equal to 0 are as good as each other.

IncludeMessageID

Specifies whether or not to include messageID as part of the message.

Valid values:

Default value is **no**

- **yes** - messageID will be included as part of the message.
- **no** - messageID will not be included as part of the message.

Resumetimeoutinterval

Integer specifying the resume timeout to be used by an LMS (Large Message Support) application. If resume does not happens in this timeout it comes out from resuming and starts normal transfer. For more details about LMS, refer to Chapter 8 in FioranoMQ Handbook.

Valid values:

Default value is **-1**, which means that the application will wait indefinitely.

Range of long values in java. (-2^63 to 2^63 -1)

All values less than zero are as good as each other.

MessageSelectorFactoryClassName

Class Name that provides CBR Engine functionality for FioranoMQ Server. This property would be used only if Content Based Routing (CBR) is enabled in FioranoMQ Server.

Valid values:

Default value is **fiorano.jms.cbr.cbr1.def.MessageSelectorFactory**

SSLEnabled

Property which enables SSL connection with the server.

Valid values:

Default value is **no**

- **yes** - SSL connections are enabled with the server.
- **no** - SSL connections are not enabled with the server.

PingEnabled

This property controls if client connections are automatically pinged. Pinging is essential for detecting Network problems.

Valid values:

Default value is **yes**

- **yes** - Client connections are automatically pinged.
- **no** - Client connections are not pinged.

SystemEncodingFormat

Encoding Format for UTF

Valid values:

Any string. Default value is **UTF-8**

Chapter 3: Queue Subsystem Level Configurations

Note: From FioranoMQ 9.1.0 release onwards, a unique Destination Level Configuration support is included. For more on this, please refer to section **29.1 Support for Destination Level Configuration** of *FioranoMQ Handbook*.

UnAckedQueueBufferSize

Buffer size of receiver queue for unAcked messages. In case of the size of the message queue for unAcked messages has exceeded this limit, then the client will get an appropriate message to commit the received messages.

Values:

The default value set for this parameter is **10485760** in bytes. And any positive integral value can be assigned to this parameter.

MaxNumberOfQueues

Maximum number of queues that can be created. It also includes Temporary Queues. The JMS provider will allow to create up to these number of queues and any attempt to create beyond this number will be unsuccessful and a proper message will be propagated in case this limit is reached.

Values:

The default value set for this parameter is **-1** which means infinite and the upper limit for the number of queues that can be created is unlimited. And non-negative integral number & -1 can be assigned to this parameter.

RedeliveryTriesOnListenerException

Parameter specifying number of redelivery tries when RuntimeException is thrown in Message Listener. This is used in AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE mode only. The redelivered message is discarded once this value is reached.

Values:

The default value set for this parameter is **2**. And any positive **integral** value can be assigned to this parameter. If this parameter is given a negative value, then the message will not be delivered to the consumer.

Example:

In case of any RuntimeException is thrown while processing the message received by the MessageListener, it is ensured that the message gets redelivered before the number of attempts reaches this maximum limit, which makes it compliant with JMS specifications.

Dependencies:

Only applicable when the client session is either in AUTO_ACK or DUPS_OK_ACKNOWLEDGE modes.

DelayInMsgDeliveryOnListenerException

Parameter specifying the delay in message redelivery when RuntimeException is thrown in Message Listener. This is used in AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE mode only. The redelivered message is discarded once max redelivery tries value is reached.

Values:

The default value set for this parameter is **-1**, which means that there is no delay between redelivery tries for a message, when RuntimeException occurs in MessageListener. And any positive value which in **Long** range can be assigned to this parameter. And there will not be any delay between attempts, if any negative value is assigned to this parameter.

Dependencies:

Only applicable when the client session is either in AUTO_ACK or DUPS_OK_ACKNOWLEDGE modes.

CleanupDmqAtStartup

Enabling the cleanup of DeadMessageQueue at server startup. This flag is used only at the server startup. Before starting the server, if this flag is set to **true**, all the messages from the SYSTEM_DEADMESSAGES_QUEUE will be purged.

Values:

The default value for this parameter is **false**. The valid values are true/false.

EnableSnooperOnAllQueues

Specifies whether snooper service is enabled on all queues. When snooping is enabled on a queue, all the messages that are sent to this queue are copied to a topic by name **SYSTEM_MESSAGESNOOPER_QUEUE**, so that, any subscribers registered to this topic can snoop the messages sent to that particular queue.

Values:

The default value set for this parameter is **false**. The valid values are true/false. When set to **true**, a SnooperListener is added to each one of the queues and all the incoming messages are snooped and added to the topic **SYSTEM_MESSAGESNOOPER_QUEUE**.

Example:

This parameter can be used when it is needed to monitor the kind of messages are being sent to the queues by the clients. In order to do this, just enable this parameter and subscribe to the topic **SYSTEM_MESSAGESNOOPER_QUEUE**.

EnableDMQOnAllQueues

Specifies whether DMQ service is enabled on all queues. When DMQ is enabled on a queue, all the messages that are dead and are required to be stored are stored in the **SYSTEM_DEADMESSAGES_QUEUE** by default.

Values:

The default value set for this parameter is **false**. And the valid values are **true / false**.

Example:

This parameter can be used when it is needed to monitor the messages that are expired.

Dependencies:

The expired messages are stored only when **StoreWhenDead** flag which is set for FioranoMessage is set to **true**. The getter and setter methods for this flag are **getStoreWhenDead** and **setStoreWhenDead**, respectively.

PrefetchCount

Specifies the prefetch count for single receive call. Prefetch count is the number of messages fetched in single receive call. It denotes the maximum number of messages that are fetched from the server in one single received call. The number of messages fetched also depends on the parameter **MaxPrefetchSize**. Suppose, the total size of the batch of messages exceeds the MaxPrefetchSize, then the number of messages that are actually fetched can be less than the **PrefetchCount** set.

Values:

The default value set for this parameter is **three**. And the valid values are any positive integral value such that it is greater than or equal to PrefetchThreshold value set.

Dependencies:

If the message count to be delivered to the consumers approaches the PrefetchThreshold, then only the client will send a request to fetch the number of messages which are set by PrefetchCount value, from the Server.

PrefetchThreshold

PrefetchThreshold count used for single receive call. When the number of messages in the message queue of the client approaches this limit, then the client will send a request to the server for fetching more messages. The number of these fetched messages depends upon **PrefetchCount** value set.

Values:

The default set for this parameter is **1**. And the valid values are any positive integral value such that it is less than PrefetchCount value set.

Dependencies:

If the message count to be delivered to the consumers approaches the PrefetchThreshold, then only the client will send a request to fetch the number of messages which are set by PrefetchCount value, from the Server.

DefaultStorageTypeForQueues

String specifying the default storage type, for the data related to all the queues, to be used by the application.

Values:

The default value set for this parameter is **File**. And the valid values are **File / RDBMS**. In case this parameter is set to **File**, all the relating data about queues is stored in File based storage and of this parameter is set to **RDBMS**, all the relating data about queues is stored in RDBMS sbased storage.

Dependencies:

If this value is set to **RDBMS**, then the parameter **EnableRDBMS** should be set to **true** and if this value is set to **File**, then it should be set to **false**.

MaxPrefetchSize

Specifies the maximum prefetched size of fetched messages in one receive call. It denotes the maximum size of the batch of messages that can be fetched from the server.

Values:

The default value set for this parameter is 264144, in bytes which comes to 256KB. And the valid values are any positive integral value.

Example:

Suppose all the parameters are set to default values. Also suppose, if the size of the each message is 100KB, then at most three messages can be fetched from the server. This is because; the total batch size has already exceeded the MaxPrefetchSize which is ~256KB.

OverflowFromBottomEnabled

In over flow of non-persistent messages, specifies whether messages should be dropped from the bottom of queue.

Values:

- Default: False
- Valid: True/False

DMQExpiryTime

Time in milliseconds after which any stored message in **Dead Message Queue** is deleted. If this value is set to some positive value, a timer will be registered which cleans up the SYSTEM_DEADMESSAGES_QUEUE in regular intervals.

Values:

The default value set for this parameter is **0** in milliseconds. And the valid values are any positive value which falls in **Long** range.

EnableNotificationOnDeadMessage

Enabling the notification on DeadMessage. So, whenever an expired message from a queue reaches the SYSTEM_DEADMESSAGES_QUEUE for storage, if this parameter is set to **true** an event is raised.

Values:

The default value set for this parameter is **false**. And the valid values are **true / false**.

Dependencies:

Please note that the DMQ should be enabled on the queue on which the message was originally sent. Only then, it will be sent to SYSTEM_DEADMESSAGES_QUEUE on expiry. Also, event notification is only fired if the **NotifyWhenDead** flag for FioranoMessage is set to true. The getter and setter methods for this flag are **getNotifyWhenDead** and **setNotifyWhenDead** respectively.

EnableMessageMonitoring

Enables the message monitoring and stores the data. This will take a maximum of five seconds to start.

Values:

The default value for this parameter is **false**. And the valid values that can be assigned to this parameter are **true / false**.

FilePath

Returns the file path set to store server message monitoring data.

Values:

The default value for this parameter is %FIORANO_HOME%\fmq\profiles\FioranoMQ\run\PTP\JMSX_MESSAGEMONITOR. And any valid file path can be assigned to this parameter.

Dependencies:

To store server message data, EnableMessageMonitoring flag should be set to true.

TimeDuration

Returns the time duration in **seconds** to monitor message flow. This value should be more than five seconds. This is used to calculate the InBound and OutBound message rate while monitoring server message data.

Values:

The default value for this parameter is **3600**, in seconds. And the valid values that can be assigned to this parameter are any positive integral value that is greater than five.

CleanupInterval

Specifies the interval after which QueueCleaner utility cleans the messages from the queue. This parameter is particularly useful only if **IsDbCleanupEnabled** is set to true.

Values:

- Default: 60000
- Valid: Any positive value in long range.

Example:

In case of deleting expired messages from the queue, this parameter is used. After certain period of time, the queue gets cleaned up from any expired messages.

InMemoryBlockingInterval

Blocking interval to push a message into QueueBuffer if queue is full.

Values:

- Default: 10
- Valid: Any positive value in long range.

Example:

Blocks sender for that amount of time in case QueueBuffer is full.

InMemoryBufferSize

Specifies the buffer size of the In-Memory Cache. In case the InMemoryBuffer is full, then re-attempts to push the message will take place, based on the value, the parameter **MaxPushAttempts** is set with a wait between each attempt. This wait interval is set using the parameter **WaitBetweenAttempts**.

Values:

- Default: 1048576
- Valid: Any positive value in long range

PersistentInMemoryBufferSize

Specifies the buffer size of the InMemory Cache for persistent messages. In case the PersistentInMemoryBuffer is full, then re-attempts to push the message will take place, based on the value, the parameter **MaxPushAttempts** is set with a wait between each attempt. This wait interval is set using the parameter **WaitBetweenAttempts**.

Values:

- Default: 524288
- Valid: Any positive value in long range.

MaxPushAttempts

Maximum number of attempts for pushing it InMemoryBuffer. This parameter is particularly useful when the InMemoryBuffer is full. It makes a maximum of these many number of attempts to push the message to InMemoryBuffer.

Values:

- Default: 64
- Valid: Any positive integral value.

WaitBetweenAttempts

Specifies the wait time between attempts to push a message into the queue. This parameter is particularly useful when attempts to push the incoming message is not successful in one attempt. Then, it will wait for this much amount of time before trying to push the message into InMemoryBuffer.

Values:

- Default: 16 (in msecs)

- Valid: Any positive value in long range.

IsDbCleanupEnabled

Specifies whether queue cleaner utility is enabled or not

Values:

- Default: False
- Valid: True/False

UseInMemStatusCache

Status Cache for Messages.

Values:

- Default: True
- Valid: True/False

True - Uses InMemoryCache to store status of the messages in DB (Deleted/Rolledback/Delivered and so on).

LoadBitSetOnStartup

Loads inMemory message index status BitSet on startup.

Values:

- Default: False
- Valid: True/False

EnableGMSQueueExpose

Boolean indicates about the exposing QueueAttributes at runtime on GMS. If set to true, a runtime MBean related to the GMS queue is exposed.

Values:

- Default: False
- Valid: True/False

Exclusive

Boolean indicates about the sharing of cache tables. If set to true, then exclusive cache tables are created for each queue. And right now, non-exclusive table is not supported in FioranoMQ.

Values:

- Default: True
- Valid: True/False

JumpDeletedFiles

Boolean specifying whether the deleted table files can be jumped or not. In case of finding the index of next undeleted message while reading from DB, this flag is used, so that it jumps the deleted cache table files.

Values:

- Default: False
- Valid: True/False

DepthMonitoringEnabled

This will turn on or off depth (pending message count) monitoring at server level. If this flag is set to true, then depth monitoring will be enabled at server level. Based on the depth monitoring flag at queue level, depth monitoring will be enabled or disabled for that particular queue. If this flag is set to false, then depth monitoring won't work for any queue irrespective of this flag set at queue level.

Behavior at server level

If this value is set to no at server level, queue depths monitoring is stopped and notifications are not sent for any queues. If this value is set to yes, then queue depth monitoring is enabled or disabled based on the configuration for this property at the queue level.

Behavior at destination level

This property will come into effect only if the queue depth monitoring is enabled at server level.

Values:

- Queue Subsystem level (server level): no
- Destination level: no

DepthMonitoringLevels

Jmx notifications will be fired, whenever pending message count crosses one of these depth values. All thresholds should be specified as positive values with each of them separated by comma, for example '10,20,30'. This value will be taken for all destinations, unless specified explicitly for a particular destination.

Behavior at server level

Depth levels defined here will be used only if the value for this property at queue level is null.

Behavior at destination level

Depth levels defined here will override the value for levels specified at server level.

Values:

- Server level: null
- Destination level: null

Chapter 4: Topic Subsystem Level Configuration

Note: From FioranoMQ 9.1.0 release onwards, a unique Destination Level Configuration support is included. For more on this, please refer to section **29.1 Support for Destination Level Configuration** of *FioranoMQ Handbook*.

DupsOkBatchSize

This is number of messages received by the consumer for every consumer acknowledgement. An user can specify number of messages after which client should send acknowledgement to the server. This will have effect only when client application is consuming messages and acknowledgement mode is DUPS_OK. Acknowledgement mode is specified while creating session. See JMS specifications for more details on how to create a session.

Values:

The default value for this parameter is 20. Any positive integer can be given. Negative values will not throw any exception, so always positive integers must be given.

Example:

This mode is used when duplicate messages are acceptable. There is a possibility of duplicate messages delivered when client or server is down.

Dependencies:

This flag is purely depends on the consumer session acknowledgment mode.

RedeliveryTriesOnListenerException

This parameter specifies number of redelivery tries when RuntimeException is thrown in the consumer's message listener. This flag is used only when consumer's session acknowledgment mode is auto_ack or dups_ok_ack. The redelivered message will be discarded once this value is reached. This parameter will be useful if the OnMessage logic in the consumer may fail occasionally because of threads or CPU usage is very high.

Values:

This parameter default value is two. Any positive value can be given. A negative value to this parameter will make the client application not to deliver the message to consumer.

Example:

This flag is used especially when message is inserted in some other database or sent to some other server which may fail first time and subsequent calls may pass. The OnMessage logic of client should make sure that only RuntimeException is thrown.

Dependencies:

This flag is purely dependent on the consumer session acknowledgement mode.

DelayInMsgDeliveryOnListenerException

The parameter specifying the delay in message redelivery when RuntimeException is thrown in essage istener. This is used in AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE mode only. The redelivered message is discarded once max redelivery tries value is reached.

Values:

The default value set for this parameter is **-1**, which means that there is no delay between redelivery tries for a message, when RuntimeException occurs in MessageListener. And any positive value which in **Long** range can be assigned to this parameter. And there will not be any delay between attempts, if any negative value is assigned to this parameter.

Dependencies:

Only applicable when the Client Session is either in AUTO_ACK or DUPS_OK_ACKNOWLEDGE modes.

PublishWait

This parameter specifies the wait interval for the publisher for the next publish call in case server is not able to accommodate messages in memory. This is the minimum wait time when server crosses the PublishBackOffThreshold limit. This flag is used in scenarios where the subscriber's processing logic is going to take significant amount of time.

Values:

The default value for this parameter is 50 milliseconds (this cannot take values less than equal to zero).

Example:

This flag can be used in all scenarios for pub/sub model. This flag should be increased with a large message and more processing time for a subscriber.

Dependencies:

This flag will be used only if EnablePublisherSlowDown is set to yes.

PublishWaitTimeRecheckUnits

Singular wait parameter for flow control algo. Whenever the PSQ size reaches 95% then the publisher is made to wait repeatedly until the size drop downs to 85%. The MaxWaitTime is distributed into chunks after which it again checks the size of the PSQ. This applies to all the publishers.

Values:

The default value for this parameter is eight. And any positive integral value can be assigned to this parameter.

Example:

In case of a system, where there are certain number of slow subscribers, this flag comes to use.

Dependencies:

This parameter will be used only when **EnablePublisherSlowdown** is set to **true** and **MaxPersistentStoreSize** is not set to **-1**.

MaxNumberOfTopics

Maximum number of topics that can be created, by default its -1, that is infinite. (It also includes Temporary Topics). In case it is set to a finite value, the JMS provider will allow to create up to these number of queues and any attempt to create beyond this number will be unsuccessful and a proper message will be propagated in case this limit is reached.

Values:

The default value set for this parameter is **-1** which means infinite and the upper limit for the number of queues that can be created is **unlimited**. And non-negative integral number & -1 can be assigned to this parameter.

EnablePublisherSlowdown

Whether slowdown the publisher when message is stored in disk after session buffer overflows due to slow subscriber and the PSQ size has reached its threshold value. Setting this parameter to false may lead to message loss for slow subscribers. This property applies to all the publishers.

Values:

The default value for this parameter is **true**. And the valid values that can be assigned to this parameter are **true/false**.

Example:

In case of a system, where there are certain number of slow subscribers, this flag comes to use.

Dependencies:

Publish Slowdown behavior occurs only when MaxPersistentStoreSize is set to a finite positive integral value.

DropOldestMessage

Flag indicating whether oldest message should be dropped or the latest message should be dropped after persistent storage size (PSQ) crosses the maximum value and all the server buffers are filled.

Values:

The default value for this parameter is **true**. And the valid values are **true / false**.

Dependencies:

Generally this parameter comes to use only when PublisherSlowDown is disabled in the server and slow subscribers are active. This applies at the session level of the subscriber. This situation can also occur when the PSQ has been filled for persistent messages.

DefaultStorageTypeForTopics

Get the default storage types for Topics.

Values:

The default value set for this parameter is **File**. And the valid values are **File / RDBMS**. In case this parameter is set to **File**, all the relating data about topics is stored in file based storage and of this parameter is set to **RDBMS**, all the relating data about topic is stored in RDBMS sbased storage.

Dependencies

If this value is set to **RDBMS**, then the parameter **EnableRDBMS** should be set to **true** and if this value is set to **File**, then it should be set to **false**.

StoreMessageToDisk

Store message to disk when session buffer overflows due to slow subscriber. The messages will be stored in the PSQ only when this is set to true. In case it is set to false then the messages will be directly added to the ConnectionBuffer which will lead to extreme slowdown of publishers and message loss in extreme cases. This applies to all the topics.

Values:

The default value for this parameter is **true**. And the valid values that can be assigned are **true / false**.

EnableSnooperOnAllTopics

Specifies whether Snooper Service is enabled on all queues. When snooping is enabled on a queue, all the messages that are sent to this queue are copied to a topic by name **SYSTEM_MESSAGESNOOPER_TOPIC**, so that, any subscribers registered to this topic can snoop the messages sent to that particular topic.

Values:

The default value set for this parameter is **false**. The valid values are **true/false**. When set to **true**, a SnooperListener is added to each one of the topics and all the incoming messages are snooped and added to the topic **SYSTEM_MESSAGESNOOPER_TOPIC**.

Example:

This parameter can be used when it is needed to monitor the kind of messages are being sent to the topics by the clients. In order to do this, just enable this parameter and subscribe to the topic **SYSTEM_MESSAGESNOOPER_TOPIC**.

EventTopicName

Specifies the name of the topic on which events are published and processed.

Values:

The default value for this parameter is **Events_Topic** and is not assignable.

IsForceFullyKillDSubscriber

Allow forcefully killing of Durable Subscriber. If set to true, then if a new Durable Subscriber is getting registered with the server whose Subscription ID is already present as active with the server then the Former Durable Subscriber will be forcefully killed and the later will be successfully registered with the server. This applies to all the Durable Subscribers.

Values:

The default value for this parameter is **false**. And the valid values are **true/false**.

Dependencies:

This flag is used only in case of Durable Subscribers. This parameter has no effect if **EnsureUniquenessOfClientID** is set to true

MaxPersistentStoreSize

Maximum size of the persistent queue in disk. If EnablePublisherSlowDown is enabled, then this limit will never be reached, and in case that is disabled, then any NP messages coming in after the limit has exceeded will be dropped. In case IgnorePSQSizeForPersistentMessages is also disabled at the same time, then even Persistent Messages will be dropped after the limit has reached. This buffer is popularly known as PSQ and applies to each session of a slow subscriber.

Values:

The default value for this parameter is 1073741824, in bytes. And the valid values are any positive value in **Long** range and **-1**. If set to -1, it is assumed that, the max limit for PSQ size is unlimited.

UseMessagePersisterForDurableSubscriber

This parameter is used to Enable/Disable usage of a separate algorithm for persisting messages for a Durable Subscriber.

Values:

The default value for this parameter is **false**. And the valid values are **true / false**.

MaxMsgInOneRecv

Maximum number of messages that can be received in one call. This applies to all the subscribers

Values:

The default value for this parameter is **128**. And the valid values for this parameter are any positive i**Integral** value.

BasicAllowedSize

Maximum buffer limit on a per connection basis. In case it exceeds, then every push in the connection buffer will have to wait until some space is created. Generally this buffer is not exceeded as it is taken care by the flow control between the session layers at the RTL and server.

Values:

The default value for this parameter is **131072**, in bytes. And the valid values that can be assigned to this parameter are any positive integral value.

UseOptimizedTCPReceive

Use optimized TCP Receive is enabled. This algo was introduced in FioranoMQ 7.5 version and is meant to provide high performance at TCP layer on the subscriber side. This applies to all the Subscribers.

Values:

The default value for this parameter is **true**. And the valid values for this parameter are **true / false**.

EnableMessageMonitoring

Enables the message monitoring and stores the data. If this is set to true, then the data-traffic on the topic is stored InMemory as well as to disk. Whenever the user wishes to monitor the traffic he can do so by viewing the files or by calling the required JMX API. This applies to all the Topics.

Values:

The default value for this parameter is **false**. And the valid values that can be assigned to this parameter are **true/false**.

FilePath

Returns the file path set to store server message monitoring data, data will be stored only if EnableMessageMonitoring is set to true.

Values:

The default value for this parameter is **%FIORANO_HOME%\fmq\profiles\FioranoMQ\run\PUBSUB\JMSX_MESSAGEMONITOR**. And any valid file path can be assigned to this parameter.

Dependencies:

To store server message data, EnableMessageMonitoring flag should be set to true.

TimeDuration

Time duration in **secs** to monitor message flow. This value should be more than five seconds. After every interval the traffic will be monitored if EnableMessageMonitoring is set to **true**.

Values:

The default value for this parameter is **3600**, in seconds. And the valid values that can be assigned to this parameter are any positive integral value that is greater than 5.

PurgeLevelIndex

Number of persistent messages after which purging is to be done if **optimized tcp receive** is false. This applies to all the Topics.

Values:

The default value for this parameter is **100**. And the valid values that can be assigned to this parameter are any positive Integral values.

Dependencies:

This parameter is used only in case of **persistent** messages and only when **UseOptimizedTCPReceive** is set to **false**.

MinPurgeDifference

Minimum number of persistent messages after which purging can be done. Generally purging is done in batches in the server and this is the minimum number of delete request which the server will accumulate before actually deleting the messages. This applies to all the topics.

Values:

The default value for this parameter is **50**. And the valid values that can be assigned to this parameter are any positive integral values.

Dependencies:

This parameter is used only in case of **persistent** messages and only when **UseOptimizedTCPReceive** is set to **false**.

AllowDeletionOfSubtopics

Boolean determining if deletion of SubTopics is allowed. If set to true then deleting a topic will automatically delete the SubTopics. This applies to all the topics.

Values:

The default value for this parameter is **true**. And the valid values that can be assigned to this parameter are **true/false**.

NPMessagePublishTimeout

This is the maximum time for which a publisher will wait while trying to push the messages to a connection level buffer when it is full or trying to push the message to the PSQ after it has reached 95% of its maximum limit. It mainly applies to **non_persistent** messages as for **persistent** messages. The wait never ends until buffer reaches 85% again. This applies to all the publishers.

Example: NPMessagePublishTimeout = 1

Setting to a low value (as we've done), ensures that the publisher essentially does not slow down. So a "low value setting" on this parameter achieves the same result as setting the parameter "EnablePublisherSlowdown = False"

Values:

The default value for this parameter is **120000** in milliseconds. And the valid values that can be assigned to this parameter are any positive integral value.

PublishBackoffThreshold

Threshold on PSQ size for backoff algo to kick-in. Once the threshold is reached then flow-control will start the publishers will be made to wait/block for the buffers to clear.

Values:

The default value for this parameter is **0.6**. And the valid values assignable to this parameter are any positive Double values < 1.

IgnorePSQSizeForPersistentMessages

Whether to persist persistent messages in PSQ even through the PSQ is full. If set to true then even if the MaxPersistentStoreSize (PSQ) is exceeded still the messages will be stored to disk. This applies to all the Publishers.

Values:

The default value for this parameter is **true**. And the valid values assignable to this parameter are **true / false**.

Dependencies:

This parameter is applicable only for messages whose delivery mode is **persistent**.

Chapter 5: Connection Manager Configurations

Default

Specifies whether this Connection Manager is default. All connection factories will contain the URL information from this Connection Manager. If out of all Connection Managers added, no Connection Manager has this flag set to **true**, then the connection factories will get URL information from default server URL.

Values:

The default value for this parameter is **true**. And the valid values assignable to this parameter are **true / false**.

CFNameAppender (Not being used in FioranoMQ)

Suffix for the name of ConnectionFactory created for ConnectionManager.

Values:

The default value for this parameter is null. And the valid values assignable to this parameter is any string of characters.

ReaderCacheDisabled

Boolean indicating whether for this connection cache is created on read socket or not. This cache is used while reading bytes from the Sockets. If this parameter is set to **true**, then the necessary memory is allocated for each Socket Read and if set to **false**, then memory is allocated for Socket Read, based on the requirement, thus decreasing the number of memory re-allocations.

Values:

The default value for this parameter is **false**. And the valid values that can be assigned to this parameter are **true / false**.

TCPWindowSize (Not being Used in FioranoMQ)

Used TCP window size in three way handshaking process for connection creation.

Values:

The default value for this is **133120** in bytes. And the valid values assignable to this parameter are any positive value in **Long** range.

HandshakeInWorkerThread

Boolean indicating whether HandShaking process is done while accepting the connection request from the client or in socket initialization context.

Values:

The default value for this parameter is **true** and the valid values that can be assigned to this parameter are **true/false**.

MagicTimeout

Time set as a socket timeout for created socket. If any request does not arrive on the socket within this interval then socket cleanup is done.

Values:

The default value set for this parameter is **60000**, in seconds. And the valid values that are assignable are any non-negative Integral value. Also, note that, if this value is set to zero, then it is considered to have infinite timeout.

Example:

Suppose this value is set to 60000 (1 minute) and connection is made to FioranoMQ Server and server fails to communicate,the socket gets cleaned up after one minute. In the worst case this connection can communicate with JMS Server within one minute otherwise it will be cleaned up and the Socket will be closed.

Port

Port used for binding the socket on server startup.

Values:

The default value set for this parameter is **1856**. And a valid port value is between 0 (zero) and 65535.

Example:

Suppose this value is set to **1856** which is the default value and the machine IP address is **192.168.1.209**. So, clients will have to mention **http://192.168.1.209:1856** as the provider URL, and connect to the JMS provider.

Protocol

Protocol used for establishing the connection between clients and server.

Values:

The default value set for this parameter is **TCP**. And the valid values that are assignable to this parameter are TCP/HTTP/SUN_SSL/HTTPS_SUN.

- TCP - Accepts connections based on TCP Protocol.
- HTTP - Accepts Connections based on HTTP Protocol.
- SUN_SSL - Accepts Secured Connections based on TCP Protocol.
- HTTPS_SUN - Accepts Secured Connections based on HTTP Protocol.

Dependencies:

If protocol is set to either SUN_SSL or HTTPS_SUN, then **SSLEnabled** is required to be set to **true**.

UseNagle

Boolean indicating whether Nagle Algo is used in creating the socket or not. Nagle's algorithm is a means of improving the efficiency of TCP/IP networks by reducing the number of packets that need to be sent over the network. Nagle's algorithm is used to automatically concatenate a number of small buffer messages; this process (called nagling) increases the efficiency of a network application system by decreasing the number of packets that must be sent.

Values:

The default value set for this parameter is **false**. And the valid values that can be assigned are **true / false**.

MonitoringRequest

Default monitoring request ID that is in validation of exchanged communication versions in case of connection establishment.

Values:

The default value for this parameter is **13**.

MonitoringResponse

Default monitoring request response in validation of exchanged communication versions in case of connection establishment.

Values:

The default value for this parameter is null.

EnforceJsseAuthentication

Check whether authentication should be done for JSSE client requests accept by this server or not. This value is set and is useful only for sockets in the server.

Values:

The default value for this parameter is **false**. And the valid values that can assigned are **true / false**.

- True - If client authentication is required on newly accepted connections.
- False - If client authentication is not required on newly accepted connections.

Example:

If this parameter is set to true and if the client does not provide authentication information about itself, the server socket will not continue the communication with the client and the connection will be dropped at that time.

MaxClientConnectionsCount

The maximum number of client connections for this Connection Manager. A value of -1 indicates no upper limit

Values:

The default value for this parameter is **1024**. And the valid values that can be assigned are -1 and any positive integral value. If this value is set to -1, then the maximum client connection limit is considered to be unlimited.

Example:

Say that this value is set for 200 and already 200 external connections from clients are made to the server. Then, any attempts to create more connections which are not LPC Enabled will not be successful and all these connections will get dropped.

AdminConnection (Not being used)

Serving the client requests check whether this connection is adminConnection or a Normal Connection.

Values:

The default value set for this parameter is **false**. And the valid values that can be assigned are **true / false**.

Path

Root directory in which this Connection Manager searches for digital signatures for handling secure connections.

Values:

The default value for this parameter is **certs**. And any file path that is valid can be assigned to this parameter.

Example:

By default, this parameter for the profile **FioranoMQ** will be directing to the folder **$FIORANO_HOME/fmq/profiles/FioranoMQ/certs**.

ServerAddress

InetAddress of the local machine that the MQServer needs to bind to. This option is very useful when MQServer is executed on multi-homed machines.

Values:

The default value for this parameter is null. And any value that is either a valid machine name or a textual representation of its IP address can be assigned to this parameter.

Example:

If a machine has IP address <192.168.1.209> and hostname is <example.hostname>, we can give either one as ServerAddress.

HandShakingDisabled

Boolean representing whether handshake is disabled on connection creation or not.

Values:

The default value for this parameter is **false**. And the valid values that can be assignable to this parameter are **true/false**.

- True - If hand shake between clients and server is not needed after creating a new connection.
- False - To perform hand shake between clients and server after creating a new Connection.

Dependencies:

See special note

Special Note:

While setting this value to true in the server, make sure you set a System Property **DisableHandShake=true** in the client JVM while launching it.

Caution: Setting this may make server DOS attack prone, and client JVM may hang if unable to set Client System Property for the same.

ManagerClassName

Class Name of the Security Manager for wrapping TCP/HTTP communication.

Values:

The default value for this parameter is **fiorano.jms.ex.sm.def.DefaultJSSESecurityManager**. And its the only valid value that can be assigned to this parameter.

Example:

This is an implementation of IExSecurityManager. One can write and plug-in one's own server Security Manager. This can be used to manage certificates which are used for handling secure connections.

Dependencies:

This parameter is used only when SSLEnabled is set to **true**.

FioranoMQ Server

Boolean determines the connection created is for the FioranoMQ Server or for external servers.

Values:

The default value for this parameter is **true**. And the valid values for this parameter is **true / false**.

- True - In case of FioranoMQ Server.
- False - In case of FES/FPS Servers.

UseForPeerToPeerCommunication (Not being used in FioranoMQ)

Boolean which determines whether connectionManager created is to be used for communication between Peer servers (Applicable to FPS only).

Values:

The default value for this parameter is **false**. And the valid values for this parameter is **true / false**.

Chapter 6: Dash Board Configurations

JettyServer

Following are the parameters present in **Fiorano->etc->JettyServer**

MaxThreads

Maximum number of threads that can created for the embedded jetty server running.

Allows the user to configure the maximum number of threads that the dashboard can have running at any time. This value can be decreased if the user does not expect much traffic being directed towards the jetty server.

Default Value:

250 - The jetty server thread pool has a limit of 250 threads.

EnableStart

Boolean value describing whether jetty server should be started or not with FioranoMQ Server.

If the user does not wish the jetty server to be running (in case the user does not plan on using the WMT and related features) then setting it to false will prevent the jetty server from running.

Valid Values:

- Yes (in Studio)/True - The jetty server is started with the FioranoMQ Server.
- No (in Studio)/False - The jetty server is not started with the FioranoMQ Server.

Default Value:

Yes (In Studio)/True - The jetty server is started with the FioranoMQ server.

MinThreads

Minimum number of threads that should created for embedded dashboard running.

Allows the user to configure the minimum number of threads that should be present in the dashboard thread pool. This value can be increased in case the user expects high traffic being directed towards the dashboard.

Default Value:

10 - There should be a minimum of 10 threads in the jetty server's thread pool.

LowThreads

Low Threads for the embedded jetty server.

This allows the user to set the low resource threads threshold for the thread pool of the jetty server. When a low resource state is detected a threshold or a limit is set on the number of threads that the jetty server's thread pool can have. This is specified by this parameter.

Default Value:

25 - When a low resource is detected, the jetty server's thread pool's threshold is set to 25

MaxIdleTime

MaxIdle Time for the embedded jetty server

This value sets the maximum idle time for a connection. The max idle time is applied:

- When waiting for a new request to be received on a connection.
- When reading the headers and content of a request.
- When writing the headers and content of a response.

Default Value:

30000 - The maximum time in milliseconds for which the jetty server connection can remain idle.

Dependencies:

LowResourceMaxIdleTime

LowResourceMaxIdleTime

The Low Resource Max Idle Time of the embedded jetty server.

This value allows the user to reduce the max idle time of the connection when a low resource status is detected.

Default Value:

5000 - The maxIdleTime when there is a low resource state.

Dependencies:

MaxIdleTime

PortNumber

The port used by the embedded jetty server.

This value allows the user to configure the jetty server to run on different ports. When the user wants to start two or more instances of the FioranoMQ server on the same box, then the jetty server port for each of these server instances should be different as all of them cannot bind to the same port.

Default Value: 1780 - The default port for the jetty server

Chapter 7: Security Configuration

NativeFileBasedAclManager

Following are the parameters present in **FioranoMQ->Fiorano->Security->AclManager->NativeFileBasedAclManager.**

MaxAcePerAcl

This parameter provides a way to limit the maximum number of access control entries per ACL.

Valid value:

Any string

Default value is **100**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to 0 are as good as each other.

FileDBManager3

This section explains all the parameters present in **FioranoMQ->Fiorano->Security->AclManager->FileDBManager->FileDBManager3**

Path

The security configuration, namely the ACL, is stored in the file based data storage in the the path specified here.

Valid values:

Default value is **SDB/REALM.ACL**

The path specified is relative to the run directory of the respective FioranoMQ profile.

DeletedThresholdPercent

A cache stores and manages the list of all the tables pertaining to group based information in this file-storage based ACL. After repeated modification of the ACL, old entries in the table are marked deleted. This value provides a threshold percentage of deleted entries after which the Cache Compaction Process is started, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with **DeletedThresholdCount** ensures it is not invoked too often.

Valid values:

Default value is **50**, which means if the ratio of deleted entries to total entries is equal to or greater than 0.5 then the Cache compaction process is started. Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

All values greater than or equal to 100 are as good as each other.

Example:

Use a factor that best defines the ratio of deleted entries/total entries in the table. If there is good chance that many entries are marked deleted quite often then choose a higher number.

DeletedThresholdCount

A cache stores and manages the list of all the tables pertaining to group based information in this file-storage based ACL. After repeated modification of the ACL, old entries in the table are marked deleted. This value provides a threshold count of deleted Entries for starting the Cache Compaction Process, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with DeletedThresholdPercent (No. 2) ensures it is not invoked too often.

Valid values:

Default value is **10**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

Example:

Although the DeleteThresholdPercent (No. 2) provides a ratio based method of ensuring that the cache compaction process does not occur too often, it does not provide an absolute lower limit on the number of deleted entries that triggers a cache compaction. Choose a value that you feel is high enough to trigger a cache compaction.

DbTableNamePrefix

DB table names are usually prefixed with a string that can be modified by this property.

Valid value:

Default value is **#**

Any String

Example:

If you want to save the present state of the DB table and are considering using the saved state sometime in the future, then modify the prefix to some other relevant string and change it back to the original string when you want to retrieve the saved state of the DB table.

NativeFilePrincipalManager

This section explains all the parameters present in **FioranoMQ->Fiorano->Security->PrincipalManager->NativeFilePrincipalManager**

MaxPasswordLength

Each principal is allotted a password and this parameter provides a way to the configure the maximum allowable password length.

Valid value:

Default value is **50**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

MaxMemberCount

This parameter allows limiting the maximum number of members a principal/group can have.

Valid value:

Default value is **50**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

FileDBManager2

This section explains all the parameters present in **FioranoMQ->Fiorano->Security->PrincipalManager->FileDBManager->FileDBManager2**

Path

The security configuration, for the principal, is stored in the file based data storage in the the path specified here.

Valid value:

Default value is **SDB/REALM.PRINCIPA**L

The path specified is relative to the run directory of the respective FioranoMQ profile.

DeletedThresholdPercent

A cache stores and manages the list of all the tables pertaining to principal related information in this file-storage based table. After repeated modification of the table, old entries in the table are marked deleted. This value provides a threshold percentage of deleted entries after which the Cache Compaction Process is started, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with DeletedThresholdCount ensures it is not invoked too often.

Valid value:

Default value is 50, which means if the ratio of deleted entries to total entries is equal to or greater than 0.5 then the Cache compaction process is started.

Note:

- Range of int values in java. (-2^32 to 2^32 -1)
- All values less than or equal to zero are as good as each other.
- All values greater than or equal to 100 are as good as each other.

Example:

Use a factor that best defines the ratio of deleted entries/total entries in the table. If there is good chance that many entries are marked deleted quite often, then choose a higher number.

DeletedThresholdCount

A cache stores and manages the list of all the tables pertaining to principal related information in this file-storage based table. After repeated modification of the table, old entries in the table are marked deleted. This value provides a threshold count of deleted entries for starting the Cache Compaction Process, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with DeletedThresholdPercent (No. 2) ensures it is not invoked too often.

Valid value:

Default value is **10**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

Example:

Although the DeleteThresholdPercent provides a ratio based method of ensuring that the cache compaction process does not occur too often, it does not provide an absolute lower limit on the number of deleted entries that triggers a cache compaction. Choose a value that you feel is high enough to trigger a cache compaction.

DbTableNamePrefix

DB table names are usually prefixed with a string that can be modified by this property.

Valid value:

Default value is **#**

Any String

Example:

If you want to save the present state of the DB table and are considering using the saved state sometime in the future, then modify the prefix to some other relevant string and change it back to the original string when you want to retrieve the saved state of the DB table.

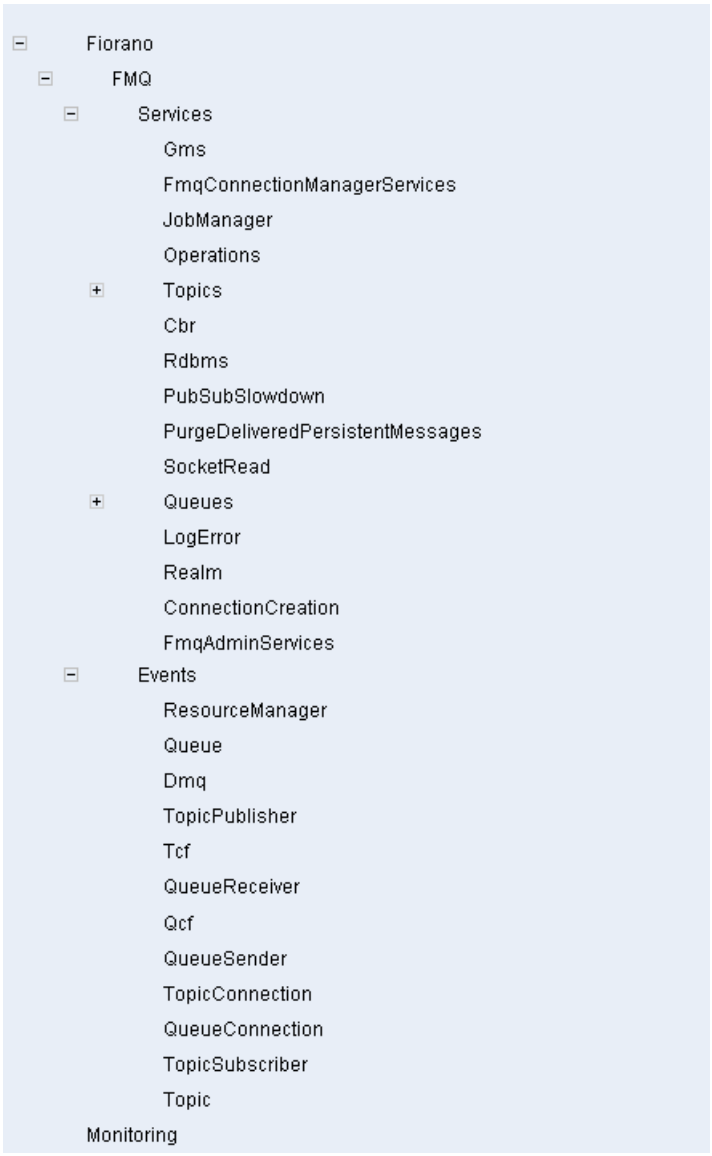
Chapter 8: Logger Configuration

Logging in FioranoMQ is achieved with the help of Log4J, with which it is possible to enable logging at runtime without modifying the application binary. Any log entry is logged at a particular log level. Each logger has with itself associated a logging level, which can be anyone of the following values.

Value	Logger	Description
-1	INHERIT	Inherits log level from its parent
0	QUIET	No logged information
1	FATAL	Severe errors that might cause premature termination
2	ERROR	Other runtime errors or unexpected conditions
3	WARN	Use of deprecated APIs, poor use of API, **almost** errors, other runtime situations that are undesirable or unexpected, but not necessarily "wrong".
4	INFO	Interesting runtime events (startup/shutdown)
5	DEBUG	Detailed information on the flow through the system
6	TRACE	More detailed information
10	ALL	Any logged information

Table 1: Log levels

These values are defined in such a way that given a value **x**, all entries logged with a value less than or equal to **x** will be received by this logger.


Fiorano
FMQ
Services
Gms
FmqConnectionManagerServices
JobManager
Operations
Topics
Cbr
Rdbms
PubSubSlowdown
PurgeDeliveredPersistentMessages
SocketRead
Queues
LogError
Realm
ConnectionCreation
FmqAdminServices
Events
ResourceManager
Queue
Dmq
TopicPublisher
Tcf
QueueReceiver
Qcf
QueueSender
TopicConnection
QueueConnection
TopicSubscriber
Topic
Monitoring

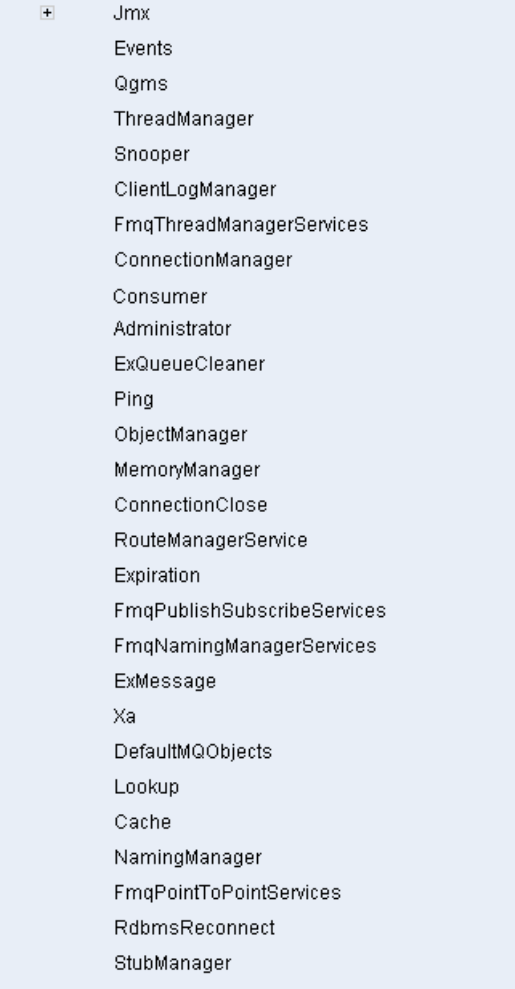


Figure 1: Loggers in hierarchical manner

Additionally the loggers are arranged in a hierarchical manner with parent-child relationships. Each element in the tree (Figure 1) has these two attributes as shown Figure 2.

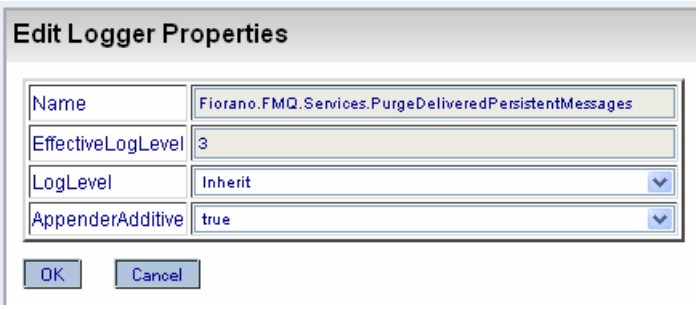


Figure 2: Edit Logger Properties dialog box

The log level parameter of these loggers can be set to any one of the values present in Table 1. The information present in the log files would depend on the value set. Some examples of using the logger configurations are given below.

If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Message is set to trace (that is, 6), all the messages (published to a topic) along with its properties are logged.

If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Publish is set to trace (that is, 6), the life cycle of the message (that is, various stages of send job) is logged.

If the 'Log Level' attribute of Fiorano.Services.Queues.Push.Message is set to trace (that is, 6), all the messages (sent to a queue) along with its properties are logged.

If the 'Log Level' attribute of Fiorano.Services.Queues.Pop is set to trace (that is, 6), the life cycle of the message after it has been pushed into the queue is logged.

If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Message is set to trace (that is, 6), all the messages (published to a topic) along with its properties are logged.

If the 'Log Level' attribute of Fiorano.Events.TopicPublisher is set to trace (that is, 6), all the events of publisher like open and close are logged.

If the 'Log Level' attribute of Fiorano.Events.QueueReceiver is set to trace (that is, 6), all the events of receiver, like open and close are logged.

If the 'Log Level' attribute of Fiorano.Services.Jmx.Operations is set to trace (that is, 6), the details of all Jmx operations performed are logged.

If the 'Log Level' attribute of Fiorano.Services.Jmx.Attributes is set to trace (that is, 6), the details of all attributes accessed using Jmx connections are logged.

AppenderAdditive

Specifies the AppenderAdditivity for this Logger. The output of a log statement of a logger C will go to all the appenders in C and its ancestors. This is the meaning of the term **appender additivity**. However, if an ancestor of logger C, say P, has the additivity flag set to false, then C's output will be directed to all the appenders in C and it is ancestors up to and including P but not the appenders in any of the ancestors of P.

Valid value:

Default value is **yes**

- **yes** - Logging output of present element will be directed to its parent as well.
- **no** - Logging output of present element will not be directed to its parent.

LogLevel

Level of logging. No log event will be generated for a logging statement above this level.

Valid value:

Default value is **-1**.

Legal values: -1 to 6. Refer table 1 for more information.

LogAppender

Following are the parameters present in **Fiorano->config->LogAppender**

AppenderName

Parameter to provide a unique Appender Name.

Valid value:

Default value is **LogAppender**

Any String

AppenderType

Parameter to specify the type of appender to be used. A LogAppender's type decides the endpoint which receives the logging information.

Valid value:

Default value is file

- **file** - the appender in use will be of type file. Log messages will be written to a file.
- **console** - the appender in use will be of type console. Log messages will be written to the console.

Example:

In a situation where you feel that the logged messages qualify(that is are importance enough) for being displayed on the console then change the Appendertype to console otherwise it can be left with its default value of file.

LogPattern

This is Log4J specific format for printing logs. Please refer to Log4J documentation for more details.

Valid value:

Default value is **[%d{dd/MMM/yyyy HH:mm:ss}] %-10c{1} %-10p %m%n**

Example:

{%F, %M} %-5p [%c{1}] %m%n will print CodeFileName, MethodName where the log event is generated.

ThresholdLevel

Log events logged above this log level will not be received/logged by this Appender

Valid value:

Default value is 10

Range of int values in java. (-2^32 to 2^32 -1)

Note:

- All values less than or equal to zero are as good as each other.
- All values greater than or equal to 10 are as good as each other.

FilterPattern

Regex based string filter pattern for filtering the log events received by this Appender. Only Events which matches the filter criteria are logged.

Valid value:

Default value is null.

- Any valid regex expression.
- null - all strings are accepted by a null regular expression.

Example:

In a situation where you would like to log only the messages that match the criteria defined by the regex, then you should set the appropriate regular expression.

PrintTarget

This parameter is applicable only to console appenders and the printtarget can be chosen appropriately to reflect what target you would like the console output printed out to.

Valid value:

Default value is System.out

For any value set in this field to have any effect, AppenderType should be console.

- **System.out** - When appendertype is set to console, this will mean that the console output would be printed to the standard output stream.
- System.err - When appendertype is set to console, this will mean that the console output would be printed to the standard error output stream.

FileName

This is the name of the log file to which the log events will be logged if the AppenderType (No. 2) is file.

Valid value:

Any valid filename. Default value is **server.log**

Given a relative path for a filename, a log file is created in <current-profile>/run/logs/, or if its an absolute path for a filename then the log file is created in the respective directory

IsAppend

This option is of any use only when the AppenderType is specified as File. When the server boots up if the log file specified already exists, this option specifies whether it is overwritten or the log entries are appended.

Valid value:

Default value is **yes**

- **yes** - Log entries are appended to the already existing log file.
- **no** - The log file is overwritten with the new entries.

MaxBackupIndex

The specific type of FileAppender used by FioranoMQ is RollingFileAppender. The RollingFileAppender can be used to roll log files based on size. When the filesize exceeds **MaxFileSize** the log file will be rolled, that is, a new log file will be created and that will have an incremental index appended to its filename. This property helps to specify the maximum number of such files. Example: 2 means, we will have at most 3 log files. **file.log**, **file1.log**, **file2.log**. This property makes sense only when the **AppenderType** used is file.

Valid value:

Default value is **4**

Range of int values in java. (-2^32 to 2^32 -1)

All values less than or equal to zero are as good as each other.

Dependencies

See also MaxFileSize and AppenderType

MaxFileSize

Maximum size of a log file in bytes, after which the log file is rolled (For explanation on RollingFileAppender, see **MaxBackupIndex**. This property makes sense only when the **AppenderType** used is file.

Valid value:

Default value is 1000000

Legal values: range of int values in java. (-2^32 to 2^32 -1)
All values less than or equal to zero are as good as each other.

Dependencies

See also MaxBackupIndex and AppenderType.

MaxFilterLevel

Maximum Filter level for Log events. Events logged above this log level will not be received/logged by this Appender.

Valid value:

Range of int values in java (-2^32 to 2^32 -1). Default value is **10**

All values less than or equal to zero are as good as each other.

MinFilterLevel

Minimum Filter level for Log events. Events logged below this log level will not be received/logged by this Appender.

Valid value:

Range of int values in java (-2^32 to 2^32 -1). Default value is **1**

All values less than or equal to zero are as good as each other.

LogAppender

Monitoring->config->LogAppender

AppenderName

Parameter to provide a unique Appender Name.

Valid value:

Any String. Default value is **LogAppender**

AppenderType

Parameter to specify the type of appender to be used. A LogAppender's type decides the endpoint which receives the logging information.

Valid values:

Default value is **file**

- **file** - the appender in use will be of type file. Log messages will be written to the file.
- **console** - the appender in use will be of type console. Log messages will be written to the console.

Example:

In a situation where you feel that the logged messages qualify (that are importance) for being displayed on the console then change the Appendertype to console otherwise it can be left with its default value of file.

LogPattern

This is Log4J specific format for printing logs. Please refer to Log4J documentation for more details.

Valid value:

Default value is **[%d{dd/MMM/yyyy HH:mm:ss}] %-10c{1} %-10p %m%n**

Example:

{%F, %M} %-5p [%c{1}] %m%n will print CodeFileName, MethodName where the log event is generated.

ThresholdLevel

Log events logged above this log level will not be received/logged by this Appender

Valid values:

Range of int values in java (-2^32 to 2^32 -1). Default value is 10

Note:

- All values less than or equal to zero are as good as each other.
- All values greater than or equal to 10 are as good as each other.

FilterPattern

Regex based string filter pattern for filtering the log events received by this Appender. Only Events which matches the filter criteria are logged.

Valid values:

Default value is null.

Any valid regex expression.

null - all strings are accepted by a null regular expression.

Example:

In a situation where you would like to log only the messages that match the criteria defined by the regex, then you should set the appropriate regular expression.

PrintTarget

This parameter is applicable only to console appenders and the printtarget can be chosen appropriately to reflect what target you would like the console output printed out to.

Valid values:

Default value is **System.out**

For any value set in this field to have any effect, **AppenderType** should be console.

- **System.out** - When appendertype is set to console, this will mean that the console output would be printed to the standard output stream.
- System.err - When appendertype is set to console, this will mean that the console output would be printed to the standard error output stream.

Dependencies:

AppenderType

FileName

This is the name of the log file to which the log events will be logged if the AppenderType (No. 2) is file.

Valid values:

Default value is **server.log**

Any valid filename

Given a relative path for a filename the log file is created relative to the directory : <current-profile>/run/logs/, or if its an absolute path for a filename then the logfile is created in the respective directory

Dependencies:

AppenderType

IsAppend

This option is of any use only when the AppenderType is specified as File. When the server boots up if the log file specified already exists, this option specifies whether it is overwritten or the log entries are appended.

Valid values:

Default value is **yes**

- **yes** - Log entries are appended to the already existing log file.
- **no** - The log file is overwritten with the new entries.

MaxBackupIndex

The specific type of FileAppender used by FioranoMQ is RollingFileAppender. The RollingFileAppender can be used to roll log files based on size. When the filesize exceeds MaxFileSize the log file will be rolled, that is, a new log file will be created and that will have an incremental index appended to its filename. This property helps to specify the maximum number of such files and will have at most 3 log files. file.log, file1.log, file2.log. This property makes sense only when the **AppenderType** used is file.

Valid values:

Range of int values in java (-2^32 to 2^32 -1). Default value is **4**

All values less than or equal to zero are as good as each other.

Dependencies:

MaxFileSize and AppenderType

MaxFileSize

Maximum size of a log file in bytes, after which the log file is rolled (For explanation on RollingFileAppender see MaxBackupIndex No. 9). This property makes sense only when the AppenderType (No. 2) used is file.

Valid values:

Range of int values in java (-2^32 to 2^32 -1). Default value is **1000000**

All values less than or equal to zero are as good as each other.

Dependencies:

MaxBackupIndex and AppenderType.

MaxFilterLevel

Maximum Filter level for Log events. Events logged above this log level will not be received/logged by this Appender.

Valid values:

Range of int values in java. (-2^32 to 2^32 -1). Default value is **10**

All values less than or equal to zero are as good as each other.

MinFilterLevel

Minimum Filter level for Log events. Events logged below this log level will not be received/logged by this Appender.

Valid values:

Range of int values in java. (-2^32 to 2^32 -1) Default value is **1**

All values less than or equal to zero are as good as each other.

Chapter 9: HA Replicated Configuration

Primary Server Configuration

FioranoHAConnectionManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

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**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

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

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

Example:

The Nagle algorithm should not be enabled in situations involving **Delayed ACK** as that will result in TCP performance problems.

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).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

RealmStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->RealmStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

QGMSStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->QGMSStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

TGMSStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->TGMSStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncRealmStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncRealmStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncNamingStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncNamingStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

NamingStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->NamingStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncQueueStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncQueueStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

RpStateStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->RpStateStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is 2001.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is 3000.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpRealmManager

Following are the parameters present in **Fiorano->HA->ReplicableRealmMgr->RpRealmManager->FioranoRpRealmManager->config**

AclZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize realm related (that is ACL based) information a zip file is created of the ACL information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_acl.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

AclUnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize realm related (that is ACL based) information a zip file is created of the ACL information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **ACL_TEMP**.

Any valid filename.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

PrincipalZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize principal related (that is user based) information a zip file is created of the principal related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_principal.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

PrincipalUnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize principal related (that is user based) information a zip file is created of the principal related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **PRINCIPAL_TEMP**.

Any valid filename.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of the principal and realm related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpNamingManager

Following are the parameters present in **Fiorano->HA->ReplicableNamingMgr->RpNamingManager->FioranoRpNamingManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize naming related information a zip file is created of the naming related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_nmdb.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize naming related information, a zip file is created of the naming related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **NM_TEMP**.

Any valid filename

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the naming related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAKRPCProvider

Following are the parameters present in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

BackupHAPort

This parameter specifies the port of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **3000**

Note: This parameter is mandatory to run HA.

BackupHAIPAddress

This parameter specifies the IP address of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **localhost**.

Note: This parameter is mandatory to run HA.

SocketTimeout

Time in milliseconds before which the socket created for communication with the peer server times out.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

HaStateNotifBroadcaster

Following are the parameters present in **Fiorano->HA->HAManager->EventManager->HaStateNotifBroadcaster**

LoggerName

Parameter to provide a unique Logger Name.

Valid values:

Default value is **null**

Any String

FioranoStatusPersister

Following are the parameters present in **Fiorano->HA->HAManager->StatusPersister->FioranoStatusPersister->config**

Path

In Replication Based HA mode state information is replicated between FioranoMQ Servers through replication channels. Loading and storing the replicated state management is done by the StatusPersister. This parameter helps configure the path of file where the status is persisted.

Valid values:

Default value is **haStatus.dat**.

Any valid filename.

Given a relative path for a filename the status file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the status file is created in the respective directory.

FioranoHAManager

Following are the parameters present in **Fiorano->HA->HAManager->RpManager->FioranoHAManager->config**

Primary

This boolean parameter indicates whether the owning server acts as the primary HA Server in Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server, and the other is called the secondary server. To understand more about FioranoMQ's HA implementation refer to *FioranoMQ Concepts Guide Chapter 16 or FioranoMQ High Availability Guide*.

Valid values:

Default value is **yes**

- yes - Indicates that this server is the primary server.
- no - Indicates that this server is the secondary server.

GatewayListenerNotUpMesg

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

This parameter helps specify the message which will be displayed when the owning server finds that no listener is present on the configured port but machine is up.

Valid values:

Default value is **Connection refused**

Any String.

PingInterval

The owning HA Server pings its peer server intermittently to exchange status information. Time interval (in milliseconds) after which the remote server is pinged in Replication HA mode.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

GatewayServerPort

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

This parameter helps specify the port on which Gateway machine is listening for incoming requests.

Valid values:

Default value is **7**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **10000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

ServerName

This parameter specifies the name of the server running in Replication HA mode.

Valid values:

Default value is **FMQServer**

Any String.

Example:

Naming the server in the most relevant way that explains its purpose makes distinguishing it from other servers easier.

Choose something relevant if you want to change this parameter.

GatewayServerIPAddress

In FioranoMQ's HA implementation a gateway machine is used to detect the HA Server which is no longer available on the network. It makes sense to use the actual gateway server of the network in which enterprise server is deployed as the Gateway machine for HA.

This parameter helps specify the IP address on which the Gateway machine is listening for incoming requests.

Valid values:

Default value is **localhost**.

Example:

Choose a machine that which is least expected to be out of network, thus ensuring that it performs its function as a Gateway machine correctly.

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.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoFMQKernel

Following are the parameters present in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

In Active state the HA Server works normally. While in Passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes dead then the owning server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be undeployed when necessary.

Valid values:

Default value is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalUnDeploymentLists

In Active State the HA Server works normally. While in Passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes dead then the owning server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Since there might be external components that could have been deployed when the server was in the Active state these components will have to be undeployed.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be undeployed when necessary.

Valid values:

Default value is null.

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if it's an absolute path then it searches in the respective directory.

MQDeploymentLists

In Active State the HA Server works normally. While in Passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. These are components that have to redeployed when the server has to be able to accept connections.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be redeployed when necessary.

Valid values

Default value is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalDeploymentLists

In Active State the HA Server works normally. While in Passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Also there might be external components that would have to be redeployed when the server goes back to the Active state.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be redeployed when necessary.

Valid values:

Default value is null.

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

FioranoRpTopicManager

Following are the parameters present in **Fiorano->HA->ReplicableTopicMgr->RpTopicManager->FioranoRpTopicManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize all the topic related information a zip file is created of the topic related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_tgmsdb.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize topic related information, a zip file is created of the topic related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values

Default value is **TGMS_TEMP**.

Any valid filename.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the topic related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpQueueManager

Following are the parameters present in **Fiorano->HA->ReplicableQueueMgr->RpQueueManager->FioranoRpQueueManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize all the queue related information a zip file is created of the queue related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_qgmsdb.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize queue related information, a zip file is created of the queue related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values

Default value is **QGMS_TEMP**.

Any valid filename.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the queue related information must be done within this time.

Valid values

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

Secondary Server Configuration

FioranoHAConnectionManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

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 **3000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

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

Example:

The Nagle algorithm should not be enabled in situations involving **Delayed ACK** as that will result in TCP performance problems.

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).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncTopicStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncTopicStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

RealmStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->RealmStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

QGMSStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->QGMSStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

TGMSStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->TGMSStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncRealmStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncRealmStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncNamingStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncNamingStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

NamingStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->NamingStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SyncQueueStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncQueueStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values

Default value is null. If value is null, there is no defined range of serviceIDs. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is null. If value is null, there is no defined range of serviceIDs.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

RpStateStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->RpStateStubManager**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is 2001

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is 3000

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpRealmManager

Following are the parameters present in **Fiorano->HA->ReplicableRealmMgr->RpRealmManager->FioranoRpRealmManager->config**

AclZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize realm related (that is ACL based) information a zip file is created of the ACL information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_acl.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

AclUnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize realm related (that is ACL based) information a zip file is created of the ACL information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **ACL_TEMP**.

Any valid filename.

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

PrincipalZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize principal related (that is user based) information a zip file is created of the principal related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_principal.zip**.

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory: <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

PrincipalUnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize principal related (that is user based) information a zip file is created of the principal related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **PRINCIPAL_TEMP**.

Any valid filename

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of the principal and realm related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpNamingManager

Following are the parameters present in **Fiorano->HA->ReplicableNamingMgr->RpNamingManager->FioranoRpNamingManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize naming related information a zip file is created of the naming related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_nmdb.zip**

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize naming related information, a zip file is created of the naming related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **NM_TEMP**.

Any valid filename

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the naming related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAKRPCProvider

Following are the parameters present in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

ackupHAPort

This parameter specifies the port of the backup peer server on which the peer is listening for status requests sent by this Server.

Valid values:

Default value is **2000**

Note: This parameter is mandatory to run HA.

BackupHAIPAddress

This parameter specifies the IP address of the backup peer Server on which the peer is listening for status requests sent by this Server.

Valid values:

Default value is **localhost**.

Note: This parameter is mandatory to run HA.

SocketTimeout

Time in milliseconds before which the socket created for communication with the peer server times out.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

HaStateNotifBroadcaster

Following are the parameters present in **Fiorano->HA->HAManager->EventManager->HaStateNotifBroadcaster**

LoggerName

Parameter to provide a unique Logger Name.

Valid values:

Default value is **null**

Any String

FioranoStatusPersister

Following are the parameters present in **Fiorano->HA->HAManager->StatusPersister->FioranoStatusPersister->config**

Path

In Replication Based HA mode state information is replicated between FioranoMQ Servers through replication channels. Loading and storing the replicated state management is done by the StatusPersister. This parameter helps configure the path of file where the status is persisted.

Valid values:

Default value is **haStatus.dat**.

Any valid filename

Given a relative path for a filename the status file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the status file is created in the respective directory.

FioranoHAManager

Following are the parameters present in **Fiorano->HA->HAManager->RpManager->FioranoHAManager->config**

Primary

This boolean parameter indicates whether the owning server acts as the primary HA Server in Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server, and the other is called the secondary server. To understand more about FioranoMQ's HA implementation refer to FioranoMQ Concepts Guide Chapter 16 or FioranoMQ High Availability Guide.

Valid values:

Default value is **no**

- **yes** - Indicates that this server is the primary server.
- **no** - Indicates that this server is the secondary server.

GatewayListenerNotUpMesg

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

This parameter helps specify the message which will be displayed when the owning server finds that no listener is present on the configured port but machine is up.

Valid values:

Default value is **Connection refused**

Any String

PingInterval

The owning HA Server pings its peer server intermittently to exchange status information. Time interval (in milliseconds) after which the remote server is pinged in Replication HA mode.

Valid values:

Default value is **30000**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

GatewayServerPort

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

This parameter helps specify the port on which Gateway machine is listening for incoming requests.

Valid values:

Default value is **7**.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **10000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

ServerName

This parameter specifies the name of the server running in Replication HA mode.

Valid values:

Default value is FioranoMQ Server

Any String

Example:

Naming the server in the most relevant way that explains its purpose makes distinguishing it from other servers easier.

Choose something relevant if you want to change this parameter.

GatewayServerIPAddress

In FioranoMQ's HA implementation a gateway machine is used to detect the HA Server which is no longer available on the network. It makes sense to use the actual gateway server of the network in which enterprise server is deployed as the Gateway machine for HA.

This parameter helps specify the IP address on which the Gateway machine is listening for incoming requests.

Valid values:

Default value is localhost

Example:

Choose a machine that which is least expected to be out of network, thus ensuring that it performs its function as a Gateway machine correctly.

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.

Valid values:

Default value is 30000.

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoFMQKernel

Following are the parameters present in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

In ACTIVE State the HA Server works normally. While in PASSIVE mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the STANDALONE state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be undeployed when necessary.

Valid values:

Default value is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory : <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalUnDeploymentLists

In ACTIVE State the HA Server works normally. While in PASSIVE mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the STANDALONE state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Since there might be external components that could have been deployed when the server was in the ACTIVE state these components will have to be undeployed.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be undeployed when necessary.

Valid values:

Default value is null

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory : <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

MQDeploymentLists

In ACTIVE State the HA Server works normally. While in PASSIVE mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the STANDALONE state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. These are components that have to redeployed when the server has to be able to accept connections.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be redeployed when necessary.

Valid values:

Default value is kernel.lst,DefaultMQObjects.lst

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory : <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalDeploymentLists

In ACTIVE State the HA Server works normally. While in PASSIVE mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes DEAD then the owning server goes into the STANDALONE state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Also there might be external components that would have to be redeployed when the server goes back to the ACTIVE state.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be redeployed when necessary.

Valid values:

Default value is null

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory : <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

FioranoRpTopicManager

Following are the parameters present in **Fiorano->HA->ReplicableTopicMgr->RpTopicManager->FioranoRpTopicManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize all the topic related information a zip file is created of the topic related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_tgmsdb.zip**

Any valid filename for a zip file.

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize topic related information, a zip file is created of the topic related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **TGMS_TEMP**

Any valid filename

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the topic related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoRpQueueManager

Following are the parameters present in **Fiorano->HA->ReplicableQueueMgr->RpQueueManager->FioranoRpQueueManager->config**

ZipPath

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize all the queue related information a zip file is created of the queue related information present in this server. This parameter allows to configure the path in which that file is placed.

Valid values:

Default value is **fmq_qgmsdb.zip**

Any valid filename for a zip file

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

UnzipDir

In the Replicable mode of the FioranoMQ HA Server, consistency between the individual HA Servers must be maintained at all times. In order to be able to synchronize queue related information, a zip file is created of the queue related information. This parameter allows to configure the path in which that zip file is temporarily unzipped.

Valid values:

Default value is **QGMS_TEMP**

Any valid filename

Given a relative path for a filename the zip file is created relative to the directory : <current-profile>/run/ or if its an absolute path for a filename then the zip file is created in the respective directory.

SocketTimeout

Time in milliseconds before which the socket times out for this replicable component. The replication of all the queue related information must be done within this time.

Valid values:

Default value is **240000** (that is equal to 4 minutes).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

Chapter 10: HA Shared Configuration

Primary Server Configuration

FioranoHAKRPCProvider

Following are the parameters present in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

BackupHAPort

This parameter specifies the port of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **3000**

Note: This parameter is mandatory to run HA.

BackupHAIPAddress

This parameter specifies the IP address of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **localhost**

Note: This parameter is mandatory to run HA.

SocketTimeout

The time in milliseconds before which the socket created for communication with the peer server times out.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAConnectionManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

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**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

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

Example: The Nagle algorithm should not be enabled in situations involving **Delayed ACK** as that results in TCP performance problems.

SocketInitializationTimeout

The time (in msecs) within any client socket should identify itself and be 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).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAStateStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HAStateStubManager->FioranoHAStateStubManager->config**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is **null**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is **null**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAManager

Following are the parameters present in **Fiorano->HA->HAManager->FioranoHAManager->config**

Primary

This boolean parameter indicates whether the owning server acts as the primary HA Server in Shared HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server, and the other is called the secondary server. To understand more about FioranoMQ's HA implementation refer to *FioranoMQ Concepts Guide Chapter 16 or FioranoMQ High Availability Guide*.

Valid values:

Default value is **yes**

- **yes** - Indicates that this server is the primary server
- **no** - Indicates that this server is the secondary server

GatewayListenerNotUpMesg

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

This parameter helps specify the messages which display when the owning server finds that no listener is present on the configured port, but machine is up.

Valid values:

Default value is **Connection refused**

Any String

GatewayServerPort

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

This parameter helps specify the port on which Gateway machine is listening for incoming requests.

Valid values:

Default value is **7**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

PingInterval

The owning HA Server pings its peer server intermittently to exchange status information. Time interval (in milliseconds) after which the remote server is pinged in Shared HA mode.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **10000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

ServerName

This parameter specifies the name of the server running in Shared HA mode.

Valid values:

Default value is **FMQServer**

Any String

Example: Naming the server in the most relevant way that explains its purpose makes distinguishing it from other servers easier.

Choose something relevant if you want to change this parameter.

GatewayServerIPAddress

In FioranoMQ's HA implementation a gateway machine is used to detect the HA Server which is no longer available on the network. It makes sense to use the actual gateway server of the network in which enterprise server is deployed as the Gateway machine for HA.

This parameter helps specify the IP address on which the Gateway machine is listening for incoming requests.

Valid values:

Default value is **localhost**

Example: Choose a machine that which is least expected to be out of network, thus ensuring that it performs its function as a Gateway machine correctly.

FioranoFMQKernel

Following are the parameters present in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

In active State the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be undeployed when necessary.

Valid values:

Default value is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalUnDeploymentLists

In active mode, the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive mode, a few components of the server have to be undeployed. Since there might be external components that could have been deployed when the server was in the active state these components will have to be undeployed.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be undeployed when necessary.

Valid values:

Default value is **null**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

MQDeploymentLists

In active mode, the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive mode, a few components of the server have to be undeployed. These are components that have to redeployed when the server has to be able to accept connections.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be redeployed when necessary.

Valid values:

Default value is **kernel.lst, DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalDeploymentLists

In active State the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Also there might be external components that would have to be redeployed when the server goes back to the active state.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be redeployed when necessary.

Valid values:

Default value is **null**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

Secondary Server Configuration

FioranoHAKRPCProvider

Following are the parameters present in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

BackupHAPort

This parameter specifies the port of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **2000**

Note: This parameter is mandatory to run HA.

BackupHAIPAddress

This parameter specifies the IP address of the backup peer server on which the peer is listening for status requests sent by this server.

Valid values:

Default value is **localhost**

Note: This parameter is mandatory to run HA.

SocketTimeout

The time in milliseconds before which the socket created for communication with the peer server times out.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAConnectionManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

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 **3000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

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

Example: The Nagle algorithm should not be enabled in situations involving **Delayed ACK** as that will result in TCP performance problems.

SocketInitializationTimeout

The 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).

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAStateStubManager

Following are the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HAStateStubManager->FioranoHAStateStubManager->config**

MinServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the minimum value of that range.

Valid values:

Default value is **null**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

MaxServiceID

FioranoMQ defines a serviceID with each request associated with a subsystem and thus the range of serviceIDs that are associated with a subsystem defines the operations that are accessible to that subsystem. This value defines the maximum value of that range.

Valid values:

Default value is **null**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

FioranoHAManager

Following are the parameters present in **Fiorano->HA->HAManager->FioranoHAManager->config**

Primary

This boolean parameter indicates whether the owning server acts as the primary HA Server in Shared HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server, and the other is called the secondary server. To understand more about FioranoMQ's HA implementation refer to *FioranoMQ Concepts Guide Chapter 16 or FioranoMQ High Availability Guide*.

Valid values:

Default value is **no**

- **yes** - Indicates that this server is the primary server.
- **no** - Indicates that this server is the secondary server.

GatewayListenerNotUpMesg

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

This parameter helps specify the message which will be displayed when the owning server finds that no listener is present on the configured port but machine is up.

Valid values:

Default value is **Connection refused**

Any String

GatewayServerPort

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

This parameter helps specify the port on which Gateway machine is listening for incoming requests.

Valid values:

Default value is **7**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

PingInterval

The owning HA Server pings its peer server intermittently to exchange status information. Time interval (in milliseconds) after which the remote server is pinged in Shared HA mode.

Valid values:

Default value is **30000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

SocketCreationTimeout

In HA, the time in milliseconds within which socket has to created with the peer server failing which the socket creation attempt will fail.

Valid values:

Default value is **10000**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

ServerName

This parameter specifies the name of the server running in Shared HA mode.

Valid values:

Default value is **FMQServer**

Any String

Example: Naming the server in the most relevant way that explains its purpose makes distinguishing it from other servers easier.

Choose something relevant if you want to change this parameter.

GatewayServerIPAddress

In FioranoMQ's HA implementation, a gateway machine is used to detect the HA Server which is no longer available on the network. It makes sense to use the actual gateway server of the network in which enterprise server is deployed as the Gateway machine for HA.

This parameter helps specify the IP address on which the Gateway machine is listening for incoming requests.

Valid values:

Default value is **localhost**

Example: Choose a machine which is least expected to be out of network, thus ensuring that it performs its function as a Gateway machine correctly.

FioranoFMQKernel

Following are the parameters present in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

In active State the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be undeployed when necessary.

Valid values:

Default value is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalUnDeploymentLists

In ACTIVE STATE the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Since there might be external components that could have been deployed when the server was in the active state these components will have to be undeployed.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be undeployed when necessary.

Valid values:

Default value is **null**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

MQDeploymentLists

In ACTIVE STATE the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. These are components that have to redeployed when the server has to be able to accept connections.

These components are listed in FioranoMQ specified files. This parameter helps specify a comma separated list of files containing Fiorano specific components that have to be redeployed when necessary.

Valid values:

Default value is **kernel.lst, DefaultMQObjects.lst**

A comma separated list of .lst files.

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if its an absolute path then it searches in the respective directory.

ExternalDeploymentLists

In ACTIVE STATE the HA Server works normally. While in passive mode, the HA Server only monitors its peer and does not handle any client requests. Any client connection to the server in passive mode is refused. In case a peer server goes **dead** then the owning server goes into the **standalone** state. In order to **not accept any client connections** while in passive state, a few components of the server have to be undeployed. Also there might be external components that would have to be redeployed when the server goes back to the active mode.

These components are listed in files. This parameter helps specify a comma separated list of files containing external components that have to be redeployed when necessary.

Valid values:

Default value is **null**

A comma separated list of .lst files

Given a relative path for a filename the files are searched for in the directory: <current-profile>/deploy/ or if it's an absolute path then it searches in the respective directory.

Chapter 11: Dispatcher Configuration

FioranoJobManager2

Following are the parameters present in **Fiorano->JobManager->Impl2->FioranoJobManager2->config**

MinimumJobWorkers

This parameter is used to specify the minimum number of job workers that would be added as soon as the server is started and be present in the system at any time. This parameter can be increased or decreased depending on the expected load on each server.

Valid values:

Default value is **4**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than zero are as good as each other.

Example:

In case the server is expected to handle a large number of queues/topics simultaneoulsysimultaneously, this could be set at a higher number as we can be assured that this would not result in idle JobWorkers. Similarly for a very case where the load of the server is expected to be very less, this can be decreased so as to not constantly have idle JobWorkers in the system.

MaximumJobWorkers

This parameter allows the user to set the limit on the number of JobWorkers allowed for this JobManager. This parameter can be increased or decreased depending on the expected load on each server.

Valid values:

Default value is **40**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

All values less than 0 zero are as good as each other.

Example:

Allowing unchecked addition of JobWorkers could result in too many threads (JobWorkers) being created hampering the performance of the system. Depending on the scenario, this value can be increased or decreased.

Dispatcher

Following are the parameters present in **Fiorano->Dispatcher->config**

Name

This parameter can be used to specify a name for the dispatcher.

Valid values:

Default value is **FioranoMQ Dispatcher**

Any String

PingTimeoutAttribute

This parameter describes the time out (in msecs) after which the connection to the server is pinged.

The dispatcher has to make sure that the server it has connected to is still alive, and this is acheivedachieved by pinging it intermittently. This intermittent time interval is specified by this parameter.

Valid values:

Default value is **30000**

Legal values: range of long values in java. (-2^63 to 2^63 -1)

All values less than zero are as good as each other.

InitialTimeoutAttribute

This parameter describes the initial timeout (in milliseconds) for connecting with a server in a cluster that is the Dispatcher will time out if it cannot connect to the server within this time.

Valid values:

Default value is **2000**

Legal values: range of long values in java. (-2^63 to 2^63 -1)

All values less than zero are as good as each other.

Chapter 12: Repeater Configuration

FioranoRepeaterManager

Following are the parameters present in **Fiorano->Repeater->RepeaterManager->FioranoRepeaterManager->config**

Name

This parameter is used to specify the name of the Repeater Instance.

Valid values:

Default value is **Repeater**

Any String.

DurableSubscriptionInfoFileName

This parameter is used to specify the name of the XML file that contains information related to durable subscriptions.

Valid values:

Default value is **rp_durable.xml**

Any valid xml filename

Given a relative path for a filename the server looks for the XML file in the directory specified in the system property FMQ_CONFIG_PATH or if its an absolute path for a filename then that file is used.

PingInterval

This parameter describes the time out (in msecs) after which the connection to the server is pinged.

The repeater makes sure that the server it has connected to is still alive, and this is achieved by pinging it intermittently. This intermittent time interval is specified by this parameter.

Valid values:

Default value is **10000**

Legal values: range of long values in java. (-2^63 to 2^63 -1)

All values less than zero are as good as each other.

AvoidLoopback

This parameter is used to specify whether cyclic links are valid or not in this repeater.

Loopback occurs when you have a set of links connecting FioranoMQ Servers in such a way that, they form a cycle. If a message were to originate at a particular FioranoMQ Server then the message would traverse the entire cycle and be returned to the originating server. If AvoidLoopback is enabled then this message is not transferred further, otherwise the message stays indefinitely in the cycle.

Valid values:

Default value is **yes**

- **yes** - Loopback is avoided
- **no** - Loopback is not avoided

TransactionSize

This parameter is used to specify the number of messages that will be seen as a complete transaction.

Valid values:

Default value is **0**

Legal values: range of int values in java. (-2^31 to 2^31 -1)

All values less than or equal to zero are as good as each other.

TransactionTimeout

In a transaction a group of messages are seen together as one transaction. If a message which is a part of the transaction is not received (by the subscriber) within this timeout period (in milliseconds), the transaction commits the already received messages.

Valid values:

Default value is **10**

Legal values: range of long values in java. (-2^63 to 2^63 -1)

All values less than zero are as good as each other.

SourceServer

Following are the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->SourceServer**

Name

This parameter describes the name given to the source FioranoMQ Server.

Valid values:

Default value is **LocalhostSourceServer**

ConnectionInfo

Following are the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->SourceServer->ConnectionInfo**

ProtocolType

This parameter describes the underlying protocol that will be used for the connection to the source FioranoMQ Server.

Valid values:

Default value is **TCP**

JMXProtocol

This parameter describes the underlying communication protocol that will be used for the JMX connection to the source FioranoMQ Server.

Valid values:

Default value is **rmi**

JMXPort

This parameter specifies the JMX port on the machine, hosting the source FioranoMQ Server, to which the JMX connection will be made.

Valid values:

Default value is **1858**

Legal values: range of integer values in java. (-2^31 to 2^31 -1)

Note: The port must not be anything that is being used already, obviously.

ServerURL

This parameter specifies the URL of the FioranoMQ Server that will be a part of this link. This FioranoMQ Server serves as the source server in this link.

Valid values:

Default value is **http://localhost:1856**

(Format: http://hostname:port)

UserName

This parameter specifies the login name used by the repeater to connect to the source FioranoMQ Server.

Valid values:

Default value is **anonymous**

Any String

Password

This parameter specifies the login name used by the repeater to connect to the source FioranoMQ Server.

Valid values:

Default value is **anonymous**

Any String

TopicConnectionFactory

This parameter specifies the topic connection factory used by the repeater to connect to the source FioranoMQ Server.

Valid values:

Default value is **primaryTCF**

Any String

ServerSecurityManager

This parameter specifies the Security Manager used by the source FioranoMQ Server.

Valid values:

Default value is **null**

Any String

ProxyURLAttribute

This parameter specifies the URLs of the proxy servers to be used. Multiple backup URLs may be specified as a semicolon-separated string of URLs.

Valid values:

Default value is **null**

Any String

LinkTopicInfo

Following are the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->LinkTopicInfo**

IsDurable

This parameter specifies whether the link between the source and the target server is durable or not. A durable link ensures that no messages are lost across the repeater in case of network failure.

Valid values:

Default value is **false**

- **true** - Link is durable
- **false** - Link is not durable

ConnectionMode

This parameter specifies whether the same JMS connection would be used for replicating the data across JMS Servers or a separate connection would be created for each link.

Valid values:

Default value is **Shared**

- Shared - In this mode the repeater will use same JMS connection for replicating the data across JMS Servers.
- Exclusive - In this mode the repeater will use separate JMS connections for replication of data across servers. In exclusive connection mode a new Replication Link is added for each link.

Example:

The Shared connection mode can be used when the replication is directed towards multiple topics on the destination side, since the connection mode is shared only a single connection would be used for all the multiple topics.

Type

This parameter specifies whether the link should be permanently connected to the target server or only replicate if a subscriber exists.

Valid values:

Default value is **SUBSCRIBER EXISTS**

- SUBSCRIBEREXISTS - The link with the target server that is used for replication does not exist unless there is a subscriber on the target server.
- ALWAYS - In this mode the link that the repeater has with the server will be permanent.

ReplyOn

This parameter specifies the destination on which the repeater will listen for replies, which it receives on the requests it forwards.

Valid values:

Default value is **null**

SourceTopicName

This parameter specifies the name of the topic on which subscriptions are made on source server of the link.

Valid values:

Default value is **primaryTopic**

TargetTopicName

This parameter specifies the name of the topic on the target server to which messages are replicated.

Valid values:

Default value is **secondaryTopic**

MessageSelector

This parameter specifies the selector that is set on a link between the servers which can be used to ensure that only the required messages are exchanged between them.

Valid values:

Default value is **null**

Only messages whose header and property values match those of the selector are browsable. A message selector matches a message if the selector arrives at **true** when the message's header field values and property values are substituted by their corresponding identifiers in the selector. If the value of a message selector is an empty string, the value is treated as a null and indicates that there is no message selector for the message consumer.

Chapter 13: Bridge Configurations

ConnectionInfo

Following are the parameters present in **Fiorano->Bridge->FioranoBridge->FioranoConnectionManager->config-><LinkName>-><ServerName>->ConnectionInfo**

ServerType

Need to mention the type of the server.

Default Value:

JMS - The default type of server

ProviderUrl

The URL where the server is currently running.

This parameter allows the user to configure the bridge to link the queues present on different FioranoMQ servers. The URL of the source/target server should be given here.

Default Value:

http://localhost:1856 - The default provider URL of the server.

InitialContextFactory

The class name of the initial context factory which is used to open connection with the specified server.

Default Value:

fiorano.jms.runtime.naming.FioranoInitialContextFactory - The initial context factory class used.

BridgeUser

The UserName which the bridge uses to connect to the server.

Default Value:

Anonymous - The default bridge user name

BridgePassword

The password which the bridge uses to connect to the server.

Default Value:

Anonymous - The default password

QCF

The connection factory with which the connection is to be established.

Default Value:

primaryQCF - The default connection factory used for the connection.

Protocol

The transport protocol used for establishing a connection with the server.

Default Value:

TCP - The default transport protocol used for the connection.

SecurityPrincipal

The SecurityPrincipal (username) used for connecting to the server.

Default Value:

ayrton - The default username

SecurityCredentials

The SecurityCredentials (password) used for connecting to the server.

Default Value:

Senna - The default password

ServerSecurityManager

ServerSecurityManager of the server linked by the bridge.

Default Value:

null - No server Security Manager

QueueInfo

Following are the parameters present in **Fiorano->Bridge->FioranoBridge->FioranoConnectionManager->config-><LinkName>-><ChannelName>-><Target/Source Queue>->QueueInfo**

MsgSelector

The message selector of the queue. Allows the user to set message selectors for the source/target queues which have been linked using bridges.

Default Value:

null - No message selector is used

QueueName

The name of the queue present in target/source server which is to be linked by the bridge. Allows users to configure links between different queues available on the server.

Default Value:

primaryQueue - The name of the queue which the bridge links.

Chapter 14: JMX Connector Configuration

JMS Based Connector Configuration

Following are the parameters present in **JMX->Connector->JMSBasedJMXConnector**

InterceptorClassName

Class Name that would be used to intercept all jmx calls. When JMX calls arrive at the container, this class is used to intercept them and process them.

Default Value:

fiorano.jmx.connector.FioranoJMXInterceptor - The name of the class to be used to intercept all jmx calls.

SecurityProtocol

Secure protocol used in case Server to which this Connector Server will connect is running on secure protocol. Example: SUN_SSL,NO_SECURITY.

Valid values:

- NO_SECURITY - No Security protocol is used
- SUN_SSL - Sun_SSL is the security protocol used

Default Value:

NO_SECURITY -No security protocol is used by default.

Protocol

Transport protocol used for making connection to JMS Server. Allows the user to choose between the protocols available namely LPC, TCP, and HTTP.

Valid values:

- TCP - TCP is the transport protocol used
- HTTP - HTTP is the transport protocol used
- LPC - LPC is the transport protocol used

Default value:

LPC - LPC is used for making connections to the JMS Server.

SecurityManagerClass

Security Manager class which provides an implementation of IFMQSecurityManager

Valid values:

NO_SECURITY - No Security Manager class is used.

fiorano.jmx.connector.fmq.security.JSSESecurityManager - The Security Manager class name for Sun_SSL.

Default value:

NO_SECURITY - Since by default no SecurityProtocol is used, no SecurityManagerClass is used by default. However if a SecurityProtocol is used, say SUN_SSL then the corresponding SecurityManagerClass that is fiorano.jmx.connector.fmq.security.JSSESecurityManager should be used.

Dependencies:

SecurityProtocol

QueueConnectionFactory

QueueConnectionFactory used by Connector Server. It can be either LPC enabled or normal. LPC Enabled connectionfactory is used when Connector Server connects to the local server.

Default value:

primaryLQCF - The name of the default QueueConnectionFactory used.

ServerPort

Port of JMS Server to which this ConnectorServer connects.

Allows the user to configure the server to run on different ports. When the user has to run two or more servers on the same box, changing this value is mandatory as two servers can not bind to the same port.

Default value:

1856 - The default port number of the JMS server

ServerAddress

The IP of FioranoMQ Server to which this ConnectorServer connects.

Note: ServerAddress should be **localhost/LocalIP/MachineName** if using LPC QueueConnectionFactory

Default Value:

localhost - The default ServerAddress of the JMS Server.

Username

User name used for making connection to JMS Server.

Default Value:

Admin - Default username

Password

Password used for making connection to JMS Server.

ConnectQueueName

Connect Queue name on which Connector Server will listen for connect request.

Default Value: JMX_CONNECT_QUEUE

RequestQueueName

Request Queue name on which Connector Server will listen for jmx queries

Default Value: JMX_SERVICE_QUEUE

NotifQueueName

Notification Queue name on which Connector Server will send notification

Default Value: JMX_SERVICE_QUEUE

RMI Based Connector Configuration

Following are the parameters present in **JMX->Connector->RMIBasedJMXConnector**

RMIServerPort

The port on which Mx4J RMIConnector Server will bind.

Allows the user to configure the RMI port for the server. When the user has to run two or more servers on the same box, changing this value is mandatory as two servers cannot bind to the same RMI port.

Default Value: 1858 (The default RMI port of the server).

InterceptorClassName

The class Name that would be used to intercept all jmx calls

Default Value: fiorano.jmx.connector.FioranoJMXInterceptor (The name of the class to be used to intercept all jmx calls).

JMX Engine Configuration

Following are the parameters present in **Fiorano->jmx->engine->JMXEngine->ClientJMXEngine**

SecurityProtocol

Secure protocol used in case Server to which this Connector Server will connect is running on secure protocol. Example: SUN_SSL,NO_SECURITY

Valid Values:

- NO_SECURITY - No security protocol is used
- SUN_SSL - Sun_SSL is used as the security protocol
- PHAOS_SSL – Phaos_SSL is used as the security protocol

Default Value:

NO_SECURITY - No Security protocol is used.

Protocol

Transport protocol used for making connection to JMS Server.

Valid Values:

- TCP - TCP is used as the transport protocol.
- HTTP - HTTP is used as the transport protocol.
- LPC - LPC is used as the transport protocol.

Default value:

LPC - LPC is the transport protocol used.

SecurityManagerClass

Security Manager class which provides an implementation of IFMQSecurityManager

Valid Values:

NO_SECURITY - No Security Manager Class is used.
fiorano.jmx.connector.fmq.security.JSSESecurityManager – The Security Manager class name used for the Sun_SSL security protocol.

fiorano.jmx.connector.fmq.security.PhaosSecurityManager – The class name used for the Phaos_SSL security protocol.

Default value:

NO_SECURITY - No Security Manager class is used.

QueueConnectionFactory

QueueConnectionFactory used by Connector Server. It can be either LPC enabled or normal. LPC Enabled connectionfactory is used when Connector Server connects to the local server.

Default Value:

primaryLQCF - This is the default QueueConnectionFactory used.

ServerPort

Port of JMS Server to which this ConnectorServer connects.

Allows the user to configure the server to run on different ports (when two or more instances of FioranoMQ Server need to be run on the same box).

Default value:

1856 - This is the default server port.

ServerAddress

IP of FioranoMQ Server to which this ConnectorServer connects.

Note: ServerAddress should be **localhost/LocalIP/MachineName** if using LPC QueueConnectionFactory

Default Value:

localhost - The default server address

Dependencies:

QueueConnectionFactory

Username

User name used for making connection to JMS Server.

Default Value:

admin - Default username.

Password

Password used for making connection to JMS Server.

ConnectQueueName

Connect Queue name on which Connector Server will listen for connect request

Default Value:

JMX_CONNECT_QUEUE - The default queue name on which the Connector Server will listen.

RequestQueueName

Request Queue name on which Connector Server will listen for jmx queries

Default Value:

JMX_SERVICE_QUEUE - The default queue name on which Connector Server will listen for jmx queries.

NotifQueueName

Notification Queue name on which Connector Server will send notification

Default Value:

JMX_SERVICE_QUEUE - The default queue name on which Connector Server will send notification

Monitoring Manager Configuration

Following are the parameters present in **Fiorano->jmx->notifications->MonitoringManager**

ClockTickForMonitoring

Time interval of monitoring in milliseconds. The number of milliseconds after which the timer for monitoring is reset. This parameter sets the time interval in milliseconds between successive monitoring.

The value set to this parameter is assigned as the clock tick interval for the monitoring timer. Hence, using this parameter the user can control the interval between successive Monitoring.

Default Value:

5000 - Five seconds is the default time after which the monitoring timer is reset.

Chapter 15: Advanced Configuration

Route Manager Configuration

Following are the parameters present in **Fiorano->etc->RouteManager**

MaxTopicBuffer

Route level upper limit for message buffer size in bytes.

This parameter allows the user to specify the buffer size (in bytes) of the message queue. Depending on the density of the message traffic this value can be increased or decreased.

Default Value:

131072 - This is the maximum number of bytes that can be stored in the buffer.

RdbmsManager Configurations

Following are the parameters present in **Fiorano->etc->RdbmsManager**

URL

URL of the database where messages are stored. Refer to RDBMS Server's documentation for correct format.

Allows the user to set/change the URL of the Database Server.

Default Value:

jdbc:hsqldb:../profiles/FioranoMQ/run/RDBMS/FMQ - The default URL.

JdbcDriver

JdbcDriver used for RDBMS based datastorage. Refer to database driver's documentation for correct format.

Default Value:

org.hsqldb.jdbcDriver - The default driver used.

Password

Password for RDBMS User - Used for connecting to the RDBMS Server.

Username

Database User Name - Used for connecting to the RDBMS Server.

Default Value:

sa

MaxConnections

Allowed Maximum number of DB connections that would be created.

To restrict the total number of DB connections created. Having too many connections could affect the performance of the server.

Default Value:

200 - A maximum of 200 DB connections are allowed.

DefaultConnections

Number of DB connections created on startup.

If the user wants to increase the number of DB connections created on startup, this parameter should be increased.

Default Value:

1 - By default only 1 DB connection is created at startup.

MaxNoOfRowsFetch

Maximum number of rows fetched in a shot from the database as a result of any select query.

In case of having to access a very huge number of records then increasing the number of entries fetched per query statement would be faster as it would require lesser number of queries.

Default Value:

100 - The maximum number of rows (entries) fetched at a shot from the database is 100.

EnableReconnect

Boolean indicates whether reconnect needs to be enabled or not.

Allows the user to decide whether the server has to make an attempt to reconnect to the RDBMS Server when the connection is lost. In cases where is reconnect is not necessary or should not be performed this parameter should be set to False.

Valid values:

- True - Enables reconnect attempts to the RDBMS Server when the connection is lost.
- False - Disables reconnect attempts to the RDBMS Server when the connection is lost.

Default Value:

True - Enables reconnect attempts to the RDBMS Server when the connection is lost.

ReconnectTimeInterval

Time interval in milliseconds between RDBMS reconnect attempts.

Allows the user to specify the interval between successive reconnect attempts when the connection to the RDBMS Server is down. This value can be increased in case the time taken for the reset of the RDBMS Server is long.

Default Value:

10000 - Time in milliseconds between RDBMS reconnect attempts (10 seconds).

WaitTime

Wait time if no additional connections are available in the connecting Pool. 0 (zero) indicates indefinite wait.

Allows the user to configure the time for which the server waits when the connection pool is empty.

Default Value:

0 - Infinite wait

ConnectionTimeoutInterval

Connection timeout interval in milliseconds.

Allows the user to configure the connection timeout interval (that is the time after when the connection times out if no activity occurs during that time).

Default Value:

10000 - Connection timeout interval is ten seconds.

PropertiesFilename

The JDBC properties file name. This file contains database specific details for various SQL Types. Point this to the name of the file jdbc_*.cfg which should be available in %FIORANO_HOME%/fmq/profiles/%PROFILE%/conf directory.

Default Value:

jdbc_hsql.cfg - The default JDBC properties file name.

DbConnectivityQuery

Query that should be executed to check health of a connection.

Allows the user to configure the query to be used to test the health of the connection.

Default Value:

select * from tab - The default query used to check if the connection is still alive.

EnableRdbms

Enables/Disables DB Manager. Setting this flag to true is a prerequisite for using an RDBMS Server as a back-end message store.

EnableRdbms is needed to allow the user to enable/disable the RDBMS message store feature.

Valid Values:

- True - Enables the RDBMS message store feature.
- False - Disables the RDBMS message store feature.

Default Value:

True - Enables the RDBMS message store feature.

Resource Manager Configurations

Following are the parameters present in **Fiorano->etc->ResourceManager**

MaxMemoryThresholdAllowed

Maximum memory threshold allowed, value can range from 0.0 to 1.0. This parameter allows the user to set the percentage of total memory that can be used by the server. 1.0 corresponds to 100% and 0.0 corresponds to 0%.

This allows the user to restrict the memory usage and prevent too much memory being used up by the server processes.

Default Value:

0.8 - The default memory threshold is 80%.

TimeInterval

Interval after which the monitoring thread will run again in milliseconds.

This parameter will allow the user to decide on the intervals between the start of one monitoring thread and the end of the previous one. If the monitoring threads are to be run widely spaced then this could be set to a much higher value. However, if monitoring is to be done over a short period of time this could be set to a smaller value.

Default Value:

10000 - A monitoring thread runs once every 10000 milliseconds that is ten seconds.

EnableMonitoringThread

Whether to enable monitoring thread or not.

Allows the user to enable or disable the monitoring thread. In case the user did not want the monitoring thread running periodically he could set it to No (in Studio)/False.

Valid Values:

- Yes (in Studio)/True – Enables periodic running of Monitoring thread.
- No (in Studio)/False - Disables periodic running of Monitoring thread.

Default Value:

Yes (In Studio)/True - Enables the monitoring thread so that it could run periodically.

EnableVerboseLogging

Whether to enable verbose logging or not, if set to false logs will be maintained in monitor.txt file in the format:[Date][space][Time][comma][QueueConnections][comma][TopicConnections][comma][sessions][comma][senders][comma][receivers][comma][publisher][comma][subscriber][comma][threads][comma][usedMemory]

It allows the user to choose the format of the monitoring logs.

Valid Values:

- Yes (in Studio)/True - Enables Verbose Logging
- No (in Studio)/False - Disables Verbose Logging

Default Value:

Yes (in Studio)/True - Enables Verbose logging

EnableThreadDumpInfo

Whether to enable Thread Dump info in case server goes low on memory.

The user can choose whether or not to enable the thread dump info. When the server goes low on memory, if the parameter is set to Yes (in Studio)/True, then the thread info is dumped into a file so that the user will not lose the data.

Valid Values:

- Yes (in Studio)/True – Enables dumping of thread info when the server is low on memory.
- No (in Studio)/False - Disables dumping of thread info when the server is low on memory.

Default Value:

Yes (in Studio)/True - Enables the dumping of thread info when the server goes low on memory.

PingManager Configurations

Following are the parameters present in **Fiorano->etc->PingManager**

PingTimeout

Ping timeout time. The amount of time after which a dead connection is disconnected. The time is in milliseconds.

This parameter allows the user to configure the amount of time after which it should stop trying to connect to a dead connection, that is it is the timeout used for disconnecting from a dead connection.

Default Value:

60000 - The number of milliseconds after which a dead connection is disconnected.

JobManager Configurations

MinimumJobWorkers

Minimum number of JobWorkers always in the system.

Allows the user to configure the minimum number of JobWorkers that can be added as soon as the server is started and be present in the system at any time. This parameter can be increased or decreased depending on the expected load on each server. In case the server is expected to handle a large number of queues/topics simultaneously, this could be set at a higher number as we can be assured that this would not result in idle JobWorkers. Similarly for a very case where the load of the server is expected to be very less, this can be decreased so as to not constantly have idle JobWorkers in the system.

Default Value: 4 (The Minimum number of JobWorkers Available in the system is 4).

MaximumJobWorkers

Maximum number of job workers employed by the system at any instance.

The minimum number of JobWorkers is specified by the MinimumJobWorkers (18) parameter. As more jobs arrive, there is a need to increase the JobWorkers dynamically. This parameter allows the user to set the limit on the number of JobWorkers allowed for this JobManager. Allowing unchecked addition of JobWorkers could result in too many threads (JobWorkers) being created hampering the performance of the system. Depending on the scenario, this value can be increased or decreased.

Default Value: 40 (The Maximum number of JobWorkers available in the system is 40)

JobsPerWorker

Number of jobs per worker.

This parameter is used to decide whether or not to add more JobWorkers. If the number of jobs present in the job queue becomes greater than number of job workers present multiplied by JobsPerWorker, then new JobWorkers are added, provided current number of job workers is less than MaximumJobWorkers (19). This value cannot be changed.

Default Value: 4 (The default number of jobs assigned per job worker at any point of time is 4).

ObjectManager Configurations

Following are the parameters present in **Fiorano->etc->runTimeObjectManagers->ObjectMaanger/gmsObjectManager**

HashcodeReuseAlgoUsed

Specifies whether Reuse HashCode Algo is used in creation of Object Handles or not.

Valid Values:

- Yes (in Stubio)/True – HashcodeReuse Algorithm is used.
- No (in Studio)/False - HashcodeReuse Algorithm is not used.

Default Value:

Yes (in Studio)/True - The hash code reuse algo is used in the creation of Object Handles.

ThreadManager Configurations

Following are the parameters present in **Fiorano->etc->ThreadManager**

DelayBetweenAttempts

The delay in milliseconds between two successive attempts to create a thread when the thread creation fails for a critical operation thread.

If for any reason, the creation of a critical operation thread fails, this along with the MaximumAttempts attribute ensures further attempts to create the thread without affecting the performance of the server.

Default Value:

100 -100 milliseconds delay between successive attempts to create a thread.

Dependencies:

MaximumAttempts

MaximumAttempts

Maximum number of attempts to create a thread.

If the creation of a thread results in a failure, after a specified amount of time (DelayBetweenAttempts) another attempt is made to create the thread. The number of attempts is limited by this parameter.

Default Value:

8 - A maximum of 8 attempts are made to create a thread before reporting failure of thread creation.

Dependencies:

DelayBetweenAttempts

MaxIdleThreads

Maximum number of idle threads in thread pool.

Idle threads more than this number are automatically deleted. This ensures that the thread pool does not become too populated with idle threads.

Default Value:

4 - A Maximum of 4 threads can remain idle in the thread pool.

DefaultLogManager Configurations

Following are the parameters present in **Fiorano->etc->ClientLogManager->DefaultLogManager**

Logging in FioranoMQ is achieved with the help of Log4J, with which it is valid to enable logging at runtime without modifying the application binary.

LogDir

Base Directory where logs will be stored.

Allows the user to specify the directory in which the logs are to be stored. The user can change this if he wishes to have the logs stored in a more convenient place.

Default Value: ../profiles/FioranoMQ/run/logs/ (The directory where the logs are stored)

Encoding

The name of a supported character encoding.

Allows the user to specify any character encoding to use while saving the logs. This should be set before any logs have been written.

Default Value: UTF-8 (The default character encoding is UTF-8)

Level

Log level that is used by newly constructed handler objects.

This parameter allows the user to increase or decrease the log level. Increasing the log level would give a more detailed information on any errors or warnings that occur. Decreasing it would give briefer version of the same.

Valid Values:

-1 to 6, 10 (Any integer between 0 to 6, 10 and -1.0 being the lowest level of description and 10 being the highest).

Any log entry is logged at a particular log level. Each logger has associated a logging level, which can be anyone of the following values:

- -1 INHERIT - Inherits log level from its parent
- 0 QUIET - No logged information
- 1 FATAL - Severe errors that might cause premature termination
- 2 ERROR - Other runtime errors or unexpected conditions
- 3 WARN - Use of deprecated APIs, poor use of API, almost errors, other runtime situations that are undesirable or unexpected, but not necessarily wrong.
- 4 INFO - Interesting runtime events (startup/shutdown)

- 5 DEBUG - Detailed information on the flow through the system.
- 6 TRACE - More detailed information
- 10 ALL - Any logged information

Default Value: two (The default log level is 2)

FileSizeLimit

Maximum size of each log file. 0 => infinite. This is used by FileHandler.

This parameter allows the user to limit the size of the log file. In case of storage space restrictions, specifying a limit to the log size would prevent running out of memory.

Default Value: 0 - 0 means the file size has no limit.

FileCount

Maximum number of log files that should be created. This is used by FileHandler.

It allows the user to specify the number of log files to be created.

Default Value:

1 - By default only 1 log file is created.

AppendToFile

Specifies append mode for file handler. By default it is appended.

This parameter allows the user to specify whether or not to append data to log file. If set to false then, the entries to be added to the log file are added to a new file each time till the number of files reaches the FileCount (24). After that, the earliest log file is over written with the current log entries. The log files keep rotating. If set to true, the log entries are just appended to the end of the current log file.

Valid Values:

- Yes (in Studio)/True - Enables Appending to file.
- No (in Studio)/False - Disables Appending to file.

Default Value:

Yes (in Studio)/True - Enables Appending to file.

Formatter

Specifies the formatter. The class name of the class which is used to format the log files.

Default Value:

fiorano.jms.log2.def.CompactFormatter - The class used for formatting the log files.

MQ Default Object Creator Configurations

Following are the parameters present in **Fiorano->etc->MQDefaultObjCreator**

AutoUpdationAllowedForCFs

Sets the AutoUpdate property for MQ default ConnectionFactorys (CF) based on this flag. Sets a boolean in ConnectionFactory indicating whether updation are allowed in connection factories during startup.

Valid Values:

- Yes (in Stuido)/True – Enables AutoUpdation for default connection factories.
- No (in Studio)/False - Disables AutoUpdation for default connection factories.

Default Value:

Yes (In Studio)/True - Enables the AutoUpdate property for the MQ default CFs.

BackupServerIp

Specifies the backup server IP. This IP is configured in backup - URL for default connection factories.

This allows the user to set a backup server IP for the default connection factories.

Default Value:

No Value - This is left blank by default. The user can set the backup server IP by setting a value to this parameter.

Dependencies:

BackupServerPort

BackupServerPort

Specifies the backup server port. This port is configured in backup - URL for default connection factories.

This allows the user to set a backup server port for the default connection factories.

Default Value:

-1 - This is the default value but it is an invalid port number and it is hence ignored. To have a valid backupUrl both the BackupServerIP and the BackupServerPort should be valid.

Dependencies: BackupServerIp

Timer Service Configurations

Following are the parameters present in **Fiorano->etc->TimerService**

Name

The name of the timer service.

Default Value:

FioranoTimerService - The default name of the timer service.

ClockTick

Time period in milliseconds after which processes are checked for scheduling.

The time interval in milliseconds between successive ticks of the timer's clock. Increasing this value results in less frequent checks for scheduling of processes.

Default Value:

500 - The time in milliseconds after which the process are checked for scheduling.

JNDI Configurations

NativeFileNamingManager

Following are the parameters present in **Fiorano->jndi->NamingManager->File->NativeFileNamingManager**

DefragmentationLowerThreahold

Minimum number of deleted entries for triggering auto-defragmentation of AdminObjectManager

It is necessary to remove the deleted entries (entries that are marked as deleted entries) from the cache to prevent the cache from becoming too large in size. However the cache compaction should not occur for every deleted entry as the process is quite costly. The minimum number of entries that should have been deleted to trigger the defragmentation is set by this parameter. The user could increase or decrease the value depending upon his needs.

Default Value:

10 - The defragmentation is triggered only if there are a minimum of ten deleted entries.

Dependencies:

DefragmentationUpperThreshold, DefragmentationPercentage

DefragmentationUpperThreshold

The maximum number of deleted entries after which auto-defragmentation of AdminObjectManager is guaranteed during start up.

When the number of deleted entries in the cache becomes more than this parameter, the triggering of the defragmentation process is guaranteed. This value should be greater than the DefragmentationLowerThreshold (40). However having a very high value could result in a waste of memory.

Default Value:

15 - The Defragmentation is guaranteed only if there are 15 deleted entries.

DefragmentationLowerThreshold, DefragmentationPercentage

DefragmentationPercentage

The threshold percentage for auto-defragmentation during startup if the number of deleted entries fall between the above two thresholds.

When the number of deleted entries becomes more than the DefragmentationLowerThreshold (40), the defragmentation is not triggered immediately. It is triggered either if the number of deleted entries is more than the DefragmentationUpperLimit (41) too or if the percentage of deleted entries present in the cache becomes more than the value set in this parameter.

Default Value:

20 - The percentage of deleted entries present in the cache should be more than 20% to trigger the defragmentation.

Path

Root path for this Naming Manager.

Default Value:

The current directory is the default root path for the NativeFileNamingManager

CacheNamingManager

Following are the parameters present in **Fiorano->jndi->CacheNamingManager**

TTL

Duration in milliseconds for which admin object remains available in memory.

This parameter allows the user to determine how long the admin object entries will remain in the cache. If the user wants it to remain for a long time this value should be increased. In case the data are needed only for a very short time this value can be reduced.

Default Value:

60000 - The time in milliseconds for which the admin objects remain in memory (60 Seconds).

XMLFileNamingManager

Following are the parameters present in **Fiorano->jndi->XMLFileNamingManager**

Path

Absolute/relative path of the xml file used for persisting information about admin objects.

Default Value - Current directory

Filename

ServiceProvider instance part of the profile.

Default Value:

admin.xml - The default xml filename

LDAPNamingManager

Fiorano->jndi->LDAPNamingManager

LdapProviderUrl

URL of LDAP Server from where information related to admin objects are read.

This parameter allows the user to change the LDAP provider URL.

Default Value:

ldap://164.164.128.11:389/ - The default URL of the LDAP Server

LdapBasedn

Base DomainName in the LDAP Server under which all information related to admin objects are stored.

This parameter allows the user to set the Base Domain from which to retrieve the information related to the admin objects.

Default Value:

o=fiorano,c=US - The default Base DomainName in the LDAP Server.

Principal

The UserName using which NamingManager connects to LDAP Server for accessing/storing information related to Admin objects.

Allows the user to set the authentication details to connect to the LDAP Server.

Default Value:

o=fiorano,c=US - The default username with which to connect to the LDAP Server

Credentials

Password using which user with specified login connects to the specified LDAP Server.

LdapInitialCtxFactory

The Class name of the initial context factory which is used to open connection with the specified LDAP Server.

Default Value:

com.sun.jndi.ldap.LdapCtxFactory - The default initial context factory class used for connecting to the LDAP Server

LdapPollInterval

TimeInterval after which Naming Manager tries to reconnect to LDAP Server in case of a connection breakup.

This allows the user to set the time interval after which an attempt to connect to the LDAP Server should be made in case of a connection breakup. This value coupled with the LdapPrimaryServerReconnectAttempts (64) determines the total amount of time after which no more attempts are made to connect to the primary server.

Default Value:

4000 - The default time interval between successive attempts to connect to the LDAP Server.

Dependencies:

LdapPrimaryServerReconnectAttempts

LdapPrimaryServerReconnectAttempts

Number of reconnect attempts to primary server that Naming Manager makes before trying to connect to secondary server.

This allows the user to decide on the number of attempts made trying to connect to the primary server before connecting to the secondary server. This value coupled with the LdapPollInterval (63) determines the total amount of time after which no more attempts are made to connect to the primary server.

Default Value: 5

Dependencies: LdapPollInterval

BackupLdapProviderUrl

The URL of backup LDAP Server which Naming Manager tries to connect if primary LDAP Server is not available.

This parameter allows the user to setup the back up LDAP Server to which the Naming Manager can connect to when the primary LDAP Server is down. After trying to connect to the primary server LdapPrimaryServerReconnectAttempts (64) number of times, an attempt to establish connection with the server running at the URL specified by this parameter is made.

Dependencies: LdapPrimaryServerReconnectAttempts

BackupLdapBasedn

Base DomainName in the backup LDAP Server under which all information related to admin objects are stored.

This parameter allows the user to set the Base Domain from which to retrieve the information related to the admin objects.

Default Value:

o=fiorano, c=US - The default base domain name in the backup LDAP Server

BackupPrincipal

UserName using which NamingManager connects to backup LDAP Server for accessing/storing information related to Admin objects.

Allows the user to set the authentication details to connect to the backup LDAP Server.

Default Value:

o=fiorano, c=US - The default username used to connect to the backup LDAP Server

BackupCredentials

Password - using which user with specified login connects to the specified backup LDAP Server.

BackupLdapInitialCtxFactory

Class name of the initial context factory which is used to open connection with the specified backup LDAP Server.

Default Value:

com.sun.jndi.ldap.LdapCtxFactory - The default initial context factory class used for connecting to the backup LDAP Server

BackupLdapPollInterval

The TimeInterval after which Naming Manager tries to reconnect to backup LDAP Server in case of a connection breakup.

This allows the user to set the time interval after which an attempt to connect to the backup LDAP Server should be made in case of a connection breakup.

Default Value:

4000 - The default time interval between successive attempts to connect to the backup LDAP Server.

TopicCFBaseDn

Base Domain name in LDAP Server under which all topic connection factories will be stored.

Allows the user to set the domain under which the Topic connection factories are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the topic connection factories are stored.

QueueCFBaseDn

Base Domain name in LDAP Server under which all queue connection factories will be stored.

Allows the user to set the domain under which the queue connection factories are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the queue connection factories are stored.

AdminCFBaseDn

Base Domain name in LDAP Server under which all admin connection factories will be stored.

Allows the user to set the domain under which the Admin connection factories are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the admin connection factories are stored.

UnifiedCFBaseDn

Base Domain name in LDAP Server under which all unified connection factories will be stored.

Allows the user to set the domain under which the unified connection factories are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the unified connection factories are stored

TopicsBaseDn

Base Domain name in LDAP Server under which all topics will be stored.

Allows the user to set the domain under which all Topics are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the topics are stored.

QueueBaseDn

Base Domain name in LDAP Server under which all queues will be stored

Allows the user to set the domain under which all Queues are stored.

Default Value:

o=fiorano, c=US - The default domain name under which the queues are stored.

RDBMSNamingManager

Following are the parameters present in **Fiorano->jndi->RDBMSNamingManager**

URL

URL of database server from where information related to admin objects are read.

Allows the user to set/change the URL of the database server.

Default Value:

jdbc:hsqldb:Database - The default URL of the database server.

Password

Password using which user with specified login connects to the specified database server.

Username

UserName using which NamingManager connects to RDBMS for accessing/storing information related to Admin objects.

DatabaseDriver

Database driver class name which is used to open connection with the specified RDBMS server.

Default Value:

org.hsqldb.jdbcDriver - The default database driver class name

SQLStatementFile

Name of the properties file that is used to define name-value properties specific to a particular database.

Allows the user to set/change the properties file.

Default Value:

nm_sqls.properties - The default properties file used.

ConnectionLostErrorMessage

Error message to be posted in case of lost connection.

Allows the user to set the custom messages to be displayed when connection to the RDBMS Server is lost.

Default Value:

DB_CONNECTION_LOST - The default connection lost message.

EnableReconnect

Specifies if Naming Manager has to try to reconnect to database server in case of a connection breakup.

Allows the user to decide whether the Naming Manager has to make an attempt to reconnect to the RDBMS Server when the connection is lost. In cases where is reconnect is not necessary or should not be performed this parameter should be set to False.

Valid Values:

- True - Enables reconnect attempts to database server when the connection is down.
- False - Disables reconnect attempts to database server when the connection is down.

Default Value:

True - Enables reconnect attempts to database server when the connection is down.

DatabaseConnectionTimeout

Time till which Naming Manager will keep trying to connect to the database server for accessing/storing information related to admin objects.

The time for which the Naming Manager will keep trying to connect to the database server when the connection is lost. This can be increased or decreased depending on the scenario.

Default Value:

10000 - The time in milliseconds till which the Naming Manager will keep trying to connect to the database server.

InitializeDB

Specifies whether to initialize DB or not for accessing/storing information related to admin objects.

Valid values:

- True - Enables initializing the DB
- False - Disables initializing the DB

Default Value:

True - Enables initializing the DB.

Connection Consumer Configuration

FileDBManager5

Following are the parameters present in **Fiorano->mq->connection-consumer->FileDBManager->FileDBManager5**

Path

The configuration related to the consumer is stored in the file based data storage in the path specified here.

Valid values:

Default value is **CONN_CONSUMER**

The path specified is relative to the run directory of the respective FioranoMQ profile.

DeletedThresholdPercent

A cache stores and manages the list of all the tables pertaining to consumer related information in this file-storage based table. After repeated modification of the table, old entries in the table are marked deleted. This value provides a threshold percentage of deleted entries after which the Cache Compaction Procress is started, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with DeletedThresholdCount (No. 3) ensures it is not invoked too often.

Valid values:

Default value is **50**, which means if the ratio of deleted entries to total entries is equal to or greater than 0.5 then the Cache compaction process is started.

Range of int values in java. (-2^32 to 2^32 -1)

Note:

- All values less than or equal to zero are as good as each other.
- All values greater than or equal to 100 are as good as each other.

Example

Use a factor that best defines the ratio of deleted entries/total entries in the table. If there is good chance that many entries are marked deleted quite often then choose a higher number.

Dependencies

DeletedThresholdCount

DeletedThresholdCount

A cache stores and manages the list of all the tables pertaining to consumer related information in this file-storage based table. After repeated modification of the table, old entries in the table are marked deleted. This value provides a threshold count of deleted entries for starting the Cache Compaction Process, at the end of which the table does not have any deleted entries. Since cache compaction involves CPU intensive File operations, this flag along with DeletedThresholdPercent (No. 2) ensures it is not invoked too often.

Valid values:

Range of int values in java. (-2^32 to 2^32 -1). Default value is **10**

All values less than or equal to zero are as good as each other.

Example

Although the **DeletedThresholdPercent** provides a ratio based method of ensuring that the cache compaction process does not occur too often, it does not provide an absolute lower limit on the number of deleted entries that triggers a cache compaction. Choose a value that you feel is high enough to trigger a cache compaction.

Dependencies

DeletedThresholdPercent

DbTableNamePrefix

DB table names are usually prefixed with a string that can be modified by this property.

Valid values:

Any String. Default value is **#**

Example

If you want to save the present state of the DB table and are considering using the saved state sometime in the future, then modify the prefix to some other relevant string and change it back to the original string when you want to retrieve the saved state of the DB table.

Directory Service

Following are the parameters present in **Fiorano->etc->DirectoryService.**

EnableCluster

Sets whether cluster manager should be started or not with FioranoMQ Server.

Default Value:

Yes

EnableProfileMonitoring

This parameter will define whether to monitor the configurational changes made to the profiles residing inside the Management Server.

If enabled, all the configurational changes will be logged automatically to the file $DBPATH/logs/profile-monitor.log

Default Value: no

DirectoryServiceAddress

Sets the IPAddress on which cluster manager will be running.

Default Value: localhost

DirectoryServicePort

Sets the port on which cluster manager must be running.

Default Value: 10389

SecurityPrincipal

Sets the securityPrincipal required for creating a connection to cluster manager.

Default Value: Uid=admin,ou=system

SecurityCredentials

Sets the securityCredentials required for creating a connection to cluster manager.

Default Value: secret

AuthenticationMode

Sets the authentication mode in which connection will be created to cluster manager.

Default Value: simple

InstancePath

Sets the path of the Instance on which cluster manager will be running.

If set to null, the default value set for the working directory is $FIORANO_HOME/DSInstance/$PROFILE_NAME.

Default Value: null

SynchPeriodMillis

Sets the synchronization time for the Directory Service in milliseconds.

Default Value: 15000