



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

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

FIORANO END-USER LICENSE AGREEMENT

This Fiorano end-user license agreement (the Agreement) is a legal agreement between you (hereinafter Customer), either an individual or a corporate entity, and Fiorano Software, Inc., having a place of business at 718 University Ave, Suite 212 Los Gatos, CA 95032, USA, or its affiliated companies (hereinafter Fiorano) for certain software developed and marketed by Fiorano as defined in greater detail below. By opening this package, installing, copying, downloading, extracting and/or otherwise using the software, you are consenting to be bound by and are becoming party to this agreement on the date of installation, copying, download or extraction of the software (the Effective Date). If you do not agree with any of the terms of this Agreement, please stop installing and/or using the software and promptly return the unused software to the place of purchase. By default, the Software is made available to Customers in online, downloadable form. The terms of this Agreement shall apply to each Software license granted by Fiorano under this Agreement.

1. DEFINITIONS.

- a. **Affiliate** means, in relation to Fiorano, another person firm or company which directly or indirectly controls, is controlled by or is under common control with Fiorano and the expression 'control' shall mean the power to direct or cause the direction of the general management and policies of the person firm or company in question.
- b. **Commencement Date** means the date on which Fiorano delivers the Software to Customer, or if no delivery is necessary, the Effective Date set forth in this Agreement or on the relevant Order Form.
- c. **Designated Center** means the computer hardware, operating system, customer-specific application and Customer Geographic Location at which the Software is deployed as designated on the corresponding Order Form.
- d. **Designated Contact** shall mean the contact person or group designated by Customer and agreed to by Fiorano who will coordinate all Support requests to Fiorano.
- e. **Documentation** means the user guides and manuals for installation and use of the Software. Documentation is provided in CD-ROM or bound form, whichever is generally available.
- f. **Error** shall mean a reproducible defect in the Supported Program or Documentation when operated on a Supported Environment which causes the Supported Program not to operate substantially in accordance with the Documentation.
- g. **Excluded Components** shall mean such components as are listed in Exhibit B. Such Excluded Components do not constitute Software under this Agreement and are third party components supplied subject to the corresponding license agreements specified in Exhibit B.
- h. **Excluded License** shall mean and include any license that requires any portion of any materials or software supplied under such license to be disclosed or made available to any party either in source code or object code form. In particular, all versions and derivatives of the GNU GPL and LGPL shall be considered Excluded Licenses for the purposes of this Agreement.
- i. **Resolution** shall mean a modification or workaround to the Supported Program and/or Documentation and/or other information provided by Fiorano to Customer intended to resolve an Error.
- j. **Residuals** shall mean information in non-tangible form which may be retained by persons who have had access to the Confidential Information, including ideas, concepts, know-how or techniques contained therein.
- k. **Order Form** means the document in hard copy form by which Customer orders Software licenses and services, and which is agreed to in writing by the parties. The Order Form shall reference the Effective Date and be governed by the terms of this Agreement. Customer understands that any document in the nature of a purchase order originating from Customer shall not constitute a contractual offer and that the terms thereof shall not govern

any contract to be entered into between Fiorano and Customer. The Order Form herein shall constitute an offer to purchase made by the Customer under the terms of the said Order Form and this Agreement.

- l. **Software** means each of the individual Products, as further outlined in Exhibit-A, in object code form distributed by Fiorano for which Customer is granted a license pursuant to this Agreement, and the media, Documentation and any Updates thereto.
- m. **Support** shall mean ongoing support provided by Fiorano pursuant to the terms of this Agreement and Fiorano's current support policies. **Supported Program or Supported Software** shall mean the then current version of the Software in use at the Designated Center for which the Customer has paid the then-current support fee (**Support Fee**).
- n. **Support Hours** shall mean 9 AM to 5 PM, Pacific Standard Time, Monday through Friday, for Standard Support.
- o. **Support Period** shall mean the period during which Customer is entitled to receive Support on a particular Supported Program, which shall be a period of twelve (12) months beginning from the Commencement Date, or if applicable, twelve (12) months from the expiration of the preceding Support Period. Should Fiorano withdraw support pursuant to section 1 (q), the Support Period shall be automatically reduced to the expiration date of the appropriate Software.
- p. **Supported Environment** shall mean any hardware and operating system platform which Fiorano provides Support for use with the Supported Program.
- q. **Update** means a subsequent release of the Software that Fiorano generally makes available for Supported Software licensees at no additional license fee other than shipping and handling charges. Update shall not include any release, option, feature or future product that Fiorano licenses separately. Fiorano will provide Updates for the Supported Programs as and when developed for general release in Fiorano's sole discretion. Fiorano may withdraw support for any particular version of the Software, including without limitation the most current Update and any preceding release with a notice of three (3) months to Customer.

2. SOFTWARE LICENSE.

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

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

- (1) to use the Software solely for Customer's operations at the Designated Center consistent with the use limitations specified or referenced in this Agreement, the Documentation for such Software or any Order Form accepted by Fiorano pursuant to this Agreement. Customer may not relicense, rent or lease the Software or use the Software for third party training, commercial timesharing or service bureau use;
- (2) to use the Documentation provided with the Software in support of Customer's authorized use of the Software;
- (3) to make a single copy for back-up or archival purposes and/or temporarily transfer the Software in the event of a computer malfunction. All titles, trademarks and copyright or other restricted rights notices shall be reproduced in any such copies;
- (4) to allow third parties to use the Software for Customer's operations, so long as Customer ensures that use of the Software is in accordance with the terms of this Agreement.

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

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

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

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

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

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

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

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

(d) Customer Specific Objects.

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

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

(e) Additional Restrictions on Use of Source Code.

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

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

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

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

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

(f) Acceptance tested Software

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

3. TECHNICAL SERVICES.

(a) **Maintenance and Support Services.** Maintenance and Support services 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 THE PRODUCT, USING A TWELVE (12) MONTH STRAIGHT-LINE AMORTIZATION SCHEDULE STARTING ON THE COMMENCEMENT DATE. FURTHER, IF SUCH DAMAGES RESULT FROM CUSTOMER'S USE OF THE SOFTWARE OR SERVICES, SUCH LIABILITY SHALL BE LIMITED TO THE PRORATED AMOUNT OF FEES PAID FOR THE RELEVANT SOFTWARE OR SERVICES GIVING RISE TO THE LIABILITY TILL THE DATE WHEN SUCH LIABILITY AROSE, USING A TWELVE (12) MONTH STRAIGHT-LINE AMORTIZATION SCHEDULE STARTING ON THE COMMENCEMENT DATE. NOTWITHSTANDING ANYTHING IN THIS AGREEMENT, THE FOREGOING LIMITATIONS, EXCLUSIONS AND DISCLAIMERS SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.

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

7. PAYMENT PROVISIONS.

(a) **Invoicing.** All fees shall be due and payable thirty (30) days from receipt of an invoice and shall be made without deductions based on any taxes or withholdings. Any amounts not paid within thirty (30) days 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 code of Products, and (ii) the business and technical information of either party, including but not limited to any information relating to product plans, designs, costs, product prices and names, finances, marketing plans, business opportunities, personnel, research, development or know-how;

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

(c) Use and Disclosure Restrictions. During the term of this Agreement, each party shall refrain from using the other party's Confidential Information except as specifically permitted herein, and from disclosing such Confidential Information to any third party except to its employees and consultants as is reasonably required in connection with the exercise of its rights and obligations under this Agreement (and only subject to binding use and disclosure restrictions at least as protective as those set forth herein executed in writing by such employees).

(d) Continuing Obligation. The confidentiality obligation described in this section shall survive for three (3) years following any termination of this Agreement. Notwithstanding the foregoing, Fiorano shall have the right to disclose Customer's Confidential Information to the extent that it is required to be disclosed pursuant to any statutory or regulatory provision or court order, provided that Fiorano provides notice thereof to Customer, together with the statutory or regulatory provision, or court order, on which such disclosure is based, as soon as practicable prior to such disclosure so that Customer has the opportunity to obtain a protective order or take other protective measures as it may deem necessary with respect to such information.

9. MISCELLANEOUS.

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

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

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

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

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

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

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

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

(i) Controlling Law and Jurisdiction. If this Software has been acquired in the United States, this Agreement shall be governed in all respects by the laws of the United States of America and the State of California as such laws are applied to agreements entered into and to be performed entirely within California between California residents. All disputes arising under this Agreement may be brought in Superior Court of the State of California in Santa Clara County or the United States District Court for the Northern District of California as permitted by law. If this Software has been acquired in any other jurisdiction, the laws of the 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.

Copyright © 1999-2007, Fiorano Software Technologies Pvt. Ltd. All rights reserved.

Copyright © 2008-2011, Fiorano Software Pte. Ltd. All rights reserved.

This software is the confidential and proprietary information of Fiorano Software ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement enclosed with this product or entered into with Fiorano.

Contents

Chapter 1: Connection Factory Configuration 32

AddShutdownHook	32
AllowDurableConnections.....	32
AutoDispatchConnections	33
Backup ConnectURLs	33
BatchTimeoutInterval.....	34
BatchingEnabled	34
BlockPublisherRetryInterval	34
ClientProxyURL	34
CompressionManager	35
Connection ClientID	35
CreateLocalSocket	35
Disable Ping	35
DisableSendingCSPStoredMessages.....	36
DurableConnectionReconnectInterval	36
DurableConnectionsBaseDir	37
EnableAutoRevalidation	37
UseFioranoCbr	37
EnableLMS	38
HTTPProxyURL	38
UseLocalProcedureCalls	38
LookupPreferredServer.....	38
MaxAdminConnectionReconnectAttempts	39
MaxBlockPublisherRetryAttempts	39
MaxDurableConnectionReconnectAttempts	39
MaxSocketCreationTries	40
ProxyAuthenticationRealm	40
ProxyCredentials	40
ProxyPrincipal	40
ProxyType	40
PublishBehaviourInAutoRevalidation	41
PublishWaitDuringCSPSyncup.....	41
SecurityManager	42
ServerProxyURL	42
SleepBetweenSocketCreationTries	42
SocketKeepAliveEnabled.....	42
SocketTimeout	43
SocksProxyURL	43
TCPBatchSize.....	43
TransportProtocol	43

java.naming.security.protocol	43
UpdateConnectURL	44
UseSingleSocketForSendReceive	44
XASocketTimeout	44
isThreadContextClassLoaderUsed	44

Chapter 2: Common FioranoMQ Configuration 45

FMQConfigLoader	45
LazyRSCreation	45
AllowDurableConnections	45
UseSingleSocket	46
UseSingleSocketForAdmin	46
UseFioranoCbr	47
CreateDefaultACL	47
AclBasedDestinationSecurity	48
AllPermissions	48
AllowOnTheFlyAclCheck	49
AllowOnTheFlyCreationOfDestinations	49
EnsureUniquenessOfClientId	50
MaxTransactionBufferSize	50
EnableAutoRevalidation	50
HttpPollingInterval	51
IncludeMessageID	51
ResumeTimeoutInterval	51
MessageSelectorFactoryClassName	51
SSLEnabled	52
PingEnabled	52
SystemEncodingFormat	52

Chapter 3: Queue Subsystem Level Configurations..... 53

UnAckedQueueBufferSize	53
MaxNumberOfQueues	53
RedeliveryTriesOnListenerException	53
DelayInMsgDeliveryOnListenerException	54
CleanupDmqAtStartup	54
EnableSnooperOnAllQueues	55
EnableDMQOnAllQueues	55
PrefetchCount	55
PrefetchThreshold	56
DefaultStorageTypeForQueues	56
MaxPrefetchSize	56
OverflowFromBottomEnabled	57
DMQExpiryTime	57

EnableNotificationOnDeadMessage	57
EnableMessageMonitoring	58
FilePath	58
TimeDuration	58
CleanupInterval	58
InMemoryBlockingInterval	59
InMemoryBufferSize	59
PersistentInMemoryBufferSize	59
MaxPushAttempts	59
WaitBetweenAttempts	60
IsDbCleanupEnabled	60
UseInMemStatusCache	60
LoadBitSetOnStartup	60
EnableGMSQueueExpose	61
Exclusive	61
JumpDeletedFiles	61
DepthMonitoringEnabled	61
DepthMonitoringLevels	62

Chapter 4: Topic Subsystem Level Configuration..... 63

DupsOkBatchSize	63
RedeliveryTriesOnListenerException	64
DelayInMsgDeliveryOnListenerException	64
PublishWait	64
PublishWaitTimeRecheckUnits	65
MaxNumberOfTopics	65
EnablePublisherSlowdown	66
DropOldestMessage	66
DefaultStorageTypeForTopics	66
StoreMessageToDisk	67
EnableSnooperOnAllTopics	67
EventTopicName	67
IsForceFullyKillIDSubscriber	68
MaxPersistentStoreSize	68
UseMessagePersisterForDurableSubscriber	68
MaxMsgInOneRecv	68
BasicAllowedSize	69
UseOptimizedTCPReceive	69
EnableMessageMonitoring	69
FilePath	69
TimeDuration	70
PurgeLevelIndex	70
MinPurgeDifference	70
AllowDeletionOfSubtopics	71
NPMessagePublishTimeout	71

PublishBackoffThreshold	71
IgnorePSQSizeForPersistentMessages	72
EnableSessionStateChangeNotification	72

Chapter 5: Connection Manager Configurations..... 73

Default	73
CFNameAppender (Not being used in FioranoMQ)	73
ReaderCacheDisabled.....	73
TCPWindowSize (Not being Used in FioranoMQ)	74
HandshakeInWorkerThread	74
MagicTimeout	74
Port.....	74
Protocol	75
UseNagle	75
MonitoringRequest.....	75
MonitoringResponse.....	76
EnforceJsseAuthentication	76
MaxClientConnectionsCount	76
AdminConnection (Not being used)	77
Path	77
ServerAddress	77
HandShakingDisabled	77
ManagerClassName	78
FioranoMQServer.....	78
UseForPeerToPeerCommunication (Not being used in FioranoMQ)	79
UseSystemPropsForSSL	79
CipherSuiteList	79

Chapter 6: Dash Board Configurations..... 80

JettyServer.....	80
MaxThreads.....	80
EnableStart	80
MinThreads.....	81
LowThreads	81
MaxIdleTime	81
LowResourceMaxIdleTime.....	81
PortNumber.....	82

Chapter 7: Security Configuration 83

NativeFileBasedAclManager	83
LdapBasedAclManager	83
LdapProviderUrl.....	83

LdapProviderDn.....	83
Principal.....	84
Credentials.....	84
LdapInitialCtxFactory.....	84
LdapAclContextDn.....	84
LdapSecurityAuthentication.....	85
LdapPollInterval.....	85
LdapPrimaryServerReconnectAttempts.....	85
BackupLdapProviderUrl.....	85
BackupLdapProviderDn.....	85
BackupPrincipal.....	86
BackupCredentials.....	86
BackupLdapInitialCtxFactory.....	86
BackupLdapSecurityAuthentication.....	86
AclEntryObjectClasses.....	86
MaxAcePerAcl.....	86
XMLFileBasedAclManager.....	87
Path.....	87
Filename.....	87
MaxAcePerAcl.....	87
RdbmsBasedAclManager.....	87
MaxAcePerAcl.....	87
Url.....	88
Password.....	88
Username.....	88
DBDriver.....	88
ConnectionTimeoutInterval.....	88
ReconnectTimeInterval.....	88
EnableReconnect.....	88
StatementFile.....	89
ConnectionLostErrorMessage.....	89
InitializeDB.....	89
FileDBManager3.....	89
Path.....	89
DeletedThresholdPercent.....	90
DeletedThresholdCount.....	90
DbTableNamePrefix.....	91
NativeFilePrincipalManager.....	91
LdapPrincipalManager.....	91
LdapProviderUrl.....	91
LdapProviderDn.....	92
Principal.....	92
Credentials.....	92
LdapInitialCtxFactory.....	92
LdapSecurityAuthentication.....	93
LdapPollInterval.....	93

LdapPrimaryServerReconnectAttempts	93
BackupLdapProviderUrl.....	93
BackupLdapProviderDn.....	93
BackupPrincipal.....	94
BackupCredentials	94
BackupLdapInitialCtxFactory	94
BackupLdapSecurityAuthentication.....	94
LdapUserClassTop	94
LdapUserClassPerson	95
LdapUserClassOrganizationalPerson	95
LdapUserClassInetorgPerson	95
LdapGroupClassTop	95
LdapGroupClassUniqueNames	95
LdapGroupClassUniqueMember.....	96
LdapUserNameAttribute.....	96
LdapUserPasswordAttribute.....	96
LdapGroupNameAttribute	96
LdapGroupUserNameAttribute	96
LdapUserDn.....	97
LdapGroupDn	97
MaxPasswordLength	97
MaxMemberCount	97
XMLFilePrincipalManager.....	98
Path.....	98
UserFilename	98
GroupFilename	98
PrincipalManagerRDBMS	98
Url.....	98
Password	98
Username	98
DBDriver	99
ConnectionTimeoutInterval.....	99
ReconnectTimeInterval	99
EnableReconnect	99
StatementFile.....	99
ConnectionLostErrorMessage	99
InitializeDB	100
FileDBManager2	100
Path.....	100
DeletedThresholdPercent.....	100
DeletedThresholdCount	101
DbTableNamePrefix.....	101

Chapter 8: Logger Configuration 103

Introduction	103
--------------------	-----

AppenderAdditive	106
LogLevel	106
LogAppender	107
AppenderName	107
AppenderType	107
LogPattern	107
ThresholdLevel	108
FilterPattern	108
PrintTarget	108
FileName	109
IsAppend	109
MaxBackupIndex	109
MaxFileSize	110
MaxFilterLevel	110
MinFilterLevel	110
LogAppender	111
AppenderName	111
AppenderType	111
LogPattern	111
ThresholdLevel	111
FilterPattern	112
PrintTarget	112
FileName	112
IsAppend	113
MaxBackupIndex	113
MaxFileSize	114
MaxFilterLevel	114
MinFilterLevel	114

Chapter 9: HA Replicated Configuration 115

Primary Server Configuration	115
FioranoHAConnectionManager	115
Port	115
NagleAlgo	115
SocketInitializationTimeout	116
RealmStubManager	116
MinServiceID	116
MaxServiceID	117
QGMSSubManager	117
MinServiceID	117
MaxServiceID	117
TGMSSubManager	118
MinServiceID	118
MaxServiceID	118
SyncRealmStubManager	118

MinServiceID	119
MaxServiceID	119
SyncNamingStubManager	119
MinServiceID	119
MaxServiceID	120
MinServiceID	120
MaxServiceID	120
SyncQueueStubManager	121
MinServiceID	121
MaxServiceID	121
RpStateStubManager	121
MinServiceID	121
MaxServiceID	122
FioranoRpRealmManager	122
AclZipPath	122
AclUnzipDir	122
PrincipalZipPath	123
PrincipalUnzipDir	123
SocketTimeout	123
FioranoRpNamingManager	124
ZipPath	124
UnzipDir	124
SocketTimeout	124
FioranoHAKRPCProvider	125
BackupHAPort	125
BackupHAIPAddress	125
SocketTimeout	125
SocketCreationTimeout	126
HaStateNotifBroadcaster	126
LoggerName	126
FioranoStatusPersister	126
Path	126
FioranoHAManager	127
Primary	127
GatewayListenerNotUpMesg	127
PingInterval	127
GatewayServerPort	128
SocketCreationTimeout	128
ServerName	128
GatewayServerIPAddress	128
MaxWaitTimeout	129
WaitTimeForBackupServer	129
MaxPingCount	129
FioranoFMQKernel	130
MQUnDeploymentLists	130
ExternalUnDeploymentLists	130

MQDeploymentLists	131
ExternalDeploymentLists	131
FioranoRpTopicManager	131
ZipPath	132
UnzipDir	132
SocketTimeout	132
FioranoRpQueueManager	132
ZipPath	133
UnzipDir	133
SocketTimeout	133
Secondary Server Configuration	134
FioranoHAConnectionManager	134
Port	134
NagleAlgo	134
SocketInitializationTimeout	134
SyncTopicStubManager	135
MinServiceID	135
MaxServiceID	135
RealmStubManager	135
MaxServiceID	136
QGMSSubManager	136
MinServiceID	136
MaxServiceID	136
TGMSSubManager	137
MinServiceID	137
MaxServiceID	137
MinServiceID	137
MaxServiceID	138
SyncNamingStubManager	138
MinServiceID	138
MaxServiceID	138
NamingStubManager	139
MinServiceID	139
MaxServiceID	139
SyncQueueStubManager	139
MinServiceID	139
MaxServiceID	140
RpStateStubManager	140
MinServiceID	140
MaxServiceID	140
FioranoRpRealmManager	140
AclZipPath	141
AclUnzipDir	141
PrincipalZipPath	141
PrincipalUnzipDir	142
SocketTimeout	142

FioranoRpNamingManager	142
ZipPath	142
UnzipDir	143
SocketTimeout	143
FioranoHAKRPCProvider	143
BackupHAPort	143
BackupHAIPAddress	144
SocketTimeout	144
SocketCreationTimeout	144
HaStateNotifBroadcaster	144
LoggerName	144
FioranoStatusPersister	145
Path	145
FioranoHAManager	145
Primary	145
GatewayListenerNotUpMesg	146
PingInterval	146
GatewayServerPort	146
SocketCreationTimeout	146
ServerName	147
GatewayServerIPAddress	147
MaxWaitTimeout	148
FioranoFMQKernel	148
MQUnDeploymentLists	148
ExternalUnDeploymentLists	149
MQDeploymentLists	149
ExternalDeploymentLists	150
FioranoRpTopicManager	150
ZipPath	150
UnzipDir	150
SocketTimeout	151
FioranoRpQueueManager	151
ZipPath	151
UnzipDir	152
SocketTimeout	152

Chapter 10: HA Shared Configuration 153

Primary Server Configuration	153
FioranoHAKRPCProvider	153
BackupHAPort	153
BackupHAIPAddress	153
SocketTimeout	153
SocketCreationTimeout	154
FioranoHAConnectionManager	154
Port	154

NagleAlgo.....	154
SocketInitializationTimeout	155
FioranoHAStateStubManager	155
MinServiceID	155
MaxServiceID	156
FioranoHAManager	156
Primary.....	156
GatewayListenerNotUpMesg	156
GatewayServerPort	157
PingInterval.....	157
SocketCreationTimeout	157
ServerName	158
GatewayServerIPAddress.....	158
FioranoFMQKernel.....	158
MQUnDeploymentLists	158
ExternalUnDeploymentLists.....	159
MQDeploymentLists	159
ExternalDeploymentLists	160
Secondary Server Configuration	160
FioranoHAKRPCProvider	160
BackupHAPort	160
BackupHAIPAddress.....	161
SocketTimeout	161
SocketCreationTimeout	161
FioranoHAConnectionManager.....	161
Port.....	161
NagleAlgo.....	162
SocketInitializationTimeout	162
FioranoHAStateStubManager	162
MinServiceID	163
MaxServiceID	163
FioranoHAManager	163
Primary.....	163
GatewayListenerNotUpMesg	164
GatewayServerPort	164
PingInterval.....	164
SocketCreationTimeout	165
ServerName	165
GatewayServerIPAddress.....	165
FioranoFMQKernel.....	166
MQUnDeploymentLists	166
ExternalUnDeploymentLists.....	166
MQDeploymentLists	167
ExternalDeploymentLists	168

Chapter 11: Dispatcher Configuration..... 170

FioranoJobManager2	170
MinimumJobWorkers	170
MaximumJobWorkers.....	170
Dispatcher.....	171
Name	171
PingTimeoutAttribute.....	171
InitialTimeoutAttribute.....	171

Chapter 12: Repeater Configuration 173

FioranoRepeaterManager	173
Name	173
DurableSubscriptionInfoFileName	173
PingInterval	173
AvoidLoopback	174
TransactionSize	174
TransactionTimeout.....	174
SourceServer	175
Name	175
ConnectionInfo	175
ProtocolType	175
JMXProtocol	175
JMXPort	175
ServerURL	176
UserName.....	176
Password	176
TopicConnectionFactory	176
ServerSecurityManager.....	177
ProxyURLAttribute.....	177
ServerSecurityManager parameter LinkTopicInfo.....	177
IsDurable.....	177
ConnectionMode	177
Type	178
ReplyOn	178
SourceTopicName	178
TargetTopicName.....	178
MessageSelector	179

Chapter 13: Bridge Configurations..... 180

ConnectionInfo	180
ServerType	180
ProviderUrl.....	180

InitialContextFactory	180
BridgeUser	181
BridgePassword	181
QCF	181
Protocol	181
SecurityPrincipal	181
SecurityCredentials	181
ServerSecurityManager	182
QueueInfo	182
MsgSelector	182
QueueName	182

Chapter 14: JMX Connector Configuration 183

JMS Based Connector Configuration	183
InterceptorClassName	183
SecurityProtocol.....	183
Protocol.....	184
SecurityManagerClass.....	184
QueueConnectionFactory.....	184
ServerPort	185
ServerAddress.....	185
Username	185
Password	185
ConnectQueueName	185
RequestQueueName	185
NotifQueueName.....	186
RMI Based Connector Configuration	186
RMIServerPort.....	186
InterceptorClassName	186
RmiPortExportObjects.....	186
Hostname	186
RmiServerSocketFactoryClassName	187
RmiClientSocketFactoryClassName	187
JMX Engine Configuration	187
SecurityProtocol.....	187
Protocol.....	187
SecurityManagerClass.....	188
QueueConnectionFactory.....	188
ServerPort	188
ServerAddress.....	188
Username	189
Password	189
ConnectQueueName	189
RequestQueueName	189
NotifQueueName.....	189

Monitoring Manager Configuration.....	189
ClockTickForMonitoring	190

Chapter 15: Advanced Configuration..... 191

Route Manager Configuration	191
MaxTopicBuffer	191
RdbmsManager Configurations	191
URL	191
JdbcDriver	192
Password	192
Username	192
MaxConnections.....	192
DefaultConnections	192
MaxNoOfRowsFetch	193
EnableReconnect	193
ReconnectTimeInterval	193
WaitTime	194
ConnectionTimeoutInterval.....	194
PropertiesFilename.....	194
DbConnectivityQuery	194
EnableRdbms	195
Resource Manager Configurations	195
TimeInterval	195
EnableMonitoringThread.....	196
EnableVerboseLogging	196
EnableThreadDumpInfo	197
PingManager Configurations.....	197
PingTimeout	197
JobManager Configurations	198
MinimumJobWorkers	198
MaximumJobWorkers.....	198
JobsPerWorker	198
ObjectManager Configurations.....	199
HashCodeReuseAlgoUsed.....	199
ThreadManager Configurations	199
DelayBetweenAttempts	199
MaximumAttempts	200
MaxIdleThreads	200
DefaultLogManager Configurations	200
LogDir	200
Encoding	201
Level.....	201
FileSizeLimit.....	202
FileCount	202
AppendToFile	202

Formatter	203
MQ Default Object Creator Configurations	203
AutoUpdationAllowedForCFs	203
BackupServerIp	203
BackupServerPort	204
Timer Service Configurations	204
Name	204
ClockTick	204
JNDI Configurations	205
NativeFileNamingManager	205
DefragmentationLowerThreshold	205
DefragmentationUpperThreshold	205
DefragmentationPercentage	206
Path	206
CacheNamingManager	206
TTL	206
XMLFileNamingManager	207
Path	207
Filename	207
LDAPNamingManager	207
LdapProviderUrl	207
LdapBasedn	207
Principal	208
Credentials	208
LdapInitialCtxFactory	208
LdapPollInterval	208
LdapPrimaryServerReconnectAttempts	209
BackupLdapProviderUrl	209
BackupLdapBasedn	209
BackupPrincipal	210
BackupCredentials	210
BackupLdapInitialCtxFactory	210
TopicCFBaseDn	210
QueueCFBaseDn	211
AdminCFBaseDn	211
UnifiedCFBaseDn	211
TopicsBaseDn	212
QueueBaseDn	212
RDBMSNamingManager	212
URL	212
Password	212
Username	213
DatabaseDriver	213
SQLStatementFile	213
ConnectionLostErrorMessage	213
EnableReconnect	213

DatabaseConnectionTimeout.....	214
InitializeDB	214
Connection Consumer Configuration.....	215
FileDBManager5.....	215
Path	215
DeletedThresholdPercent	215
DeletedThresholdCount	216
DbTableNamePrefix	216
Directory Service.....	217
EnableCluster	217
EnableProfileMonitoring.....	217
DirectoryServiceAddress	217
DirectoryServicePort.....	217
SecurityPrincipal	218
SecurityCredentials	218
AuthenticationMode.....	218
InstancePath.....	218
SynchPeriodMillis	218

Chapter 1: Connection Factory Configuration

This chapter explains the parameters that can be configured at the Connection Factory level. Each connection that is created to the FioranoMQ Server using the connection factory inherits the parameters that are configured for the connection factory.

For information on how to create and configure Connection Factories in FioranoMQ using Studio, refer to section **23.2 Working with Connection Factories** of *FioranoMQ Handbook*.

AddShutdownHook

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

Enabling this parameter ensures that the program exits normally when the 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 User interruption 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. This is the default value.
- **none** - uses the default value that is set in the Server.

Default Value:

No - disables shutdown hook.

By default the shut down hook is disabled in client applications.

AllowDurableConnections

AllowDurableConnections is the boolean value that is used to disable or enable durable connections for a particular client when the durable connections are enabled on the Server. When **AllowDurableConnections** is enabled, messages get stored into the Client Side Persistence (CSP) storage during the period that connection with the Server is down.

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

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

```
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 This is the default value.
- **no** - disables durable connections in the client.
- **none** – uses the default value that is 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 the Server and enabled in client this disables all durable connections. If durable connections are enabled in the Server, their effect is dependent on the client configuration.

AutoDispatchConnections

AutoDispatchConnections is the boolean value that is used to determine whether the dispatcher Server should be used for creating client connections. By default this flag is enabled. If set to true, the incoming client connections are automatically redirected to the least loaded Server that is registered to the dispatcher. If set to false, connections are created on the dispatcher without being redirected.

Valid values:

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

Dependencies:

This parameter affects only the Dispatcher Server.

Backup ConnectURLs

Backup ConnectURLs are URLs of the Backup Servers that are used in case the Primary Server is down. Multiple backup URLs may be used but need to be separated by the use of semicolons.

Example: Backup Connect URLs can be specified as:

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

BatchTimeoutInterval

BatchTimeoutInterval is the time interval after which a batch of non-persistent messages will be sent irrespective of the size of the 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. The default time is 1000 milliseconds.

BatchingEnabled

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. This is the default value.
- **false**: disables batching.

Default value: true

BlockPublisherRetryInterval

Defines the behaviour of publisher when auto-revalidation is enabled on the server and when PublishBehaviourInAutoRevalidation is set to Block. It signifies the time interval to sleep on each retry attempts i.e. the value specified in MaxBlockPublisherRetryAttempts.

Valid value:

Time in milliseconds. The default value is set to 1 second.

ClientProxyURL

ClientProxyURL parameter holds the configuration information of the client proxy provider for http tunneling.

Valid value:

String - value that contains information about the proxy provider.

Example: http://ClientProxyServer:80

CompressionManager

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

Valid value:

Class path of the implemented Compression Manager Set as the default value is:
fiorano.jms.services.msg.compression.def.CompressionManagerImpl

Connection ClientID

Connection ClientID is 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 the same clientID. Setting the ConnectionClientID will throw exceptions when more than one connection is created with the Connection Factory.

CreateLocalSocket

CreateLocalSocket parameter is used when both the Client and the Server share the same machine.

If set to true a socket is created when the client connects to the Server. If set to false no socket is created when local connections are made.

Valid values:

- **yes** - creates a local socket.
- **no** - does not create a local socket. This is the default value.

Default Value: **no**

Disable Ping

Disable Ping is the boolean value used to disable pingging for a particular client when pingging is enabled on the server.

If ping is enabled, the client will ping the Server for every **PING_INTERVAL** seconds which are, by default, set to 60 seconds. If a reply is not received from the Server, the client will re-validate the connection.

Valid values:

- **yes** - pinging is disabled on the client-side.
- **no** - pinging is enabled on the client-side. This is the default value.
- **none** - the default value is used.

DisableSendingCSPStoredMessages

DisableSendingCSPStoredMessages is the boolean value that is used to disable sending pending messages stored in the CSP to the Server.

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

Valid values:

- **true - yes** - disables sending messages stored in the CSP when the client ID is set.
- **False - no** - enables sending messages stored in the CSP when the client ID is not set.

Note: By default, **DisableSendingCSPStoredMessages** is set to **false** or **no**. Studio the possible values are **yes/no**. While setting using samples the values are **true/false**.

JNDI name

FioranoJNDIContext.DONT_SEND_PREVIOUSLY_STORED_MESSAGE

Example: env.put(FioranoJNDIContext.DONT_SEND_PREVIOUSLY_STORED_MESSAGES, "true");

Usage

This parameter is set when the messages stored in the CSP are NOT to be sent once a connection is re-established with the Server.

Example:

The Server is started with the durable connection enabled. A publisher and a subscriber are started. The publisher has the **DONT_SEND_PREVIOUSLY_STORED_MESSAGES** flag set to **true**. The server is shutdown. The publisher publishes a few messages and is shutdown as well. The server and the publisher are restarted. The subscriber does not receive the messages sent by the publisher while the Server was down.

DurableConnectionReconnectInterval

DurableConnectionsReconnectInterval is the time interval between two re-connection attempts.

When a client is disconnected from the Server with auto revalidation enabled, the client repeatedly tries to reconnect to the Server in the form of a loop. The time between each successive try is set using this flag.

Valid value:

This value is set in Milliseconds. The default time interval is 1000 milliseconds.

This parameter is used only when auto revalidation is enabled.

DurableConnectionsBaseDir

When durable connections are enabled and the Server goes down, client-side messages are stored on the local disk while the connection is revalidated. The DurableConnectionsBaseDir parameter specifies the name of the base directory where messages will be persisted.

Valid value:

Any string value that may be considered a directory name.

The default value is set to **cspCache**. Messages persist in the directory cspCache if no other variable is set.

EnableAutoRevalidation

EnableAutoRevalidation is the boolean value that is used to disable auto-revalidation for a particular client.

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

Valid value:

- **yes** - enables auto revalidation for the client. This is the default value.
- **no** - disables auto revalidation for the client.

The default value is set to **enabled**.

Revalidation occurs by the client only when auto revalidation is enabled on the Server. Revalidation by the client does not occur if auto revalidation is disabled on the Server and enabled in the Connection Factory parameter.

The Server value takes precedence over the Connection Factory parameter or the parameter set by the client.

UseFioranoCbr

The UseFioranoCbr parameter enables/disables Content Based Routing (CBR). CBR allows XPath based selection on messages with XML content. For more information please refer to chapter 18: FioranoMQ Content Based Routing available in the FioranoMQ Concept Guide.

Valid value:

- **yes** - enables Fiorano CBR.
- **no** - disables Fiorano CBR.

EnableLMS

EnableLMS is the boolean value that is used to enable/disable Large Message Support (LMS).

Valid values:

- **yes** - enables LMS on the client-side. This is the default value.
- **no** - disables LMS on the client-side.

Dependencies:

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

HTTPProxyURL

The HTTPProxyURL parameter holds the configuration information of the HTTP proxy provider.

Valid value:

A String value containing information about the proxy provider:

Example: http://ProxyServer:80

UseLocalProcedureCalls

UseLocalProcedureCalls is the boolean value used to enable/disable Local Procedure Calls (LPC). If this parameter is enabled then clients within the server using the Connection Factory will connect to the Server directly, without making any socket connections.

Valid values:

- **yes** - enables LPC.
- **no** - disables LPC. This is the default value. .

By default, LPC is **disabled**.

LookupPreferredServer

LookupPreferredServer is the boolean value that is used to instruct the client to connect to the preferred Server. When this parameter is enabled, clients connecting to the Dispatcher are redirected to the Server that is set as the preferred Server. If this parameter is disabled, clients will be connected to the least loaded Server.

Valid values:

- **yes** - enables Lookup Preferred Server
- **no** - disables Lookup Preferred Server

Dependencies:

This parameter is used only by a Dispatcher Server.

MaxAdminConnectionReconnectAttempts

The MaxAdminConnectionReconnectAttempts are the maximum number of reconnect attempts that will be made by the Admin Connection to connect to the Server when the Server is down.

Valid value:

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

Note:

- If a positive value is defined, the Admin Connection tries to reconnect that number of times.
- If a connection is not made within attempts defined, the attempted revalidations are terminated and an exception is thrown.

MaxBlockPublisherRetryAttempts

Defines the behaviour of publisher when auto-revalidation is enabled on the server and when PublishBehaviourInAutoRevalidation is set to Block. It signifies the number of retry attempts to make when server is in transition state i.e STANDALONE state and then to ACTIVE TRANSITION to ACTIVE.

Valid value:

Any positive integer value. Default value is 10.

MaxDurableConnectionReconnectAttempts

MaxDurableConnectionReconnectAttempts are the maximum number of reconnect attempts that the client will make if it is unable to connect to the Server when the Server is down.

Valid value:

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

Note:

- This flag will be used only if durable connections are enabled on the Server.
- If a positive value is defined, the connection tries to reconnect that number of times.

- If a connection is not made within attempts defined, the attempted revalidations are terminated and an exception is thrown.

MaxSocketCreationTries

MaxSocketCreationTries are the maximum number of attempts that the client will make to create a socket connection on a Server when it is unable to connect to the Server. If it fails to connect within the specified number of attempts an exception is thrown on the client-side.

Valid value:

Any positive integer value. By default, this value is 1.

ProxyAuthenticationRealm

ProxyAuthenticationRealm holds the address of the Authentication Realm being used on the proxy Server through which HTTP based connections created using the ConnectionFactory are routed.

Valid value:

String representing the address of the proxy Server on which the Authentication Realm exists.

Dependencies:

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

ProxyCredentials

ProxyCredentials is the password of the proxy principal which is used when the proxy Server is used.

ProxyPrincipal

ProxyPrincipal is the name of the proxy principal used when the proxy is used.

ProxyType

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

Valid values:

- MS_ISA_PROXY
- WIN_PROXY
- NETSCAPE_PROXY
- WINGATE_PROXY

- DEFAULT_PROXY

PublishBehaviourInAutoRevalidation

PublishBehaviourInAutoRevalidation parameter defines the behavior of the publisher when the auto revalidation is enabled and the durable connections are disabled in the Server. A new publish call will either throw an exception, or block or ignore a message. By default the value of this parameter is - **throw exception**.

Valid values:

- **0 – Exception** - Throws an exception upon publishing messages. This happens when the Server is down, auto revalidation is enabled but durable connections are disabled.
- **1 - Ignore** - Ignores further publish message calls. All messages sent when the server is not up will be lost. (The same conditions as above.)
- **2 – Block** - Blocks further messages from being published until auto revalidation is performed. (The same conditions as above.)

JNDI name:

FioranoJNDIContext.PUBLISH_BEHAVIOUR_IN_AUTO_REVALIDATION

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 auto revalidation undertaken when durable connections disabled. The User may either choose to ignore the messages sent till auto revalidation is done or the User may prevent sending messages till auto revalidation is complete.

Example:

If the FioranoMQ Server is used for real-time trading, old messages cannot be used. If the Server is down the publisher should not send any messages as these will be sent once the Server is up and running. The flag should, therefore, be set to **2** which blocks the publisher from sending messages when connection with the server is down.

Dependencies:

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

PublishWaitDuringCSPSyncup

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

Valid value:

Time is specified in milliseconds. The default value is 1000 milliseconds.

SecurityManager

The SecurityManager implementation is used to create secure connections [HTTPS or SSL] with the FioranoMQ Server. This manager should be an implementation of the **fiorano.jms.runtime.IFMQSecurityManager** interface provided by FioranoMQ.

Valid value:

Class path to the security Manager class.

ServerProxyURL

The ServerProxyURL parameter holds the URL for the Server proxy through which TCP based Connections will be routed using HTTP Tunneling.

Valid value:

String value that contains information about the proxy provider.

Example: http://ClientProxyServer:80

Dependencies:

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

SleepBetweenSocketCreationTries

SleepBetweenSocketCreationTries is the time interval between two socket creation attempts. While creating a connection to the server, client RTL creates sockets on the Server. If the Server is unavailable at that time, RTL tries to create a socket for as many times as specified in the parameter – **MaxSocketCreationTries**. Between successive tries the RTL waits for the time specified in this parameter.

Valid value:

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

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, this value is **off**.

SocketTimeout

SocketTimeout is the time interval after which the socket calls a **timeout**. Timeout is when the flow of execution is stopped if still awaiting a reply from the socket. If the connection to the Server is lost, applications hang onto the socket for the maximum time set by this parameter.

Valid value:

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

SocksProxyURL

Sets the URL for the SOCKS Proxy through which HTTP based connections are routed.

Dependencies:

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

TCPBatchSize

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

Valid value:

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

Dependencies:

Batching should be enabled if 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 the Server.

Valid value:

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

java.naming.security.protocol

Java.naming.security.protocol is the name of the security protocol used to create secure connections with the FioranoMQ Server.

Valid value:

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

UpdateConnectURL

UpdateConnectURL is the boolean value that is used to specify whether a 'connect URL' of the connection factory should be updated when the Server restarts.

Valid values:

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

By default this value is **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 a single socket for send and receive.
- **no** - uses two socket for send and receive.

By default, this is **true**.

XASocketTimeout

Time interval after which the socket call will **timeout** in the XA enabled Server.

Valid value:

Time is specified 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

This chapter explains the parameters that can be configured at the FioranoMQ Server level. These parameters are commonly used across the FioranoMQ Server and can be configured in the default profile at the node FioranoMQ->Fiorano->etc->FMQConfigLoader in Offline Configuration mode using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for Online Configuration (through the Web Management Console), refer to **Chapter 30 FioranoMQ Web Management Console** of *FioranoMQ Handbook*.

FMQConfigLoader

This chapter explains the parameters present in **FioranoMQ->Fiorano->etc->FMQConfigLoader**

LazyRSCreation

LazyRSCreation is a value that determines if the 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 has to be started. When the thread creation is optimized, a thread is created when the first consumer joins a session created by the connection.

Valid values:

- **yes** – the thread creation is optimized on the client-side. A thread is created when the first consumer joins a session created by the connection.
- **no** – the thread is not optimized. A thread is created as soon as the connection is started. This is the default value.

Example:

When the number of threads running in an application need to be economized, the value must be set to **yes** so that there are no unnecessary threads created until a consumer joins a session.

AllowDurableConnections

A Durable Connection maintains connectivity with FioranoMQ at all times. Applications do not have to take care of storing, re-connecting and then forwarding 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:

The default value is **no**.

- **yes** - durable connections are enabled in the Server. All that a client application has to do to make its connection durable is to **AllowDurableConnections** in its application code. Please refer to Section 5.1.1. in the FioranoMQ Handbook.
- **no** - durable connections are not enabled in the 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 is raised. Since this data is generated only once, the application requires to store this data upon encountering the exception and while trying to connect to the server. This adds considerable load to the application. In such cases, a Durable Connection comes to the rescue by doing the work for the application. It automatically tries to reestablish the connections, stores the data in transit and sends it to the Server as soon as the connection is restored.

Dependencies:

AutoRevalidationEnabled - If **AllowDurableConnections** is set to **yes**, **AutoRevalidationEnabled** will automatically be set to **yes**.

UseSingleSocket

Each JMS Connection results in the creation of two sockets with the Server. By turning on this flag each Fiorano client is instructed to use a single socket rather than both sockets.

Valid values:

The default value is **no**.

- **yes** - each JMSConnection that a client makes results in one socket.
- **no** - each JMSConnection that a client makes results in two sockets.

Example:

Where sockets are to be economized, set this value to **yes**.

UseSingleSocketForAdmin

Turning on this flag, will result in starting off a thread for Admin Connection that waits for data on a socket, constantly. This enables the detection of a loss of connection immediately.

Valid values:

The default value is **yes**.

- **yes** - a thread for an Admin Connection is created on the socket and it waits for data on socket, constantly. This enables the detection of a loss of connection immediately.
- **no** - loss of connection will not be detected immediately.

Example:

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

UseFioranoCbr

In group based systems, messages are classified as belonging to a certain group referred to as a '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 a pre-definition of groups.

Valid values:

The default value is **no**.

- **yes** - enables content based routing at the Server-side. The Server will now have 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. Only absolute paths are to be used. Several XPath strings can be combined with AND/OR conditions. Please refer to 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 can have thousands of subscribers interested in information on stock trades. Each subscriber has their own selection criterion based on individual requirements. 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 to create default ACLs for different FioranoMQ objects. When set to **yes** different FioranoMQ objects are attached with their respective ACLs that decide which set of users/groups have access to a particular FioranoMQ object.

Valid values:

The default value is **yes**

- **yes** - FioranoMQ objects have their respective default ACLs. For default ACLs to have an effect, **AcIBasedDestinationSecurity** must be set to true.
- **no** - no default ACLs are created for FioranoMQ objects.

Dependencies:

See also `AcIBasedDestinationSecurity`.

AcIBasedDestinationSecurity

This indicates whether ACL based security is enabled.

Valid values:

The default value is **no**.

- **yes** - if enabled, User operations will first be checked against the associated ACL.
- **no** - if disabled, User operations will not be checked against the associated ACL.

Example:

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

AllPermissions

This parameter determines whether all default permissions for a new ACL should be negative or positive. For this option to have an effect **AcIBasedDestinationSecurity** must be set to **true**.

Valid values:

The 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 Users are to be added with their respective permissions added later.

AllowOnTheFlyAclCheck

This flag indicates whether modifications in an ACL would be reflected on clients while the Server is running. If **AclBasedDestinationSecurity** is set to **yes**, this will not affect those clients that are connected unless **AllowOnTheFlyAclCheck** is set to **yes** as well.

Valid values:

The default value is **no**.

- **yes** - clients that are connected behave similar to future clients that are to connect to the Server after modifications to the ACL.
- **no** - C clients that are connected to the Server behave as they did prior to the modifications to the ACL.

Dependencies

1. The **AllowOnFlyAclCheck** flag works for all permissions except in the instance below where: A publisher is publishing non-persistent messages on a topic.
2. The permission to create a publisher is revoked on this topic.
3. No exception is thrown even though the User is not allowed to publish because messages are sent in batch mode. Therefore, to receive an exception:

Solution One:

For NP messages, batching are enabled by default which leads to the behavior mentioned above. To receive an exception at the send API, set the **BatchingEnabled** parameter in the **ConnectionFactory** to FALSE.

Solution Two:

Add the line below in the client code environment while performing the lookup function:

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

Note: **env** is the environment passed while performing a JNDI lookup. This disables batching for that particular client.

See: **AclBasedDestinationSecurity**.

AllowOnTheFlyCreationOfDestinations

This flag controls the behavior of createTopic and createQueue APIS in a JMS Session. This deters a User from creating the destination using the session.createTopic or session.createQueue APIS. This does not prevent the admin User from creating destinations. Destinations can still be created using admin connection and admin services.

Valid values:

The default value is **yes**.

- **yes** - a new destination is automatically created when a destination name that does not exist is invoked.
- **no** - when a destination name that does not exist is invoked an exception will be thrown.

Example:

When clients are not be allowed to create destinations, the FioranoMQ Server administrator can make sure only the required destinations are created and clients do not add to existing destinations by setting this value to **no**.

EnsureUniquenessOfClientId

Every FioranoMQ JMS client is supplied with a clientId. This flag indicates whether uniqueness of clientIDs is required.

Valid values:

The default value is **yes**.

- **yes** - a connection that sets a client ID set on another connection will get an exception upon doing so.
- **no** - a connection that sets a client ID set on another connection will proceed normally.

Example:

When each clientId is required to be unique, this value must be set to **yes**.

MaxTransactionBufferSize

MaximumTransactionBufferSize is the maximum amount of data (in bytes) that a transacted Session would store in an uncommitted state. Any attempt to publish more data in an uncommitted transaction will result in an exception.

Valid values:

The default value is **10240000**

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

All values less than zero are equal to each other.

EnableAutoRevalidation

EnableAutoRevalidation enables controlling auto revalidation. Auto revalidation refers to Fiorano runtime's ability to detect a loss of connection and its subsequent attempts to automatically re-connect back to the server.

Valid values:

The default value is **no**.

- **yes** - Fiorano runtime will automatically try to reconnect to the server on detecting a loss of connectivity.
- **no** - the client will not reconnect automatically upon loss of connection.

Dependencies

The parameter above depends on **AllowDurableConnections**. If **EnableAutoRevalidation** 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 waits for messages before returning a response from the Server.

Valid values:

The default value is **10000**.

Range of integer values in java are: -2^{31} to $2^{31} - 1$.

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

IncludeMessageID

IncludeMessageID determines whether to include the messageID as part of the message.

Valid values:

The 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

ResumeTimeoutInterval is the integer specifying the resume timeout to be used by a Large Message Support(LMS) application. If timeout is not resumed within the given period, resuming timeout is stopped. For more details about LMS, please refer to Chapter 8 in FioranoMQ Handbook.

Valid values:

The default value is -1. (This indicates that the application will wait indefinitely.)

Range of long values in java - -2^{63} to $2^{63} - 1$.

All values less than zero are equal to each other.

MessageSelectorFactoryClassName

MessageSelectFactoryClassName is the class name that provides the Content Based Routing (CBR) Engine functionality for FioranoMQ Server. This property is used only if CBR is enabled in the FioranoMQ Server.

Valid values:

The default value is **fiorano.jms.cbr.cbr1.def.MessageSelectorFactory**.

SSLEnabled

SSLEnabled is the property which enables a SSL connection with the Server.

Valid values:

The default value is **no**.

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

PingEnabled

PingEnabled property controls client connections pinged automatically. Pinging is essential for detecting Network problems.

Valid values:

The default value is **yes**.

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

SystemEncodingFormat

SystemEncodingFormat encoding format for UTF.

Valid values:

Any string.

The default value is **UTF-8**.

Chapter 3: Queue Subsystem Level Configurations

This chapter explains the parameters that can be configured for JMS storage objects of type 'Queue'. In general, these parameters are the most common ones used in the Point-to-Point (PTP) model of JMS and they can be configured in the default profile at the node FioranoMQ->Fiorano->mq->ptp->QueuingSubsystem through Offline configuration using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for Online Configuration (through the Web Management Console), refer to **Chapter 30 FioranoMQ Web Management Console** of *FioranoMQ Handbook*.

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** in the *FioranoMQ Handbook*.

UnAckedQueueBufferSize

UnAckedQueueBufferSize is the buffer size of the receiver queue for unAcked messages. Where the size of the message queue for unAcked messages exceeds this buffer limit, the client gets a message to commit the received messages.

Values:

The default value set for this parameter is **10485760** bytes. Any positive integer can be assigned as a value for this parameter.

MaxNumberOfQueues

MaxNumberOfQueues are the maximum number of queues that can be created. This includes Temporary Queues. The JMS provider allows the creation of queues upto the number of queues specified here. Attempts to create queues beyond the number of queues specified here will be.

Values:

The default value set for this parameter is **-1**. This suggests that infinite queues may be created. Positive integers or -1 may be assigned to this parameter.

RedeliveryTriesOnListenerException

RedeliveryTriesOnListenerException parameter specifies the number of redelivery tries when RuntimeException is thrown in the 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**. Any positive **integer** may be assigned to this parameter. If this parameter is given a negative value, the message will not be delivered to the consumer.

Example:

Where there is a `RuntimeException` 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 specified. (This makes it compliant with JMS specifications.)

Dependencies:

Dependencies are occur only when the client session is either in **AUTO_ACK** or **DUPS_OK_ACKNOWLEDGE** modes.

DelayInMsgDeliveryOnListenerException

`DelayInMsgDeliveryOnListenerException` parameter specifies 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 the maximum redelivery tries value that is specified is reached.

Values:

The default value for this parameter is **-1**. This indicates that there is no delay between the tries at redelivery of a message when `RuntimeException` occurs in the `MessageListener`. Any positive value can be assigned to this parameter. There will not be any delay between attempts if a negative value is assigned to this parameter.

Dependencies:

Dependencies occur only in when the client session is either in **AUTO_ACK** or **DUPS_OK_ACKNOWLEDGE** modes.

CleanupDmqAtStartup

`CleanupDmqAtStartup` enables the cleanup of the **DeadMessageQueue** at the startup of the Server. If this flag is set to **true** prior to the startup of the Server all messages from the **SYSTEM_DEADMESSAGES_QUEUE** will be purged.

Values:

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

EnableSnooperOnAllQueues

EnableSnooperOnAllQueues specifies whether the snooper service is enabled on all queues. When snooping is enabled on a queue all messages that are sent to the queue are copied to a topic named **SYSTEM_MESSAGESNOOPER_QUEUE**. This insures that any subscribers registered on this topic can snoop the messages sent to that particular queue.

Values:

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

Example:

This parameter can be used when it is necessary to monitor messages being sent to the queues by the clients. To monitor messages that are being sent to queues enable this parameter and subscribe to the topic **SYSTEM_MESSAGESNOOPER_QUEUE**.

EnableDMQOnAllQueues

EnableDMQOnAllQueues specifies whether the DMQ service is enabled on all queues. When the DMQ service is enabled on a queue all expired messages that need to be stored are stored in the **SYSTEM_DEADMESSAGES_QUEUE** by default.

Values:

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

Example:

This parameter can be used when it is necessary to monitor messages that have expired.

Dependencies:

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

PrefetchCount

PrefetchCount specifies the prefetch count for single receive calls. Prefetch count is the number of messages fetched in a 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**. If the total size of the batch of messages exceeds the **MaxPrefetchSize** then the number of messages fetched may be less than the **PrefetchCount** set.

Values:

The default value of this parameter is **three**. Valid values are any positive integer values greater than or equal to the **PrefetchThreshold** value specified.

Dependencies:

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

PrefetchThreshold

PrefetchThreshold is the message count used for a single receive call. When the number of messages in the message queue of the client approaches this limit the client sends a request to the Server to fetching more messages. The number of fetched messages depends upon the value of the **PrefetchCount**.

Values:

The default for this parameter is **1**. Valid values for this parameter are any positive integer values less than the value of the PrefetchCount.

Dependencies:

If the message count approaches the **PrefetchThreshold**, only the client can send a request to fetch the number of messages set by the value of the PrefetchCount from the Server.

DefaultStorageTypeForQueues

DefaultStorageTypeForQueues string specifies the default storage type of the data related to all queues that may be used by the application.

Values:

The default value for this parameter is **File**. Valid values for this parameter are **File/RDBMS**. If this parameter is set to **File** all data about the queues is stored in the File Based Storage and the value of this parameter is set to **RDBMS**. All relating data about the queues is stored in RDBMS sbased Storage.

Dependencies:

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

MaxPrefetchSize

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 of this parameter is 264144 bytes, equivalent to 256KB. Valid values of this parameter include all positive integers.

Example:

If all the parameters are set to the default values and if the size of each message is 100KB then, at most, three messages can be fetched from the Server. This is because the total batch size exceeds the **MaxPrefetchSize** which is 256KB.

OverflowFromBottomEnabled

OverflowFromBottomEnabled concerns the overflow of non-persistent messages. This parameter determines whether messages are to be dropped from the bottom of the queue.

Values:

- Default value: False
- Valid values: True/False

DMQExpiryTime

DMQExpiryTime is the time measured in milliseconds after which any stored messages in the **Dead Message Queue** is deleted. If this value is set to a positive integer, a timer is registered which cleans up the **SYSTEM_DEADMESSAGES_QUEUE** at regular intervals.

Values:

The default value for this parameter is **0** milliseconds. Valid values for this parameter include all positive integer values which falls in the **Long** range.

EnableNotificationOnDeadMessage

EnableNotificationOnDeadMessages enables notification of DeadMessage. , When an expired message from a queue reaches the **SYSTEM_DEADMESSAGES_QUEUE** for storage, if this parameter is set to **true** an event will be raised.

Values:

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

Dependencies:

The DMQ should be enabled for the queue on which the message was originally sent. Only if the DMQ is enabled will the message be sent to the **SYSTEM_DEADMESSAGES_QUEUE** on expiry. Event notification will be activated only if the **NotifyWhenDead** flag for FioranoMessages is set to **true**. The 'getter' and 'setter' methods for this flag are **getNotifyWhenDead** and **setNotifyWhenDead** respectively.

EnableMessageMonitoring

EnablingMessageMonitoring enables the monitoring of messages and stores this data. This parameter takes a maximum of five seconds to be enabled.

Values:

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

FilePath

FilePath returns the file path set to store data acquired through monitoring of Server messages.

Values:

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

Dependencies:

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

TimeDuration

TimeDuration returns the **time duration** of messages received in seconds to enable monitoring message flow. The value for this parameter should be more than five seconds. This time duration is used to calculate the InBound and OutBound message rate (at the time the Server message data is being monitored).

Values:

The default value for this parameter is **3600** seconds. Valid values for this parameter include any positive integer values greater than five.

CleanupInterval

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

Values:

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

Example:

When deleting expired messages from the queue the `TimeDuration` parameter is used. After the specified period of time the queue gets cleaned of all expired messages.

InMemoryBlockingInterval

`InMemoryBlockingInterval` is the blocking interval that pushes a message into the **QueueBuffer** if the queue is full.

Values:

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

Example:

Blocks sender for the specified amount of time if the `QueueBuffer` is full.

InMemoryBufferSize

`InMemoryBufferSize` specifies the buffer size of the InMemory Cache. If the `InMemoryBuffer` is full, then reattempts to push the message will take place based on the value of the parameter **MaxPushAttempts**.. The wait interval is set using the parameter **WaitBetweenAttempts**.

Values:

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

PersistentInMemoryBufferSize

`PersistentInMemoryBufferSize` specifies the buffer size of the InMemory Cache of persistent messages. If the `PersistentInMemoryBuffer` is full, then reattempts to push the message will take place based on the value of the parameter **MaxPushAttempts**. The wait interval is set using the parameter **WaitBetweenAttempts**.

Values:

- Default value: 524288
- Valid values: Any positive integer in the **long** range.

MaxPushAttempts

`MaxPushAttempts` is the maximum number of attempts for pushing `InMemoryBuffer`. This parameter is particularly useful when the `InMemoryBuffer` is full. A maximum number of attempts to push the message to `InMemoryBuffer` are allocated.

Values:

- Default value: 64
- Valid values: Any positive integer value.

WaitBetweenAttempts

WaitBetweenAttempts specifies the wait time between attempts at pushing messages into a queue. This parameter is particularly useful when pushing incoming message is not successful in the first attempt. This parameter determines the amount of wait time before another attempt to push the message into InMemoryBuffer is undertaken.

Values:

- Default value: 16 (in milliseconds)
- Valid values: Any positive value in long range.

IsDbCleanupEnabled

IsDbCleanupEnabled specifies whether the queue cleaner utility is enabled or not

Values:

- Default value: False
- Valid values: True/False

UseInMemStatusCache

UseInMemStatusCache is the status Cache for messages.

Values:

- Default value: True
- Valid values: True/False

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

LoadBitSetOnStartup

LoadBitSetOnStartup loads the InMemory message index status **BitSet** on startup.

Values:

- Default value: False
- Valid values: True/False

EnableGMSQueueExpose

EnableGMSQueueExpose is a boolean value that exposes the **QueueAttributes** at runtime on the **GMS** queue. If set to **true**, a runtime MBean relating to the **GMS** queue is exposed.

Values:

- Default value: False
- Valid values: True/False

Exclusive

Boolean indicates about the sharing of cache tables. If set to **true**, exclusive cache tables are created for each queue. Non-exclusive cache tables are not supported in FioranoMQ.

Values:

- Default value: True
- Valid values: True/False

JumpDeletedFiles

JumpDeletedFiles is a boolean value specifying whether the deleted table files may be jumped. If this flag is used it jumps the deleted cache table of files to the next undeleted message in the DB.

Values:

- Default value: False
- Valid values: True/False

DepthMonitoringEnabled

DepthMonitoringEnabled turns on or turns off depth (the number of pending messages) monitoring count at the Server level. If this flag is set to **true**, depth monitoring is enabled at the Server level. Based on the depth monitoring flag at queue level, depth monitoring is enabled or disabled for that particular queue. If this flag is set to **false**, depth monitoring won't work for any queue irrespective of whether this flag is set at the queue level.

Behavior at server level:

If this value is set to **no** at the Server level, queue depth monitoring is stopped and notifications are terminated. If this value is set to **yes** at Server level, queue depth monitoring is enabled or disabled based on the configuration of 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 the Server level.

Values:

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

DepthMonitoringLevels

Jmx notifications are fired when pending message count crosses one of the depth values specified. All thresholds should be specified as positive values with each of them separated by a comma, for example: '10,20,30'. The value specified will be used for all destinations unless explicitly stated that it is for a particular destination.

Behavior at server level:

Depth levels specified will be used only if the value for the property at queue level is **null**.

Behavior at destination level:

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

Values:

- Server level: null
- Destination level: null

Chapter 4: Topic Subsystem Level Configuration

This chapter explains the parameters that can be configured for JMS storage objects of type 'Topic'. In general, these parameters are the most common ones used in the Publish-Subscribe (PubSub) model of JMS and they can be configured in the default profile at the node FioranoMQ->Fiorano->mq->pubsub->TopicSubsystem through Offline configuration using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for Online Configuration (through the Web Management Console), refer to **Chapter 30 FioranoMQ Web Management Console** of *FioranoMQ Handbook*.

Note: FioranoMQ 9.1.0 onwards includes the Destination Level Configuration support. For more on this, please refer to section **29.1 Support for Destination Level Configuration** in the *FioranoMQ Handbook*.

DupsOkBatchSize

The DupsOkBatchSize parameter indicates the number of messages received by a consumer for each consumer acknowledgement. A User specifies the number of messages after which client sends an acknowledgement to the server. This command will run only when the client application is accepting messages and the acknowledgement mode is DUPS_OK. The acknowledgement mode is specified when creating a session. (See JMS specifications for details on how to create a session.)

Values:

The default value for the parameter DupsOkBatchSize is 20. Any positive integer may be used. Negative values so not throw an exception. Therefore it is important to check that positive integers have been given.

Example:

This mode is used when duplicate messages are accepted. There is a possibility of duplicate messages being delivered when the Server is down or the Client is not available.

Dependencies:

This flag is dependent on the consumer session **acknowledgment** mode.

RedeliveryTriesOnListenerException

The RedeliveryTriesOnListenerException parameter specifies the number of redelivery tries when a RuntimeException is thrown on the consumer's message listener. This flag is used only when a consumer session **acknowledgment** mode is **auto_acknowledge** or **dups_ok_acknowledge**. The redelivered messages will be discarded once this value is reached. This parameter is useful if the **OnMessage** logic in the consume fails occasionally due to very high usage of the threads or the CPU.

Values:

The RedeliveryTriesOnListenerException parameter default value is two. Any positive integer may be used. A negative value given to this parameter will make the client application not deliver messages to the 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 dependent on the consumer session **acknowledgement** mode.

DelayInMsgDeliveryOnListenerException

The DelayInMsgDeliveryOnListnerException parameter specifies the delay in message redelivery when RuntimeException is thrown in MessageListener. This parameter is used in AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE mode only. The redelivered messages are discarded once the maximum redelivery tries value is reached.

Values:

The default value set for the DelayMsgDeliveryOnListenerException parameter is **-1**. This value indicates that there is no delay between redelivery tries for a message, when RuntimeException occurs in MessageListener. Any positive value in **Long range** can be assigned to this parameter. If a negative value is assigned to this parameter there will not be a delay between attempts.

Dependencies:

Dependencies are only applicable when the Client Session is either in AUTO_ACKNOWLEDGEMENT or DUPS_OK_ACKNOWLEDGE modes.

PublishWait

The PublishWait parameter specifies the wait interval of the publisher before the next published call. 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 the PublishWait parameter is 50 milliseconds. (This parameter does not accept values less than or equal to zero).

Example:

This flag can be used for the pub/sub model. This flag should be increased when a larger message is accepted and more processing time for a subscriber is needed.

Dependencies:

The PublishWait flag can be used only if EnablePublisherSlowDown is set to yes.

PublishWaitTimeRecheckUnits

The PublishWaitTimeRecheckUnits parameter is used for the flow control algorithm. When the PSQ size reaches 95% the publisher is made to wait repeatedly until the size drops down to 85%. The MaxWaitTime is distributed into segments of time during which the size of the PSQ is checked. This applies to all the publishers.

Values:

The default value for the PublishWaitTimeRecheckUnits parameter is eight. Any positive integer may be assigned to this parameter.

Example:

This flag may be used when there are a certain number of slow subscribers in use.

Dependencies:

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

MaxNumberOfTopics

The Maximum number of topics that can be created using this parameter by default is -1, which is allowing an infinite number of topics to be created. If this value is set to a finite value, the JMS provider will allow the creation of topics to the number stated. A message is displayed when this limit is reached.

Values:

The default value set for the MaxNumberOfTopics parameter is **-1**. The number of topics that can be created is **unlimited**. Any positive integer and -1 can be assigned to this parameter.

EnablePublisherSlowdown

EnablePublisherSlowdown slows down the publisher when messages get stored in the disk after the session buffer overflows due to a slow subscriber and due to the PSQ size reaching its threshold value. Setting this parameter to false may lead to a message loss for slow subscribers. This property applies to all the publishers.

Values:

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

Example:

This flag may be used when there are slow subscribers.

Dependencies:

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

DropOldestMessage

DropOldMessage is a parameter indicating whether or whether new messages should be dropped after the PSQ crosses the maximum value and all Server buffers are filled.

Values:

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

Dependencies:

The DropOldestMessage parameter comes to use only when PublisherSlowDown is disabled in the Server and slow subscribers are active. This applies to the session level of the subscriber. This situation can also occur when the PSQ is filled with persistent messages.

DefaultStorageTypeForTopics

The parameter DefaultTypeForTopics acquires the default storage types for Topics.

Values:

The default value set for this parameter is **File**. The values that maybe assigned to this parameter are **File/RDBMS**. If this parameter is set to **File**, all data on topics is stored in the file based storage. If this parameter is set to **RDBMS**, all data on topics is stored in the RDBMS based storage.

Dependencies

If this value is set to **RDBMS**, then the parameter **EnableRDBMS** should be set to **true**. If this value is set to **File**, then the parameter **EnableRDBMS** 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. If set to false, the messages will be added directly to the ConnectionBuffer which will lead to an extreme slowdown of publishers and excess message loss in some cases. This applies to all the topics.

Values:

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

EnableSnooperOnAllTopics

EnableSnooperOnAllTopics specifies whether the Snooper Service is enabled on all topics. When snooping is enabled on a queue, all the messages that are sent to this queue are copied to a topic named **SYSTEM_MESSAGESNOOPER_TOPIC**, so that any subscribers registered to this topic can snoop messages sent to that particular topic.

Values:

The default value set for this parameter is **false**. The values that may be assigned to this parameter 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 to monitor messages 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**. This value is not assignable.

IsForceFullyKillIDSubscriber

IsForceFullyKillIDSubscriber allow the forcefully killing of the Durable Subscriber. If set to true, a new Durable Subscriber gets registered with the Server. The former Durable Subscriber will be forcefully killed and this event will be successfully registered with the Server. This parameter can be applied to all Durable Subscribers.

Values:

The default value for the IsForceFullyKillIDSubscriber parameter is **false**. The values that may be assigned to this parameter are **true/false**.

Dependencies:

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

MaxPersistentStoreSize

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

Values:

The default value for the MaxPersistentStoreSize parameter is 1073741824 bytes. The values that may be assigned to this parameter are any positive values in the **Long range** and **-1**. If set to -1, the maximum limit for the PSQ size will be unlimited.

UseMessagePersisterForDurableSubscriber

The UseMessagePersisterForDurableSubscriber parameter is used to Enable/Disable the use of a separate algorithm used in persisting messages for a Durable Subscriber.

Values:

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

MaxMsgInOneRecv

MaxMsgInOneRecv parameter indicates the maximum number of messages that can be received in one call. This parameter can be applied to all subscribers.

Values:

The default value for the MaxMsgInOneRecv parameter is **128**. The value that may be assigned to this parameter is any positive **Integer** value.

BasicAllowedSize

BasicAllowedSize parameter indicates the maximum buffer limit on a connection. If the maximum buffer limit is exceeded, every push in the connection buffer will have to wait until some space is created. This buffer does not usually go over the maximum limit allowed since any over flow is avoided through the flow control between the session layers at the RTL and at the server.

Values:

The default value for the BasicAllowedSize parameter is **131072** bytes. the values that can be assigned to this parameter are positive integral values.

UseOptimizedTCPReceive

The UseOptimizedTCPReceive parameter is enabled. This algorithm was introduced in the FioranoMQ 7.5 version and is meant to provide high performance at the TCP layer on the subscriber side. This parameter may be applied to all Subscribers.

Values:

The default value for the UseOptimizedTCPReceive parameter is **true**. The values that may be assigned to this parameter are **true/false**.

EnableMessageMonitoring

The EnableMessageMonitoring parameter enables message monitoring and storage of data. If set to true, the data-traffic on the topic is stored in **InMemory** as well and is stored on the disk as well. Traffic can be monitored by viewing the files or by calling the required JMX API. This parameter may be applied to all Topics.

Values:

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

FilePath

The FilePath parameter returns the file path set to store server messages that monitor data. Dta is stored only if **EnableMessageMonitoring** is set to **true**.

Values:

The default value for the FilePath parameter is **%FIORANO_HOME%\fmq\profiles\FioranoMQ\run\PUBSUB\JMSX_MESSAGE MONITOR**. Any valid **file path** may be assigned to this parameter.

Dependencies:

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

TimeDuration

The TimeDuration parameter monitors the time duration in **seconds** of the flow of messages. The value of this parameter should be more than five seconds. Traffic is monitored after each interval in message flow if the **EnableMessageMonitoring** is set to **true**.

Values:

The default value for the TimeDuration parameter is **3600** seconds. The values that may be assigned to this parameter are any positive integer values greater than 5.

PurgeLevelIndex

The PurgeLevelIndex parameter indicated the number of persistent messages allowed prior after which purging is done. This parameter will function when **OptimizedTCPReceive** is set to **false**. This parameter may be applied to all Topics.

Values:

The default value for the PurgeLevelIndex parameter is **100**. The values that may be assigned to this parameter are any positive integer values.

Dependencies:

This parameter is used only for **persistent** messages and only when **UseOptimizedTCPReceive** is set to **false**.

MinPurgeDifference

The MinPurgeDifference parameter indicates the minimum number of persistent messages allowed after which purging is done. Purging is done in batches in the Server and this parameter indicates the minimum number of **delete requests** that the server will accumulate prior to deleting messages. This parameter may be applied to all Topics.

Values:

The default value for the MinPurgeDifference parameter is **50**. The values that may be assigned to this parameter are any positive integer values.

Dependencies:

This parameter is used only for **persistent** messages and only when **UseOptimizedTCPReceive** is set to **false**.

AllowDeletionOfSubtopics

AllowDeletionOfSubtopics is a boolean parameter determining if deletion of SubTopics is allowed. If set to true then deleting a topic will automatically delete the SubTopics. This parameter may be applied to all Topics.

Values:

The default value for the AllowDeletionOfSubtopics parameter is **true**. The values that may be assigned to this parameter are **true/false**.

NPMessagePublishTimeout

The NPMessagePublishTimeout parameter indicates the maximum time a publisher will wait while trying to push messages to a connection level buffer when it is full. This parameter also indicates the maximum time a publisher will wait to push the message to the PSQ after it has reached 95% of its maximum limit. This parameter applies to **non_persistent** messages as well as for **persistent** messages. The waiting time does not end until the buffer reaches 85% usage. This parameter may be applied to all the publishers.

Example: NPMessagePublishTimeout = 1

Setting the NPMessagePublishTimeout to a low value ensures that the publisher does not slow down. A low value setting on this parameter achieves the same result as setting the EnablePublisherSlowdown to **False**.

Values:

The default value for the NPMessagePublishTimeout parameter is **120000** milliseconds. The values that may be assigned to this parameter is any positive integer value.

PublishBackoffThreshold

The PublishBackoffThreshold parameter indicates the PSQ size threshold after which the backoff algorithm is activated. Once the threshold is reached then flow control will be activated and the publishers will be made to wait for the buffers to clear.

Values:

The default value for the PublishBackoffThreshold parameter is **0.6**. The values that may be assignable to this parameter are any positive Double values < 1.

IgnorePSQSizeForPersistentMessages

The IgnorePSQSizeForPersistentMessages parameter indicates whether to persist persistent messages in PSQ even through the PSQ is full. If set to **true** if the **MaxPersistentStoreSize** (PSQ) is exceeded the messages will be stored to the disk. This parameter may be applied to all Publishers.

Values:

The default value for the IgnorePSQSizeForPersistentMessages parameter is **true**. The values that may be assignable to this parameter are **true/false**.

Dependencies:

This parameter is applicable only for messages with in the **persistent** delivery mode.

EnableSessionStateChangeNotification

This parameter defines whether to raise notifications related to the topicSession's state changes corresponding to slow subscribers. That is, whenever the underlying subscriber undergoes state changes - becomes slow or normal - a notification is raised to all event listeners.

In order to receive this event, the the object (which implements javax.jms.NotificationListener) should be added on the MBean specified by the object name -

```
"Fiorano.jmx.notifications:ServiceType=EventManager,Name=TopicConnectionEventManager".
```

Kindly refer to the "fmq/samples/JMX/JMXNotifications.java" for more information.

Values: The default value for this parameter is FALSE and the valid values are TRUE and FALSE.

Chapter 5: Connection Manager Configurations

This chapter explains the parameters that can be configured for the connection manager used in the FioranoMQ Server. In general, these parameters are the most common ones used for connection management in the server and they can be configured in the default profile at the node FioranoMQ->Fiorano->socketAcceptors->port-1->ConnectionManager through Offline configuration using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for Online Configuration (through the Web Management Console), refer to **Chapter 30 FioranoMQ Web Management Console** and on Connection Management, refer to **Chapter 4 Connection Management** of *FioranoMQ Handbook*.

Default

Default identifies whether the Connection Manager is set as a default option. All connection factories contain URL information of the Connection Manager. If no Connection Manager has its flag set to **true** the connection factories will retrieve their URL information from the default server URL.

Values:

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

CFNameAppender (Not being used in FioranoMQ)

The CFNameAppender parameter specifies the suffix to be attached to the ConnectionFactory created for the ConnectionManager.

Values:

The default value for the CFNameAppenderparameter is null. The values assignable to this parameter may consist of a string of characters.

ReaderCacheDisabled

ReaderCacheDisabled is a boolean parameter indicating whether the connection cache is created on the read socket. This cache is used while reading bytes from Sockets. If this parameter is set to **true**, then the appropriate amount of memory is allocated for each Socket Read. If set to **false** memory is allocated for Socket Read based on the exact memory needed decreasing the number of memory re-allocations.

Values:

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

TCPWindowSize (Not being Used in FioranoMQ)

TCPWindowSize is used for the three ways handshaking process required for connection creation.

Values:

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

HandshakeInWorkerThread

The HandshakeInWorkerThread is a boolean parameter indicating whether the HandShaking process was performed while accepting the connection request from the client or when creating the socket initialization context.

Values:

The default value for the HandshakeInWorkerThread parameter is **true**. The values that may be assigned to this parameter are **true/false**.

MagicTimeout

The MagicTimeout parameter sets the socket timeout for the socket that is created. If a request does not arrive on the socket within the interval specified, a socket cleanup is performed.

Values:

The default value for the MagicTimeout parameter is **60000** seconds. The values that may be assignable to this parameter are any non-negative integer values. If this value is set to zero the 'timeout' is considered to be infinite.

Example:

If the MagicTimeout value is set to 60000 and a connection is made to the FioranoMQ Server and the Server fails to communicate, the socket will be cleared after one minute. This connection must communicate with the JMS Server within one minute or the Socket will be cleared and the connection closed.

Port

The Port parameter is used for binding the socket at the time of Server startup.

Values:

The default value set for the MagicTimeout parameter is **1856**. A valid port value is between 0 and 65535.

Example:

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

Protocol

The Protocol parameter is used for establishing a connection between the Client and the Server.

Values:

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

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

Dependencies:

If the Protocol is set to either SUN_SSL or HTTPS_SUN, **SSLEnabled** must be set to **true**.

UseNagle

Use Nagle is a boolean parameter indicating whether the Nagle Algorithm was used in creating the socket. Using the Nagle Algorithm enables improving the efficiency of the TCP/IP networks by reducing the number of packets that need to be sent over the network. The Nagle Algorithm is used to automatically concatenate a number of small buffer messages. The process of concatenating several small messages (called nagling) increases the efficiency of a network application system by decreasing the number of packets to be sent.

Values:

The default value set for the UseNagel parameter is **false**. The values that can be assigned to the UseNagel parameter are **true/false**.

MonitoringRequest

The MonitoringRequest parameter is the default monitoring request ID sent as a validation for communication.

Values:

The default value for the MonitoringRequest parameter is **13**.

MonitoringResponse

The Monitoring Response parameter is the default monitoring request response sent as a validation for communication.

Values:

The default value for the Monitoring Response parameter is null.

EnforceJsseAuthentication

The EnforceJsseAuthemtcation parameter checks whether authentication is to be done for JSSE client requests accept by the S. The value of this parameter is preset and useful for sockets in the server.

Values:

The default value for the EnforceJsseAuthemtcation parameter is **false**. The values considered valid by this parameter 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 the client does not provide authentication information, the server socket will not continue communication with the client and the connection will be dropped.

MaxClientConnectionsCount

The MaxCientConnectionsCOunt is the maximum number of client connections allowed with the Connection Manager. A value of -1 indicates no upper limit for connections allowed.

Values:

The default value for this parameter is **1024**. The values that this parameter considers valid are -1 and any positive integer value. If this value is set to -1, there is no maximum client connection limit..

Example:

If the value for this parameter is set for 200 and 200 external connections from clients are made to the server, any attempts to create more connections which are not LPC Enabled will not be successful and all connections will be dropped.

AdminConnection (Not being used)

AdminConnection serves the client by checking whether the connection is an Administrative Connection or a Normal Connection.

Values:

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

Path

Deprecated attribute. Please refer to SSL documentation. Root directory in which this connection manager searches for digital signatures for handling secure connections

Values:

The default value for the Path parameter is certs. Any valid file may be assigned to this parameter.

Example:

By default, the Path parameter in the FioranoMQ profile directs queries regarding digital signatures to the folder *\$FIORANO_HOME/fmq/profiles/FioranoMQ/certs*.

ServerAddress

ServerAddress is the InetAddress of the local machine that the FioranoMQ Server needs to bind with. This option is useful when the FioranoMQ Server is running on multi-homed machines.

Values:

The default value for the Server Address parameter is null. A valid machine name or a textual representation of the a machine's IP address may be assigned as a value for this parameter.

Example:

If a machine has an IP address of <192.168.1.209> and hostname of <example.hostname>, either one of these may be given as its ServerAddress.

HandShakingDisabled

HandShakingDisabled is a boolean parameter indicating whether the **handshake** is disabled is enabled upon connection creation.

Values:

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

- True – Set to True if a **handshake** between the client and the server is not needed after creating a new connection.
- False – Set to False to perform a **hand shake** between the client and the server after creating a new connection.

Dependencies:

Refer to the special note below.

Special Note:

When setting the value of the HandShakingDisabled parameter to true in the server set a System Property **DisableHandShake=true** in the client JVM when.

Caution: Setting this parameter may make the Server DOS attack prone making the client JVM hang if it is unable to set the Client System Property for the same.

ManagerClassName

Deprecated Attribute. Please refer to SSL documentation. Class Name of the Security Manager for wrapping TCP/HTTP communication. This is an implementation of IExSecurityManager. One can write and plug-in his server security manager. Possible default values are fiorano.jms.ex.sm.def.DefaultJSSESecurityManager

Values:

The default value considered valid for the ManageClassName parameter is fiorano.jms.ex.sm.def.DefaultJSSESecurityManager. The value that is considered valid for this parameter.

Example:

The ManageClassName is an implementation of IExSecurityManager. The parameter may be used to manage certificates which are used for handling secure connections.

Dependencies:

The ManageClassName parameter is used only when SSLEnabled is set to true.

FioranoMQServer

The FioranoMQServer is a boolean value that determines whether the connection created is for the FioranoMQ Server or if it is for external servers.

Values:

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

- True - Connection created is for the FioranoMQ Server.
- False – Connection created if for FES/FPS Servers.

UseForPeerToPeerCommunication (Not being used in FioranoMQ)

UseForPeerToPeerCommunication is a boolean parameter which determines whether the connectionManager created should be used to communicate with Peer Servers. (This parameter is applicable to FPS only.)

Values:

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

UseSystemPropsForSSL

Boolean which determines whether System properties like javax.net.ssl.keyStore to load key store and trust store for SSL Communication. If we enable this parameter, then the FioranoMQ Server assumes that the "Context", in which the SSL Server Socket is created, is initialized by the JDK using the system properties like -

- javax.net.ssl.keyStore
- javax.net.ssl.keyStorepassword
- javax.net.ssl.keyStoreType
- javax.net.ssl.trustStore
- javax.net.ssl.trustStorePassword
- javax.net.ssl.trustStoreType
- javax.net.debug

Please see SSL documentation for further information.

Valid Values:

True/False. Default value for this parameter is false.

CipherSuiteList

Semi-colon separated list of Cipher Suites to be enabled for SSL/TLS Communication. By default every cipher suite supported by JDK/JRE is enabled.

Valid Values:

Any cipher suite supported by the JDK/JRE. This list can be obtained by invoking "javax.net.ssl.SSLServerSocket.getSupportedCipherSuites()" API. Default value for this parameter is null, which means that every cipher suite supported is enabled.

Chapter 6: Dash Board Configurations

This chapter explains the parameters that can be configured for the Dashboard, used as the Monitoring and Administration tool for the FioranoMQ Server. These parameters can be configured in the default profile at the node FioranoMQ->Fiorano->etc->JettyServer through Offline configuration using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for information on the Web Management Console, refer to **Chapter 30 FioranoMQ Web Management Console** of *FioranoMQ Handbook*.

JettyServer

The parameters present in **Fiorano->etc->JettyServer** are explained in this chapter.

MaxThreads

MaxThreads are the maximum number of threads that can be created for the Jetty Server that is running.

The MaxThreads parameter allows the User to configure the maximum number of threads the dashboard may have active at any time. This value can be decreased if the User does not expect that much traffic will be directed towards the Jetty Server.

Default Value:

The Jetty Server thread pool has a default limit of 250 threads.

EnableStart

EnableStart is a boolean value describing whether the Jetty Server should be activated along with FioranoMQ Server.

If the User does not wish the Jetty Server to be setting the Jetty Server to false will prevent the Server from running.

Valid Values:

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

Default Value:

Yes (In Studio)/True - The Jetty Server is started along with the FioranoMQ Server.

MinThreads

The MinThreads parameter is the minimum number of threads that are created in the dashboard that is running.

The MinThreads parameter allows the User to configure the minimum number of threads that should be present in the dashboard thread pool. This value can be increased if the user expects high traffic directed towards the dashboard.

Default Value:

There should be a minimum of 10 threads by default in the Jetty Server thread pool.

LowThreads

The LowThreads parameter allows the User to set the low threads threshold for the thread pool of the Jetty Server. A low resource state detects a threshold or a limit set on the number of threads that the Jetty Serverthread pool may have. The limit set on the number of threads is specified by this parameter.

Default Value:

When a low resource is detected, the Jetty Serverthread pool threshold, by default, is set to 25.

MaxIdleTime

The MaxIdleTime parameter sets the maximum idle time for a connection. The maximum idle time is applied while:

- Waiting for a new request to be received on a connection.
- Reading the headers and content of a request.
- Writing the headers and content of a response.

Default Value:

The maximum time, by default, that the Jetty Server may remain idle is 30000 milliseconds.

Dependencies:

LowResourceMaxIdleTime

LowResourceMaxIdleTime

The LowResourceMaxIdleTime is the value specified to reduce the maximum idling time of a connection when a 'low resource' status is detected.

Default Value:

The default value of the MaxIdleTime when there is a low resource state is 5000.

Dependencies:

MaxIdleTime

PortNumber

PortNumber is the value that 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, the jetty server port for each of these Server instances must be different as all of them cannot be bound to the same port.

Default Value: The default port for the jetty server is 1790.

Chapter 7: Security Configuration

This chapter explains the configurable parameters related to the FioranoMQ Security module (Principal and ACL Managers) in the server. These parameters can be configured in the default profile at the node FioranoMQ->Fiorano->security->AcIManager->* and FioranoMQ->Fiorano->security->PrincipalManager->* through Offline configuration using Studio.

For information on how to configure the FioranoMQ Server in Offline mode, refer to **Chapter 3 Offline Configuration Through Profile Manager** and for information on FioranoMQ Security, refer to **Chapter 7 FioranoMQ Security** of *FioranoMQ Handbook*.

NativeFileBasedAcIManager

Presented below are the details of the parameters in **FioranoMQ->Fiorano->Security->AcIManager->NativeFileBasedAcIManager**.

LdapBasedAcIManager

Following are the parameters present in FioranoMQ->Fiorano->Security->AcIManager ->LdapBasedAcIManager.

LdapProviderUrl

This parameter denotes the LDAP Server's URL. For correct URL format, please refer to your LDAP Server's documentation. This value along with LdapProviderDn, later on while initializing the LDAP context, is passed in the environment with key as 'java.naming.provider.url'.

Valid value: The value of the property should contain a URL string (e.g. "ldap://somehost:389").

Default value is ldap://164.164.128.11:389/

LdapProviderDn

This parameter points to the base domain on LDAP Server, under which FMQ would create its repository. This value along with LdapProviderUrl, later on while initializing the LDAP context, is passed in the environment with key as 'java.naming.provider.url'.

Valid value: The value should be a distinguished name of an LDAP entry which uniquely identifies the entry.

Default value is "o=fiorano,c=us".

Principal

This parameter takes input the LDAP User Name. This is the user name which would be used to connect to the LDAP Server. This is provided as "java.naming.security.principal" (caller authentication to LDAP service is done using this) when creating the `javax.naming.directory.InitialDirContext` to the LDAP provider for performing operations on it. This is LDAP-provider specific.

Valid value: This parameter takes any valid user name in String representation.

Default value is "cn=FMQRoot,o=fiorano,c=US".

Credentials

This parameter takes input the LDAP Server Login Password. This is provided as "java.naming.security.credentials" (caller authentication to LDAP service is done using this) when creating the `javax.naming.directory.InitialDirContext` to the LDAP provider for performing operations on it.

Valid value: This parameter takes any valid password in String representation.

Default value is "secret".

LdapInitialCtxFactory

This parameter takes input the class name for LDAP Server's Initial Context Factory to be supplied to Initial Context. This is provided as "java.naming.factory.initial" when creating the `javax.naming.directory.InitialDirContext` to the LDAP provider for performing operations on it.

Valid value: The value of the property should be the fully qualified class name of the factory class that will create an initial context.

Default value is "com.sun.jndi.ldap.LdapCtxFactory".

LdapAclContextDn

This parameter points to the location at which ACLs would be stored. This parameter takes a String value which denotes the name of the tree-node under which different ACLs are to be created.

Valid value: The value should be a relative distinguished name of an LDAP entry under the main entry (provided in `LdapProviderDn`) which uniquely identifies the entry.

Default value is "cn=ACL".

LdapSecurityAuthentication

The parameter value provided with LdapSecurityAuthentication is supplied as "java.naming.security.authentication" in the environment while creating the InitialDirContext to the LDAP provider. Possible values are none, simple, strong. "none" for no authentication, "simple" for clear-text username-password authentication and "strong" for user-binding using strong authentication mechanisms via SASL.

Valid value: This parameter takes input "none" or "simple" or "strong".

Default value is "simple".

LdapPollInterval

This parameter denotes the interval (in milliseconds) after which the LDAP Server Connection is polled. Negative Value disables polling.

Valid value: This parameter takes input any integral value.

Default value is "-1".

LdapPrimaryServerReconnectAttempts

This parameter denotes the number of reconnect attempts LDAP primary server makes in case the LDAP server connection polling is unsuccessful.

Valid value: This parameter takes input any integral value.

Default value is "5".

BackupLdapProviderUrl

This parameter denotes the backup LDAP server provider URL, which is tried when the primary LDAP Server becomes unavailable. See LdapProviderUrl.

Valid value: See LdapProviderUrl.

Default value is "".

BackupLdapProviderDn

This parameter points to the base domain on Backup LDAP Server, under which FMQ would create its repository. See LdapProviderDn.

Valid value: See LdapProviderDn.

Default value is "".

BackupPrincipal

This parameter denotes the backup LDAP Server's user name. This is the user name which would be used to connect to the Backup LDAP Server. See Principal.

Valid value: See Principal.

Default value is "cn=FMQRoot,o=fiorano,c=US".

BackupCredentials

This parameter denotes the backup LDAP Server's login password. See Credentials.

Valid value: See Credentials.

Default value is "secret".

BackupLdapInitialCtxFactory

This parameter denotes the class name for LDAP Server's Initial Context Factory to be supplied to Initial Context. See LdapInitialCtxFactory.

Valid value: See LdapInitialCtxFactory.

Default value is "".

BackupLdapSecurityAuthentication

This parameter denotes the backup LDAP server's security authentication protocol. See LdapSecurityAuthentication.

Valid value: See LdapSecurityAuthentication.

Default value is "none".

AclEntryObjectClasses

Comma separated object class names for an ACL entry in the LDAP provider. For storing ACL's in an LDAP provider, the LDAP-provider should be made configurable to store the java objects as per RFC 2713 Schema for representing Java objects in LDAP.

Valid value: Comma-separated strings which denote object class names of an LDAP-provider.

Default value is "top,javaContainer,javaObject".

MaxAcePerAcl

The MaxAcePerAcl parameter provides the ability to limit the maximum number of access control entries for each ACL.

Valid value:

Any string

The default value of this parameter is integer greater than 0.

XMLFileBasedAcIManager

Following are the parameters present in Fiorano->security->AcIManager->XMLFileBasedAcIManager

Path

String value of path where the XML file containing the persisted ACLs is created

Default Value: realm/acl

Filename

Name of the file in String format that stores the persisted ACL information

Default Value: acl.xml - The default xml filename

MaxAcePerAcl

Number of Maximum Access control entries per ACL in integer format

Valid values: Integer greater than zero

Default value: 100

RdbmsBasedAcIManager

Following are the parameters present in Fiorano->security->AcIManager->RdbmsBasedAcIManager

MaxAcePerAcl

Number of Maximum Access control entries per ACL in integer format

Valid value : Integer greater than zero

Default value: 100

Url

URL for RDBMS Server in String format

Example: jdbc:oracle:thin:@localhost:1521:ORCL for oracle server

Password

String value of password for RDBMS User - Used for connecting to the RDBMS Server

Username

String value of database User Name - Used for connecting to the RDBMS Server.

Default Value: sa

DBDriver

JdbcDriver used for RDBMS based data storage in String format

Default Value: org.hsqldb.jdbcDriver - The default database driver class name

ConnectionTimeoutInterval

Integer value of database connection open timeout

Default Value: 10000 - The time in milliseconds for connection timeout

Valid Value: Non negative Integer with value zero representing wait forever

ReconnectTimeInterval

Integer value of time interval (in msec) between two reconnect attempts in disconnected state.

Default Value: 10000 – The time in milliseconds taken by Acl Manager between reconnect attempts

Valid Value: Non negative Integer

EnableReconnect

Boolean value that specifies if Acl Manager has to try to reconnect to database server in case of a connection breakup

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.

StatementFile

String that represents name of the properties file that is used to define name-value properties specific to a particular database.

Default Value: aclsqstatements.properties - The default properties used which is located at %FIORANO_HOME%/fmq/profiles/%YOUR_PROFILE%/conf

ConnectionLostErrorMessage

String representing 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.

InitializeDB

Boolean value to enable/disable initialization of the underlying Database by creating necessary tables on startup if required

Valid values:

- **True**- Enables initializing the DB
- **False** - Disables initializing the DB

Default Value: True - Enables initializing the DB.

FileDBManager3

Presented below are the details of all the parameters in **FioranoMQ->Fiorano->Security->AcIManager->FileDBManager->FileDBManager3**.

Path

The security configuration, the ACL, is stored in the file based data storage in the path specified by the Path parameter.

Valid values:

The default values are **SDB/REALM.ACL**

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

DeletedThresholdPercent

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

Valid values:

The default value of this parameter is **50**. This indicates that if the ratio of deleted entries to total entries is equal to or greater than 0.5 then the Cache Compaction Process will be invoked.

The range of integer values in java that are considered valid by the DeletedThresholdPercent parameter are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero provide the same result. All values greater than or equal to 100 provide the same result.

Example:

Use the example of the ratio of deleted entries/total entries in the table. If there a chance that many entries will be marked deleted often, a higher number should be entered as the parameter value.

DeletedThresholdCount

A cache stores and manages the list of all tables pertaining to group based information in the file-storage based ACL. After repeated modification of the ACL, old entries in the table are marked as deleted. This value provides a threshold count of deleted Entries for starting the Cache Compaction Process. At the end of the Cache CompactionProcess the table will not have any deleted entries. Since cache compaction involves CPU intensive File Operations, this flag is to be used along with DeletedThresholdPercent to ensure it is not invoked too often.

Valid values:

The default value of this parameter is **10**.

The range of integer values in java that are considered valid by the DeletedThresholdCount parameter are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero provide the same result.

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 of the number of deleted entries needed to trigger a Cache Compaction. It is best to choose a value that is high so as to enable triggering a Cache Compaction.

DbTableNamePrefix

The database table names are usually prefixed with a string that can be modified by the DbTableNamePrefix parameter.

Valid value:

The default value of the DbTableNamePrefix is #

Any String value is a valid parameter.

Example:

If the database table is to be saved in its current state and the the current state is to be saved to be used in the future, modify the prefix to another relevant string and change it back to the original string when the saved state of the database table is to be used.

NativeFilePrincipalManager

Presented below are the details of the parameters in **FioranoMQ->Fiorano->Security->PrincipalManager->NativeFilePrincipalManager**.

LdapPrincipalManager

Following are the parameters present under the FioranoMQ->Fiorano->Security->PrincipalManager->LdapPrincipalManager.

LdapProviderUrl

This parameter denotes the LDAP Server's URL. For correct URL format, please refer to your LDAP Server's documentation. This value along with LdapProviderDn, later on while initializing the LDAP context, is passed in the environment with key as 'java.naming.provider.url'.

Valid value: The value of the property should contain a URL string (e.g. "ldap://somehost:389").

Default value is ldap://164.164.128.11:389/

LdapProviderDn

This parameter points to the base domain on LDAP Server, under which FMQ would create its repository of users and groups. This value along with LdapProviderUrl, later on while initializing the LDAP context, is passed in the environment with key as 'java.naming.provider.url'.

Valid value: The value should be a distinguished name of an LDAP entry which uniquely identifies the entry.

Default value is "o=fiorano,c=us".

Principal

This parameter takes input the LDAP User Name. This is the user name which would be used to connect to the LDAP Server. This is provided as "java.naming.security.principal" (caller authentication to LDAP service is done using this) when creating the javax.naming.directory.InitialDirContext to the LDAP provider for performing operations on it. This is LDAP-provider specific.

Valid value: This parameter takes any valid user name in String representation.

Default value is "cn=FMQRoot,o=fiorano,c=US".

Credentials

This parameter takes input the LDAP Server Login Password. This is provided as "java.naming.security.credentials" (caller authentication to LDAP service is done using this) when creating the javax.naming.directory.InitialDirContext to the LDAP provider for performing operations on it.

Valid value: This parameter takes any valid password in String representation.

Default value is "secret".

LdapInitialCtxFactory

This parameter takes input the class name for LDAP Server's Initial Context Factory to be supplied to Initial Context. This is provided as "java.naming.factory.initial" when creating the javax.naming.directory.InitialDirContext to the LDAP provider for performing operations on it.

Valid value: The value of the property should be the fully qualified class name of the factory class that will create an initial context.

Default value is "com.sun.jndi.ldap.LdapCtxFactory".

LdapSecurityAuthentication

The parameter value provided with LdapSecurityAuthentication is supplied as "java.naming.security.authentication" in the environment while creating the InitialDirContext to the LDAP provider. Possible values are none, simple, strong. "none" for no authentication, "simple" for clear-text username-password authentication and "strong" for user-binding using strong authentication mechanisms via SASL.

Valid value: This parameter takes input "none" or "simple" or "strong".

Default value is "simple".

LdapPollInterval

This parameter denotes the interval (in milliseconds) after which the LDAP Server Connection is polled. Negative Value disables polling.

Valid value: This parameter takes input any integral value.

Default value is "-1".

LdapPrimaryServerReconnectAttempts

This parameter denotes the number of reconnect attempts LDAP primary server makes in case the LDAP server connection polling is unsuccessful.

Valid value: This parameter takes input any integral value.

Default value is "5".

BackupLdapProviderUrl

This parameter denotes the backup LDAP server provider URL, which is tried when the primary LDAP Server becomes unavailable. See LdapProviderUrl.

Valid value: See LdapProviderUrl.

Default value is "".

BackupLdapProviderDn

This parameter points to the base domain on Backup LDAP Server, under which FMQ would create its repository. See LdapProviderDn.

Valid value: See LdapProviderDn.

Default value is "".

BackupPrincipal

This parameter denotes the backup LDAP Server's user name. This is the user name which would be used to connect to the Backup LDAP Server. See Principal.

Valid value: See Principal.

Default value is "cn=FMQRoot,o=fiorano,c=US".

BackupCredentials

This parameter denotes the backup LDAP Server's login password. See Credentials.

Valid value: See Credentials.

Default value is "secret".

BackupLdapInitialCtxFactory

This parameter denotes the class name for LDAP Server's Initial Context Factory to be supplied to Initial Context. See LdapInitialCtxFactory.

Valid value: See LdapInitialCtxFactory.

Default value is "".

BackupLdapSecurityAuthentication

This parameter denotes the backup LDAP server's security authentication protocol. See LdapSecurityAuthentication.

Valid value: See LdapSecurityAuthentication.

Default value is "simple".

LdapUserClassTop

LdapUserClassTop/LdapUserClassPerson/LdapUserClassOrganizationalPerson/Ldap

UserClassInetOrgPerson store the four object classes used to assign various attributes to any user information stored in an LDAP-provider. These objectClasses are used to assign almost any attributes of the user and are defined by default in the schema of the LDAP-provider. Please note that - When creating a new LDAP entry, you must always specify all of the object classes to which the new entry belongs. Because many directories do not support object class subclassing, you also should always include all of the superclasses of the entry. For example, for an "organizationalPerson" object, you should list in its object classes the "organizationalPerson", "person", and "top" classes.

Valid value: This parameter takes input any valid object class name.

Default value is "top".

LdapUserClassPerson

See LdapUserClassTop.

Valid value: This parameter takes input any valid object class name.

Default value is "Person".

LdapUserClassOrganizationalPerson

See LdapUserClassTop.

Valid value: This parameter takes input any valid object class name.

Default value is "OrganizationalPerson".

LdapUserClassInetorgPerson

See LdapUserClassTop.

Valid value: This parameter takes input any valid object class name.

Default value is "inetOrgPerson".

LdapGroupClassTop

LdapGroupClassTop and LdapGroupClassUniqueNames store the two object classes used to assign various attributes to a Group that are defined in the schema of the LDAP-provider. When creating a new LDAP entry, you must always specify all of the object classes to which the new entry belongs. Because many directories do not support object class subclassing, you also should always include all of the superclasses of the entry. For example, for an "organizationalPerson" object, you should list in its object classes the "organizationalPerson", "person", and "top" classes.

Valid value: This parameter takes input any valid object class name.

Default value is "top".

LdapGroupClassUniqueNames

See LdapGroupClassTop.

Valid value: This parameter takes input any valid object class name.

Default value is "groupofuniquenames".

LdapGroupClassUniqueMember

This value is used to initialize the attribute "uniquemember", of a Group entry, which is used when adding new users to the Group or querying for the members of a group. Suppose it can take the value : cn=something,o=fiorano,c=us. Generally, we assign the value of "LdapProviderDn" which is o=fiorano,c=us by default, for this.

Valid value: LdapGroupClassUniqueMember can take any valid value which can be given as a proper Relative Distinguished Name (RDN).

Default value is "o=fiorano,c=US".

LdapUserNameAttribute

The value of this field can be the common name of an LDAP user (cn), but usually it is an abbreviated string, such as User ID (uid) since we are storing the unique user-identities in LDAP-provider. In that case, the full Distinguished name (DN) for an entry in LDAP structure will be uid=admin,cn=FioranoMQUsers,o=fiorano,c=US.

Valid value: This parameter takes input provider specific valid common name.

Default value is "uid".

LdapUserPasswordAttribute

Attribute ID for the password attribute for the FioranoMQ user stored as a LDAP named object.

Valid value: This parameter takes input provider specific valid common name.

Default value is "userpassword".

LdapGroupNameAttribute

Attribute ID for the group name attribute for the FioranoMQ group stored as a LDAP named object.

Valid value: This parameter takes input provider specific valid common name.

Default value is "cn".

LdapGroupUserNameAttribute

Attribute ID for the user name attribute stored in a group which is stored as a LDAP named object.

Valid value: This parameter takes input provider specific valid common name.

Default value is "uniquemember".

LdapUserDn

Relative distinguished name of an LDAP entry, under the entry specified by the parameter 'LdapProviderDn', under which the FioranoMQ users' information is stored.

Valid value: This parameter takes input any RDN of an existing LDAP entry under LdapProviderDn.

Default value is "cn=FioranoMQUsers".

LdapGroupDn

Relative distinguished name of an LDAP entry, under the entry specified by the parameter 'LdapProviderDn', under which the FioranoMQ groups' information is stored.

Valid value: This parameter takes input any RDN of an existing LDAP entry under LdapProviderDn.

Default value is "cn=FioranoMQGroups".

MaxPasswordLength

Each principal is allotted a password and the MaxPasswordLength parameter enables the configuration of the maximum allowable password length.

Valid value:

The default value of the MaxWordLength is **50**.

The range of integer values in java that are considered valid by the MaxPasswordLength parameter are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero provide the same result.

MaxMemberCount

The MaxMemberCOunt parameter defines the maximum number of members a group may have.

Valid value:

The default value of this parameter is **50**.

The range of integer values in java that are considered valid by the MaxMemberCount parameter are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero provide the same result.

XMLFilePrincipalManager

Following are the parameters present in Fiorano->security->PrincipalManager->XMLFilePrincipalManager

Path

String representing directory path where user information will be stored in XML format.

Default Value: realm/principal

UserFilename

XML File name that contains user information in String format

Default Value: user.xml - The default xml filename

GroupFilename

XML File name that contains Group information in String format

Default Value: group.xml – The default xml file name

PrincipalManagerRDBMS

Following are the parameters present in Fiorano->security->PrincipalManager->PrincipalManager.RDBMS

Url

String representing database URL where user/group information will be store.

Example: jdbc:oracle:thin:@localhost:1521:ORCL for oracle server

Password

Password for RDBMS User in String format- Used for connecting to the RDBMS Server

Username

Database User Name in String format- Used for connecting to the RDBMS Server.

Default Value: sa

DBDriver

String value of JdbcDriver used for RDBMS based datastorage.

Default Value: org.hsqldb.jdbcDriver - The default database driver class name

ConnectionTimeoutInterval

Integer representing database connection open timeout

Default Value: 10000 - The time in milliseconds for connection timeout

Valid Value: Non negative Integer with value zero representing wait forever

ReconnectTimeInterval

Integer representing Time interval (in msec) between two reconnect attempts in disconnected state.

Default Value: 10000 – The time in milliseconds taken by Principal Manager between reconnect attempts

Valid Value: Non negative Integer

EnableReconnect

Boolean value which specifies if Principal Manager has to try to reconnect to database server in case of a connection breakup

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.

StatementFile

String representing Sql statement file which contains the SQLs used by PrincipalManager for Database operations.

Default Value: principalsqlstatements.properties - The default properties used which is located at %FIORANO_HOME%/fmq/profiles/%YOUR_PROFILE%/conf

ConnectionLostErrorMessage

String representing 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.

InitializeDB

Boolean value to enable/disable the initialization of the underlying Database by creating necessary tables on startup if required

Valid values:

- True- Enables initializing the DB
- False - Disables initializing the DB

Default Value: True - Enables initializing the DB.

FileDBManager2

Presented below are the details of the parameters in **FioranoMQ->Fiorano->Security->PrincipalManager->FileDBManager->FileDBManager2**.

Path

The security configuration for the principal is stored in the file based data storage specified by the Path parameter.

Valid value:

The default value of the Path parameter is **SDB/REALM.PRINCIPAL**

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

DeletedThresholdPercent

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

Valid value:

The default value of the DeletedThresholdPercent is 50. This indicates that if the ratio of deleted entries to total entries is equal to or greater than 0.5 the Cache compaction process is invoked.

Note:

- The range of integer values in java considered valid by the DeletedThresholdPercentage value are -2^{32} to $2^{32} - 1$.
- All values less than or equal to zero provide the same result.
- All values greater than or equal to 100 provide the same result.

Example:

Use an example ratio of deleted entries/total entries in the table. If there is good chance that many entries will be marked as deleted then it is best to choose a higher number.

DeletedThresholdCount

A cache stores and manages the list of all tables pertaining to the principal as well as all related information in the file-storage table. After repeated modification of the table, old entries in the table are marked as deleted. This value provides a threshold count of deleted entries for starting the Cache Compaction Process. At the end of the Cache Compact Process the table will not have any deleted entries. Since cache compaction involves CPU intensive File Operations, this flag is to be used along with DeletedThresholdPercent to ensure it is not invoked too often.

Valid value:

The default value of the DeletedThresholdCount is **10**.

The range of integer values considered valid for the DeletedThresholdCount parameter in java are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero provide the same result.

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 of the number of deleted entries needed to trigger a Cache Compaction. A value high enough to trigger a Cache Compaction should be chosen.

DbTableNamePrefix

The database table names prefixed with a string can be modified by the DbTableNamePrefix parameter.

Valid value:

The default value of the DbTableNamePrefix is **#**.

Any String value is a valid parameter.

Example:

If the database table is to be saved in its current state and the the current state is to be saved to be used in the future, modify the prefix to another relevant string and change it back to the original string when the saved state of the database table is to be used.

Chapter 8: Logger Configuration

This chapter explains the configurable parameters related to the FioranoMQ Logger module (i.e., related to server logs or monitoring logs or dashboard logs) in the server.

For information on how to configure FioranoMQ Server Loggers in Offline and online mode, refer to **Chapter 20 Logger** of *FioranoMQ Handbook*.

Introduction

Logging into FioranoMQ is achieved with the help of Log4J which makes it possible to log in at runtime without modifying the application binaries. All log entries are logged at particular log levels. Each logger is given a logging level. The various logger levels are given below.

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	All logged information

Table 1: Log levels

These values are defined so that if given a value **x**, all entries logged with a value less than or equal to **x** will be received by the 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

Loggers are arranged hierarchies depicting parent-child relationships. Each element in the tree (Figure 1) has these two attributes as shown Figure 2, below:



Figure 2: Edit Logger Properties dialog box

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

- If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Message is set to trace (6), all the messages published to a topic along with its corresponding properties are logged.
- If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Publish is set to trace (6), the life cycle of the message, in its various stages of sending, is logged.
- If the 'Log Level' attribute of Fiorano.Services.Queues.Push.Message is set to trace (6), all messages sent to a queue along with its corresponding properties are logged.
- If the 'Log Level' attribute of Fiorano.Services.Queues.Pop is set to trace (6), the life cycle of the message, after it is pushed into the queue, is logged.
- If the 'Log Level' attribute of Fiorano.Services.Topics.MsgFlow.Message is set to trace (6), all the messages published to a topic along with its corresponding properties are logged.
- If the 'Log Level' attribute of Fiorano.Events.TopicPublisher is set to trace (6), all publisher events such as **open** and **close** are logged.
- If the 'Log Level' attribute of Fiorano.Events.QueueReceiver is set to trace (6), all receiver events such as **open** and **close** are logged.
- If the 'Log Level' attribute of Fiorano.Services.Jmx.Operations is set to trace (6), details of all Jmx operations are logged.

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

AppenderAdditive

AppenderAdditive specifies the Appender Additivity for the logger. The output of a log statement of logger, say **C**, will go to all appenders and ancestors of **C**. 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** as well as to all ancestors up to and including **P** but not to the appenders of the ancestors of **P**.

Valid value:

The default value for the AppenderAdditive parameter is **yes**

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

LogLevel

LogLevel is the level of logging. No log event will be generated for a logging statement above this level.

Valid value:

The default value for the LogLevel is **-1**.

Legal values: -1 to 6. (Refer to table 1, above, for more information.)

LogAppender

Presented below are the details of the parameters in **Fiorano->config->LogAppender**

AppenderName

The AppenderName parameter provides a unique Appender Name.

Valid value:

The default value is of the AppenderName parameter is **LogAppender**.

Any String

AppenderType

AppenderType is a parameter that specifies the type of appender to be used. A LogAppender type, for example, decides on the endpoint set to receive logging information.

Valid value:

The default value for the AppenderType parameter is file.

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

Example:

When logged messages qualify, or are important enough, to be displayed on the console, the AppenderType of the console will need to be changed from its file value.

LogPattern

The LogPattern parameter specifies the Log4J format for printing logs. (Please refer to Log4J documentation for more details.)

Valid value:

The default value for the LogPattern Parameter 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 when the log event is generated.

ThresholdLevel

The ThresholdLevel parameter indicates that log events logged above this ThresholdLevellog level will not be received/logged by the Appender.

Valid value:

The default value of the ThresholdLevel parameter is 10.

The range of integer values in java for the ThresholdLevel parameter are -2^{32} to $2^{32} - 1$.

Note:

- All values less than or equal to zero retrieve the same result..
- All values greater than or equal to 10 retrieve the same result.

FilterPattern

The FilterPattern is a regex based string filter pattern for filtering log events received by the Appender. Only Events which matches the filter criteria are logged.

Valid value:

The default value for the FilterPattern parameter is null.

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

Example:

When only the messages that match the criteria defined by the regex are to be logged, the regular expression is to be set.

PrintTarget

The PrintTarget parameter is applicable only to console appenders and the print target can be chosen to reflect the target that the console output prints to.

Valid value:

The default value for the PrintTarget parameter is System.out.

For a value set in the PrintTarget field to have any effect the AppenderType should be set console.

- **System.out** - When appendertype is set to console the console output will print to the standard output stream.
- **System.err** - When appendertype is set to console the console output will print to the standard error output stream.

FileName

The FileName parameter is the name of the log file to which the log events are logged when the AppenderType is **file**.

Valid value:

The values considered valid by the FileName parameter are can be any valid filename. The default value of this parameter is **server.log**

When a relative path for a filename is given a log file is created in **<current-profile>/run/logs/**. If an absolute path for a filename is given a log file is created in its corresponding directory.

IsAppend

The IsAppend parameter option is useful only when the AppenderType is specified as a **file** type. When the server boots and a log file that is specified already exists, this parameter specifies whether the log file that exists overwritten or if the log entries are appended to the existing file.

Valid value:

The default value for the IsAppend Parameter is **yes**

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

MaxBackupIndex

The MaxBackupIndex parameter specifies whether the type of FileAppender used by **FioranoMQ** is a **RollingFileAppender**. The RollingFileAppender can be used to roll log files based on size. When the **FileSize** exceeds **MaxFileSize** the log file will be **rolled**. Rolling consists of the creation of a new log file with an incremental index appended to its filename. The MaxBackupIndex parameter helps specify the maximum number of such files to be created. Example: A **2** will have at most 3 log files. **file.log, file1.log, file2.log**. This parameter may be used only when the **AppenderType** used is file.

Valid value:

The default value for the MaxBackupIndex parameter is **9**.

The range of integer values in java for the MaxBackupIndex parameter are -2^{32} to $2^{32} - 1$.

All values less than or equal to zero retrieve the same result.

Dependencies

Please refer to MaxFileSize and to AppenderType.

MaxFileSize

The MaxFileSize parameter specifies the maximum size of a log file in bytes. Once the maximum size specified is reached the log file is rolled. (For an explanation on RollingFileAppender, refer to **MaxBackupIndex**.) This parameter may be used when the **AppenderType** is a **file**.

Valid value:

The default value for the MaxFileSize parameter is 5000000.

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

Please refer to MaxBackupIndex and to AppenderType.

MaxFilterLevel

The MaxFilterLevel parameter specifies the Maximum Filter level for log events. Events logged above the log level specified by this parameter will not be received/logged by the Appender.

Valid value:

The range of integer values considered valid for the MaxFilterLevel parameter in java are - 2^{32} to $2^{32} - 1$. The default value of this parameter is **10**.

All values less than or equal to zero retrieve the same result.

MinFilterLevel

The MinFilterLevel parameter specifies the Minimum Filter level for log events. Events logged below the log level specified by this parameter will not be received/logged by the Appender.

Valid value:

The range of integer values considered valid in java for the MinFilterLevel parameter are - 2^{32} to $2^{32} - 1$. The default value for this parameter is **1**.

All values less than or equal to retrieve the same result.

LogAppender

Monitoring->config->LogAppender

AppenderName

The AppenderName parameter provide a unique Appender Name.

Valid value:

Any String. The default value for the AppenderName parameter is **LogAppender**.

AppenderType

The AppenderType parameter specifies the type of appender to be used. The **LogAppender Type** decides the endpoint that will receive the logging information.

Valid values:

The default value for the AppenderType is **file**.

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

Example:

When the logged messages qualify or are important enough to be displayed on the console, the AppenderType to console must be changed from its default value of **file**.

LogPattern

The LogPattern parameter specifies the Log4J format for printing logs. (Please refer to Log4J documentation for more details.)

Valid value:

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

Example:

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

ThresholdLevel

Log events logged above the ThresholdLevel parameter of the log level will not be received/logged by the Appender.

Valid values:

The range of integer values that are considered valid by the ThresholdLevel Parameter in java are -2^{32} to $2^{32} - 1$. The default value for this parameter is 10.

Note:

- All values less than or equal to zero retrieve the same result.
- All values greater than or equal to 10 retrieve the same result.

FilterPattern

The FilterPatter parameter is a regex based string filter pattern for filtering log events received by the Appender. Only events which matches the filter criteria are logged.

Valid values:

The default value for the FilterPattern is null. Values considered valid by this parameter include all valid regex expression.

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

Example:

When only the messages that match the criteria defined by the regex are to be logged, the appropriate regular expression needs to be set.

PrintTarget

The PrintTarget parameter is applicable only to console appenders. The print target can be chosen to reflect the target the console output is to be printed to.

Valid values:

The default value is of the PrintTarget parameter is **System.out**.

For this parameter to work, the **AppenderType** should be set to **console**.

- **System.out** - When appendertype is set to console, the console output is printed to the standard output stream.
- **System.err** - When appendertype is set to console, the console output is printed to the standard error output stream.

Dependencies:

AppenderType

FileName

The FileName parameter specifies the name of the log file to which the log events are logged if the AppenderType is a **file**.

Valid values:

The default value of the FileName is **server.log**

Any valid filename is considered a valid value by the FileName parameter.

If a relative path for a filename is given, the log file is created in the directory relative to the file name specified: <current-profile>/run/logs/.if an absolute path for a filename is given, the log file is created in its corresponding directory.

Dependencies:

AppenderType

IsAppend

The IsAppend parameter is used only when the AppenderType is specified as **file**. If the log file specified already exists, this option specifies whether this log file will be overwritten or whether the log entries will be appended.

Valid values:

The default value of the IsAppend parameter is **yes**.

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

MaxBackupIndex

The MaxBackupIndex parameter specifies whether the **type** of **FileAppender** used by FioranoMQ is a RollingFileAppender. The RollingFileAppender can be used to roll log files based on size. When the filesize exceeds MaxFileSize the log file will be rolled. A new log file will be created which will have an incremental index appended to its filename. This parameter helps specify the maximum number of files to be created and will have at most 3 log files: file.log, file1.log, file2.log. This may be used only when the **AppenderType** is a **file**.

Valid values:

The range of integer values considered valid by this parameter in java are -2^{32} to $2^{32} - 1$. The default value of this parameter is **4**.

All values less than or equal to zero retrieve the same result.

Dependencies:

MaxFileSize and AppenderType.

MaxFileSize

The MaxFileSize parameter specifies the maximum size of a log file in bytes. Once a log file reaches this size it is rolled (For an explanation on RollingFileAppender see MaxBackupIndex.) This parameter may be used only when the AppenderType is a **file**.

Valid values:

The range of integer values considered valid by the MaxFileSize parameter in java are -2^{32} to $2^{32} - 1$. The default value of this parameter is **1000000**

All values less than or equal to zero retrieve the same results.

Dependencies:

MaxBackupIndex and AppenderType.

MaxFilterLevel

The MaxFilterLevel parameter specifies the maximum filter level for log events. Events logged above this log level will not be received/logged by the Appender.

Valid values:

The range of integer values considered valid by the MaxFilterLevel parameter in java are -2^{32} to $2^{32} - 1$. The default value for this parameter is **10**.

All values less than or equal to zero retrieve the same result.

MinFilterLevel

The MinFilterLevel parameter specifies the minimum filter level for log events. Events logged below this log level will not be received/logged by the Appender.

Valid values:

The range of integer values that are considered valid by the MinFilterLevel parameter in java are -2^{32} to $2^{32} - 1$. The default value for this parameter is **1**

All values less than or equal to zero retrieve the same result.

Chapter 9: HA Replicated Configuration

This chapter explains the parameters that can be configured for FioranoMQ Servers running in Replicated High Availability mode. These are the various parameters that are configured under the FioranoMQ→Fiorano→HA→* tree node in Replicated HA Profiles in Offline configuration using Studio.

For more information on High Availability, refer to the *FioranoMQ High Availability Guide*

Primary Server Configuration

FioranoHAConnectionManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**.

Port

The Port parameter specifies the port on which the HA Manager listens on for connections from the peer HA Server. This parameter listens for a connection from HA Peer Server in order to know the status of its peer.

Valid values:

The default value of the Port parameter is **2000**.

The range of integer values considered valid by the Port parameter in java are -2^{31} to $2^{31}-1$.

Note: The port may not be used by other services when being used to listen to Peer servers..

NagleAlgo

The NagleAlgo parameter is a boolean value that determines whether Nagle's Algorithm is enabled during 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 together, all at once. If 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 worth of output, so as to send the entire output in one go.

Valid values:

The default value for the NagleAlgo parameter is **no**.

- **yes** – Indicates that the NagleAlgo is enabled.

- **no** – Indicates that the NagleAlgo is not enabled.

Example:

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

SocketInitializationTimeout

The SocketInitializationTimeout parameter indicates the time in milliseconds within which a client socket should identify itself and exchange its version number with the Server. The version number exchange is an authentication mechanism that FioranoMQ employs.

Valid values:

The default value for the SocketInitializationTimeout parameter is **60000** (which is equal to one minute).

The range of integer values considered valid by the SocketInitializationTimeout parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

RealmStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->RealmStubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MinServiceID parameter is null. If the value is specified as null, there is no range defined for serviceIDs.

The range of integer values considered valid by the MinServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs.

The range of integer values considered valid by the MaxServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

QGMSSubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->QGMSSubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ with each request associated with a subsystem. The range of serviceIDs associated with a subsystem defines the operations that are accessible to that subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs.

The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

TGMSStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->TGMSStubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs.

The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

SyncRealmStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->SyncRealmStubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs..

The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SyncNamingStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncNamingStubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

NamingStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->NamingStubManager**

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SyncQueueStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HServiceManager->HSubSystems->SyncQueueStubManager**

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for the serviceIDs. The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

RpStateStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HServiceManager->HSubSystems->RpStateStubManager**.

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value of the MinServiceID is 2001.

The range of integer values considered valid by the MinServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

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:

The default value of the MaxServiceID is 3000.

The range of integer values considered valid by the MaxServiceID in java are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

FioranoRpRealmManager

Presented below are the details for the parameters 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 (ACL based) information a zip file containing the ACL information present in this server must be created. The AclZipPath parameter allows the configuration of the path where this file is located.

Valid values:

The default value of the AclZipPath is **fmq_acl.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that 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 (ACL based) information a zip file is created using the ACL information. The AclUnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value of the AclUnzipDir parameter is **ACL_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that 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 (User based) information a zip file is created using the information related to the principal present in the Server. The PrincipalZipPath parameter allows the configuration of the path in where this file is located.

Valid values:

The default value of the PrincipalZipPath parameter is **fmq_principal.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that 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 (User based) information a zip file is created using the information related to the principal. The PrincipalUnzipDir parameter allows the configuration of the path where this zip file is temporarily unzipped.

Valid values:

The default value of the PrincipalUnzipDir parameter is **PRINCIPAL_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of the principal and realm related information must be done within this time.

Valid values:

The default value of the ScoketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout parameter are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

FioranoRpNamingManager

Presented below are the details of the parameters 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 name related information a zip file using naming related information present in the Server is created. The ZipPath parameter allows the configuration of the path where this file is located.

Valid values:

The default value of the ZipPath parameter is **fmq_nmdb.zip**.

Any valid file name for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that 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 name related information, a zip file using naming related information is created. The UnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value for the UnzipDir parameter is **NM_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeOut parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of all naming related information must be done within this time.

Valid values:

The default value for the SocketTimeOut parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAKRPCProvider

Presented below are details of the parameters in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**.

BackupHAPort

The BackupHAPort parameter specifies the port of the backup peer Server on which the peer is listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAPort parameter is **3000**.

Note: It is mandatory to have the BackupHAPort parameter to run HA.

BackupHAIPAddress

The BackupHAIPAddress parameter specifies the IP address of the backup peer Server on which the peer listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAIPAddress parameter is **localhost**.

Note: It is mandatory to have the BackupHAIPAddress parameter to run HA.

SocketTimeout

The SocketTimeOut parameter indicates the time in milliseconds before the socket created for communication with the peer Server timesout occurs.

Valid values:

The default value for the SocketTimeOut parameter is **30000**.

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **30000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

HaStateNotifBroadcaster

Presented below are the details of the parameters in **Fiorano->HA->HAManager->EventManager->HaStateNotifBroadcaster**.

LoggerName

The LoggerName parameter provides a unique Logger Name.

Valid values:

The default value of the LoggerName parameter is **null**.

Any String may be used as a valid value for this parameter.

FioranoStatusPersister

Presented below are the details of the parameters 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 of the replicated is done by the **StatusPersister**. The Path parameter helps configure the path of file where the status that is persisted is located.

Valid values:

The default value of the Path parameter is **haStatus.dat**.

Any valid file name may be used by this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that directory.

FioranoHAManager

Presented below are the details of the parameters in **Fiorano->HA->HAManager->RpManager->FioranoHAManager->config**.

Primary

The Primary parameter is a boolean parameter that indicates whether the acting Server acts as the primary HA Server when in the Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server while the other is the secondary Server. For more information about FioranoMQ's HA implementation, please refer to *FioranoMQ Concepts Guide Chapter 16* or *FioranoMQ High Availability Guide*.

Valid values:

The default value of the Primary parameter is **yes**.

yes - Indicates that the Server is the primary server.

no - Indicates that the Server is the secondary server.

GatewayListenerNotUpMesg

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

The GatewayListenerNotUpMesg parameter helps identify the message that will be displayed when the acting Server finds that no listener is present on the port configured.

Valid values:

The default value for the GatewayListenerNotUpMesg parameter is **Connection refused**.

Any String may be used as a valid value for this parameter.

PingInterval

The acting HA Server pings its peer Server intermittently to exchange status information. PingInterval is the time interval in milliseconds after which the remote Server is pinged in Replication HA mode.

Valid values:

The default value for the PingInterval parameter is **30000**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

GatewayServerPort

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

The GatewayServerPort parameter helps specify the port on which the Gateway machine listens for incoming requests.

Valid values:

The default value GatewayServerPort parameter is **7**.

The range of integer values considered valid by the GatewayServerPort are -2^{31} to $2^{31} - 1$.

Note: The port should not be in use..

SocketCreationTimeout

SocketCreationTimeout is the time in milliseconds within which the socket has to connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **10000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

ServerName

The ServerName parameter specifies the name of the Server running in the Replication HA mode.

Valid values:

The default value for the ServerName is **FMQServer**

Any String may be used as a valid value for this parameter.

Example:

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

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

MaxWaitTimeout

When an acting Server notices that there has been a change in its peer Server state it must change its state appropriately. While the acting Server begins changing its state it starts a timer. If for some reason the state change is not completed after a reasonable amount of time the acting server must be altered.

The MaxWaitTimeout parameter specifies the maximum waiting timeout in milliseconds the local Server waits for a state transition to complete.

Valid values:

The default value for the MaxWaitTimeout parameter is **30000**.

The range of integer values considered valid by the MaxWaitTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

WaitTimeForBackupServer

This parameter indicates the wait time to detect backup server during local server bootup. If backup server is not available then local server will switch to standalone after acquiring lock. A higher value of this parameter can avoid one of the server switching to standalone, thus avoiding unnecessary database replication at the cost of more wait time during server boot up.

Default value:

30000(units in milliseconds)

Legal values:

Any positive integer which represents a few seconds.

MaxPingCount

This parameter indicates the maximum number of times gateway server will be pinged for detecting a server failure. This will be useful in case of a weak network link between gateway server and ha server. Increasing this parameter should be done after consulting Fiorano support team as increasing this parameter can disrupt ha functionality.

Default value: 1

FioranoFMQKernel

Presented below are the details for the parameters in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**.

MQUnDeploymentLists

While in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed.

These components are listed in FioranoMQ specified files. The MQUnDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be undeployed when necessary.

Valid values:

The default value for the MQUnDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalUnDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connection to the Server when it is in the Passive mode are refused. If a peer Server goes dead the Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. When the Server is in the Active state these components need to be undeployed.

The components that need to be undeployed are listed in FioranoMQ specified files. The ExternalUnDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be undeployed when necessary.

Valid values:

The default value for the ExternalUnDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

MQDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead then the acting Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. These components are redeployed when the Server starts accepting connections.

The components that need to be redeployed are listed in FioranoMQ specified files. The MQDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be redeployed when necessary.

Valid values

The default value MQDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. In case a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed. There may be external components that need to be redeployed when the Server goes back to the Active state.

The components to be undeployed are listed in Fiorano files. The ExternalDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be redeployed when necessary.

Valid values:

The default value for the ExternalDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

FioranoRpTopicManager

Presented below are the details for the parameters 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. To be able to synchronize all topic related information a zip file containing topic related information present in the Server is created. The ZipPath parameter allows the configuration of the path in which that file is located.

Valid values:

The default value for the ZipPath parameter is **fmq_tgmsdb.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: `<current-profile>/run/`. If its an absolute path for a file name is given, the zip file is created in that directory.

UnzipDir

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

Valid values

The default value of the UnZipDir parameter is **TGMS_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: `<current-profile>/run/`. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for the replicable component occurs. Replication of all topic related information must be done within this time.

Valid values:

The default value for the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$ All values less than zero retrieve the same result.

FioranoRpQueueManager

Presented below are the details for the parameters 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. To be able to synchronize all the queue related information a zip file containing queue related information present in the Server is created. The ZipPath parameter allows the configuration of the path in which that file is located.

Valid values:

The default value for the ZipPath parameter is **fmq_qgmsdb.zip**.

Any valid filename for a zip file may be used as a parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

UnzipDir

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

Valid values

The default value for the UnzipDir parameter is **QGMS_TEMP**.

Any valid filename may be used as a parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout indicates the time in milliseconds before the socket timesout for the replicable component occurs. Replication of all queue related information must be done within this time.

Valid values

The default value for the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$

All values less than zero retrieve the same result.

Secondary Server Configuration

FioranoHAConnectionManager

Presented below are the details of the parameters in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**.

Port

The Port specifies the port on which HA Manager listens for connections from the peer HA Server. In addition the HA Server listens for a connection from its HA peer Server to determine the status of its peer on this port. The Port parameter is used specifies this port.

Valid values

The default value for the Port parameter is **3000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$

Note: The port specified should not be in use..

NagleAlgo

The NagleAlgo is a boolean parameter that determines whether Nagle's Algorithm, is enabled at the time of socket creation with the 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 together. If there is a sent packet for which the sender has received no acknowledgment, the sender output needs to be buffered until it there is a full packet worth of output which can then be sent in one go.

Valid values:

The default value for the NagleAlgo is **no**.

- **yes** – Indicates that the NagleAlgo is enabled.
- **no** – Indicates that 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 SocketInitializationTimeout parameter indicates the time in milliseconds within which a client socket should identify itself and exchange its version number with the Server. The version number exchange is an authentication mechanism employed by FioranoMQ.

Valid values:

The default value of the SocketInitializationTimeout is **60000** (equal to one minute).

The range of integer values considered valid by the SocketInitializationTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

SyncTopicStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->SyncTopicStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs.

The range of integer values considered valid by the MinServiceID are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

RealmStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->RealmStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

QGMSSubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->QGMSSubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

TGMSStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->TGMSStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs.

The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SyncRealmStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HSubSystems->SyncRealmStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SyncNamingStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncNamingStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are - 2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

NamingStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->NamingStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs.

The range of integer values considered valid by the MinServiceID are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SyncQueueStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HAServiceManager->HASubSystems->SyncQueueStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values

The default value for the MinServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MinServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MaxServiceID is null. A null value indicates that there is no defined range of serviceIDs. The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

RpStateStubManager

Presented below are the details for the parameters present in **Fiorano->HA->HAConnectionManager->HServiceManager->HSubSystems->RpStateStubManager**.

MinServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MinServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is 2001.

The range of integer values considered valid by the MinServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

MaxServiceID

FioranoMQ generates a serviceID with each request associated within a subsystem. The range of serviceIDs that are associated with a subsystem define the operations that are accessible to that subsystem. The MaxServiceID value defines the minimum value of that range.

Valid values:

The default value for the MinServiceID is 3000.

The range of integer values considered valid by the MaxServiceID are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

FioranoRpRealmManager

Presented below are the details for the parameters 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 (ACL based) information a zip file containing the ACL information present in this server must be created. The AclZipPath parameter allows the configuration of the path where this file is located.

Valid values:

The default value of the AclZipPath is **fmq_acl.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that 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 (ACL based) information a zip file is created using the ACL information. The AclUnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value of the AclUnzipDir parameter is **ACL_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that 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 (User based) information a zip file is created using the information related to the principal present in the Server. The PrincipalZipPath parameter allows the configuration of the path in where this file is located.

Valid values:

The default value of the PrincipalZipPath parameter is **fmq_principal.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: **<current-profile>/run/**. If its an absolute path for a file name is given, the zip file is created in that 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 (User based) information a zip file is created using the information related to the principal. The PrincipalUnzipDir parameter allows the configuration of the path where this zip file is temporarily unzipped.

Valid values:

The default value of the PrincipalUnzipDir parameter is **PRINCIPAL_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of the principal and realm related information must be done within this time.

Valid values:

The default value of the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout parameter are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

FioranoRpNamingManager

Presented below are the details for the parameters 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 name related information a zip file using naming related information present in the Server is created. The ZipPath parameter allows the configuration of the path where this file is located.

Valid values:

The default value of the ZipPath parameter is **fmq_nmdb.zip**.

Any valid file name for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that 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 name related information, a zip file using naming related information is created. The UnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value for the UnzipDir parameter is **NM_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of the principal and realm related information must be done within this time.

Valid values:

The default value of the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout parameter are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

FioranoHAKRPCProvider

Presented below are the details for the parameters in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**.

BackupHAPort

The BackupHAPort parameter specifies the port of the backup peer Server on which the peer is listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAPort parameter is **2000**.

Note: It is mandatory to have the BackupHAIPort parameter to run HA.

BackupHAIPAddress

The BackupHAIPAddress parameter specifies the IP address of the backup peer Server on which the peer listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAIPAddress parameter is **localhost**.

Note: It is mandatory to have the BackupHAIPAddress parameter to run HA.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket created for communication with the peer Server times out occurs.

Valid values:

The default value for the SocketTimeout parameter is **30000**.

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **30000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

HaStateNotifBroadcaster

Presented below are the details for the parameters in **Fiorano->HA->HAManager->EventManager->HaStateNotifBroadcaster**.

LoggerName

The LoggerName parameter provides a unique Logger Name.

Valid values:

The default value of the `LoggerName` parameter is **null**.

Any String may be used as a valid value for this parameter.

FioranoStatusPersister

Presented below are the details for the parameters 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 of the replicated is done by the **StatusPersister**. The `Path` parameter helps configure the path of file where the status that is persisted is located.

Valid values:

The default value of the `Path` parameter is **haStatus.dat**.

Any valid file name may be used by this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: `<current-profile>/run/`. If its an absolute path for a file name is given, the zip file is created in that directory.

FioranoHAManager

Presented below are the details for the parameters in **Fiorano->HA->HAManager->RpManager->FioranoHAManager->config**.

Primary

The `Primary` parameter is a boolean parameter that indicates whether the acting Server acts as the primary HA Server when in the Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server while the other is the secondary Server. For more information about FioranoMQ's HA implementation, please refer to *FioranoMQ Concepts Guide Chapter 16* or *FioranoMQ High Availability Guide*.

Valid values:

The default value of the `Primary` parameter is **no**.

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

GatewayListenerNotUpMesg

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

The GatewayListenerNotUpMesg parameter helps identify the message that will be displayed when the acting Server finds that no listener is present on the port configured.

Valid values:

The default value for the GatewayListenerNotUpMesg parameter is **Connection refused**.

Any String may be used as a valid value for this parameter.

PingInterval

The acting HA Server pings its peer Server intermittently to exchange status information. PingInterval is the time interval in milliseconds after which the remote Server is pinged in Replication HA mode.

Valid values:

The default value for the PingInterval parameter is **30000**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

GatewayServerPort

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

The GatewayServerPort parameter helps specify the port on which the Gateway machine listens for incoming requests.

Valid values:

The default value GatewayServerPort parameter is **7**.

The range of integer values considered valid by the GatewayServerPort are -2^{31} to $2^{31} - 1$.

Note: The port should not be in use..

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **10000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

ServerName

The ServerName parameter specifies the name of the Server running in the Replication HA mode.

Valid values:

The default value for the ServerName is **FMQServer**

Any String may be used as a valid value for this parameter.

Example:

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

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

MaxWaitTimeout

When an acting Server notices that there has been a change in its peer Server state it must change its state appropriately. While the acting Server begins changing its state it starts a timer. If for some reason the state change is not completed after a reasonable amount of time the acting server must be altered.

The MaxWaitTimeout parameter specifies the maximum waiting timeout in milliseconds the local Server waits for a state transition to complete.

Valid values:

The default value for the MaxWaitTimeout parameter is **30000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

FioranoFMQKernel

Presented below are the details for the parameters in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**.

MQUnDeploymentLists

While in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed.

These components are listed in FioranoMQ specified files. The MQUnDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be undeployed when necessary.

Valid values:

The default value for the MQUnDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalUnDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead the Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. When the Server is in the Active state these components need to be undeployed.

The components that need to be undeployed are listed in FioranoMQ specified files. The ExternalUnDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be undeployed when necessary.

Valid values:

The default value for the ExternalUnDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

MQDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead then the acting Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. These components are redeployed when the Server starts accepting connections.

The components that need to be redeployed are listed in FioranoMQ specified files. The MQDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be redeployed when necessary.

Valid values

The default value MQDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. In case a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed. There may be external components that need to be redeployed when the Server goes back to the Active state.

The components to be undeployed are listed in Fiorano files. The ExternalDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be redeployed when necessary.

Valid values:

The default value for the ExternalDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

FioranoRpTopicManager

Presented below are the details for the parameters 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. To be able to synchronize all topic related information a zip file containing topic related information present in the Server is created. The ZipPath parameter allows the configuration of the path in which that file is located.

Valid values:

The default value for the ZipPath parameter is **fmq_tgmsdb.zip**.

Any valid filename for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in the 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 name related information, a zip file using naming related information is created. The UnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value for the UnzipDir parameter is **TGMS_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of the principal and realm related information must be done within this time.

Valid values:

The default value of the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout parameter are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

FioranoRpQueueManager

Presented below are the details for the parameters 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 name related information a zip file using naming related information present in the Server is created. The ZipPath parameter allows the configuration of the path where this file is located.

Valid values:

The default value of the ZipPath parameter is **fmq_qgmsdb.zip**.

Any valid file name for a zip file may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that 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 name related information, a zip file using naming related information is created. The UnzipDir parameter allows the configuration of the path where the zip file is temporarily unzipped.

Valid values:

The default value for the UnzipDir parameter is **QGMS_TEMP**.

Any valid filename may be used for this parameter.

Given a relative path for a zip file name, the zip file is created relative that path in the directory: <current-profile>/run/. If its an absolute path for a file name is given, the zip file is created in that directory.

SocketTimeout

The SocketTimeout parameter indicates the time in milliseconds before the socket times out for this replicable component occurs. Replication of the principal and realm related information must be done within this time.

Valid values:

The default value of the SocketTimeout parameter is **240000** (equal to four minutes).

The range of integer values considered valid by the SocketTimeout parameter are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

Chapter 10: HA Shared Configuration

This chapter explains the parameters that can be configured for FioranoMQ Servers running in Shared High Availability mode. These are the various parameters that are configured under the **FioranoMQ→Fiorano→HA→*** tree node in Shared HA Profiles in Offline configuration using Studio.

For more information on High Availability, refer to the FioranoMQ High Availability Guide

Primary Server Configuration

FioranoHAKRPCProvider

Presented below are the details for the parameters in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

BackupHAPort

The BackupHAPort parameter specifies the port of the backup peer Server on which the peer is listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAPort parameter is **3000**.

Note: It is mandatory to have the BackupHAPort parameter to run HA.

BackupHAIPAddress

The BackupHAIPAddress parameter specifies the IP address of the backup peer Server on which the peer listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAIPAddress parameter is **localhost**.

Note: It is mandatory to have the BackupHAIPAddress parameter to run HA.

SocketTimeout

The SocketTimeOut parameter indicates the time in milliseconds before the socket created for communication with the peer Server times out occurs.

Valid values:

The default value for the SocketTimeOut parameter is **30000**.

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **30000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAConnectionManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

Port

The Port parameter specifies the port on which the HA Manager listens on for connections from the peer HA Server. This parameter listens for a connection from HA Peer Server in order to know the status of its peer.

Valid values:

The default value of the Port parameter is **2000**.

The range of integer values considered valid by the Port parameter in java are -2^{31} to $2^{31} - 1$.

Note: The port may not be used by other services when being used to listen to Peer servers.

NagleAlgo

The NagleAlgo parameter is a boolean value that determines whether Nagle's Algorithm is enabled during 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 together, all at once. If 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 worth of output, so as to send the entire output in one go.

Valid values:

The default value for the NagleAlgo parameter is **no**.

- **yes** – Indicates that the NagleAlgo is enabled.
- **no** – Indicates that the NagleAlgo is not enabled.

Example:

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

SocketInitializationTimeout

The SocketInitializationTimeout parameter indicates the time in milliseconds within which a client socket should identify itself and exchange its version number with the Server. The version number exchange is an authentication mechanism that FioranoMQ employs.

Valid values:

The default value for the SocketInitializationTimeout parameter is **60000** (which is equal to one minute).

The range of integer values considered valid by the SocketInitializationTimeout parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAStateStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HAServiceManager->HAStateStubManager->FioranoHAStateStubManager->config**

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MinServiceID parameter is null. If the value is specified as null, there is no range defined for serviceIDs.

The range of integer values considered valid by the MinServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for serviceIDs.

The range of integer values considered valid by the MaxServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAManager

Presented below are the details for the parameters in **Fiorano->HA->HAManager->FioranoHAManager->config**

Primary

The Primary parameter is a boolean parameter that indicates whether the acting Server acts as the primary HA Server when in the Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server while the other is the secondary Server. For more information about FioranoMQ's HA implementation please refer to *FioranoMQ Concepts Guide Chapter 16* or *FioranoMQ High Availability Guide*.

Valid values:

The default value of the Primary parameter is **yes**.

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

GatewayListenerNotUpMesg

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

The GatewayListenerNotUpMesg parameter helps identify the message that will be displayed when the acting Server finds that no listener is present on the port configured.

Valid values:

The default value for the GatewayListenerNotUpMesg parameter is **Connection refused**.

Any String may be used as a valid value for this parameter.

GatewayServerPort

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

The GatewayServerPort parameter helps specify the port on which the Gateway machine listens for incoming requests.

Valid values:

The default value GatewayServerPort parameter is **7**.

The range of integer values considered valid by the GatewayServerPort are -2^{31} to $2^{31} - 1$.

Note: The port should not be in use..

PingInterval

The acting HA Server pings its peer Server intermittently to exchange status information. PingInterval is the time interval in milliseconds after which the remote Server is pinged in Replication HA mode.

Valid values:

The default value for the PingInterval parameter is **30000**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **10000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

ServerName

The ServerName parameter specifies the name of the Server running in the Replication HA mode.

Valid values:

The default value for the ServerName is **FMQServer**

Any String may be used as a valid value for this parameter.

Example:

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

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

FioranoFMQKernel

Presented below are the details for the parameters in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

While in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed.

These components are listed in FioranoMQ specified files. The MQUnDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be undeployed when necessary.

Valid values:

The default value for the MQUnDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalUnDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connection to the Server when it is in the Passive mode are refused. If a peer Server goes dead the Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. When the Server is in the Active state these components need to be undeployed.

The components that need to be undeployed are listed in FioranoMQ specified files. The ExternalUnDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be undeployed when necessary.

Valid values:

The default value for the ExternalUnDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

MQDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead then the acting Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. These components are redeployed when the Server starts accepting connections.

The components that need to be redeployed are listed in FioranoMQ specified files. The MQDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be redeployed when necessary.

Valid values

The default value MQDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. In case a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed. There may be external components that need to be redeployed when the Server goes back to the Active state.

The components to be undeployed are listed in Fiorano files. The ExternalDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be redeployed when necessary.

Valid values:

The default value for the ExternalDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

Secondary Server Configuration

FioranoHAKRPCProvider

Presented below are the details for the parameters in **Fiorano->HA->HAKRPCProvider->FioranoHAKRPCProvider->config**

BackupHAPort

The BackupHAPort parameter specifies the port of the backup peer Server on which the peer is listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAPort parameter is **2000**.

Note: It is mandatory to have the BackupHAPort parameter to run HA.

BackupHAIPAddress

The BackupHAIPAddress parameter specifies the IP address of the backup peer Server on which the peer listens for status requests sent by the Server.

Valid values:

The default value of the BackupHAIPAddress parameter is **localhost**.

Note: It is mandatory to have the BackupHAIPAddress parameter to run HA.

SocketTimeout

The SocketTimeOut parameter indicates the time in milliseconds before the socket created for communication with the peer Server times out occurs.

Valid values:

The default value for the SocketTimeOut parameter is **30000**.

The range of integer values considered valid by the SocketTimeout are -2^{31} to $2^{31} - 1$

All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **30000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAConnectionManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->FioranoHAConnectionManager->config**

Port

The Port parameter specifies the port on which the HA Manager listens on for connections from the peer HA Server. This parameter listens for a connection from HA Peer Server in order to know the status of its peer.

Valid values:

The default value of the Port parameter is **3000**.

The range of integer values considered valid by the Port parameter in java are -2^{31} to $2^{31}-1$.

Note: The port may not be used by other services when being used to listen to Peer servers.

NagleAlgo

The NagleAlgo parameter is a doolean value that determines whether Nagle's Algorithm is enabled during 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 together, all at once. If 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 worth of output, so as to send the entire output in one go.

Valid values:

The default value for the NagleAlgo parameter is **no**.

- **yes** – Indicates that the NagleAlgo is enabled.
- **no** – Indicates that the NagleAlgo is not enabled.

Example:

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

SocketInitializationTimeout

The SocketInitializationTimeout parameter indicates the time in milliseconds within which a client socket should identify itself and exchange its version number with the Server. The version number exchange is an authentication mechanism that FioranoMQ employs.

Valid values:

The default value for the SocketInitializationTimeout parameter is **60000** (which is equal to one minute).

The range of integer values considered valid by the SocketInitializationTimeout parameter in java are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

FioranoHAStateStubManager

Presented below are the details for the parameters in **Fiorano->HA->HAConnectionManager->HServiceManager->HAStateStubManager->FioranoHAStateStubManager->config**

MinServiceID

The MinServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MinServiceID defines the minimum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MinServiceID parameter is null. If the value is specified as null, there is no range defined for serviceIDs.

The range of integer values considered valid by the MinServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

MaxServiceID

The MaxServiceID parameter defines a serviceID in FioranoMQ. Each request is associated with a subsystem. A range of serviceIDs associated with a subsystem determine the operations accessible to the subsystem. The MaxServiceID defines the maximum value of the range that exists for the serviceIDs.

Valid values:

The default value for the MaxServiceID parameter is null. If the value is specified as null, there is no range defined for serviceIDs.

The range of integer values considered valid by the MaxServiceID parameter in java are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

FioranoHAManager

Presented below are the details for the parameters in **Fiorano->HA->HAManager->FioranoHAManager->config**

Primary

The Primary parameter is a boolean parameter that indicates whether the acting Server acts as the primary HA Server when in the Replication HA mode. In FioranoMQ's HA implementation, one of the two HA Servers is the primary server while the other is the secondary Server. For more information about FioranoMQ's HA implementation, please refer to *FioranoMQ Concepts Guide Chapter 16* or *FioranoMQ High Availability Guide*.

Valid values:

The default value of the Primary parameter is **no**.

- **yes** - Indicates that the Server is the primary server.
- **no** - Indicates that the 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 acting HA Server pings its peer Server intermittently to exchange status information. PingInterval is the time interval in milliseconds after which the remote Server is pinged in Replication HA mode.

Valid values:

The default value for the PingInterval parameter is **30000**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$. All values less than zero retrieve the same result.

SocketCreationTimeout

The SocketCreationTimeout indicates the time in milliseconds within which socket must connect with the peer Server failing which the socket creation attempt fails.

Valid values:

The default value for the SocketCreationTimeout is **10000**.

The range of integer values considered valid by the SocketCreationTimeout are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

ServerName

The ServerName parameter specifies the name of the Server running in the Replication HA mode.

Valid values:

The default value for the ServerName is **FMQServer**

Any String may be used as a valid value for this parameter.

Example:

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

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

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 the enterprise Server is deployed as the Gateway machine for HA.

The GatewayServerIPAddress parameter helps specify the IP address on which the Gateway machine listens for incoming requests.

Valid values:

The default value for the GatewayServerIPAddress parameter is **localhost**.

Example:

Choose a machine that is least expected to be out of the network to ensure that it is most likely to perform its function as a Gateway machine.

FioranoFMQKernel

Presented below are the details for the parameters in **Fiorano->HA->FMQKernel->FioranoFMQKernel->config**

MQUnDeploymentLists

While in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed.

These components are listed in FioranoMQ specified files. The MQUnDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be undeployed when necessary.

Valid values:

The default value for the MQUnDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalUnDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connection to the Server when it is in the Passive mode are refused. If a peer Server goes dead the Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. When the Server is in the Active state these components need to be undeployed.

The components that need to be undeployed are listed in FioranoMQ specified files. The ExternalUnDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be undeployed when necessary.

Valid values:

The default value for the ExternalUnDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

MQDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. If a peer Server goes dead then the acting Server goes into the Standalone state. In order to **not accept any client connections** while in passive state, a few components of the Server need to be undeployed. These components are redeployed when the Server starts accepting connections.

The components that need to be redeployed are listed in FioranoMQ specified files. The MQDeploymentLists parameter helps specify lists of files (separated by commas) containing Fiorano specific components that are to be redeployed when necessary.

Valid values

The default value MQDeploymentLists parameter is **kernel.lst,DefaultMQObjects.lst**

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in that directory.

ExternalDeploymentLists

When in the Active mode the HA Server works normally. While in the Passive mode, the HA Server only monitors its peer and does not handle any client requests. All client connections to the Server when it is in the Passive mode are refused. In case a peer Server goes dead the acting Server goes into the Standalone state. In order to **not accept any client connections** while in the Passive state, a few components of the Server need to be undeployed. There may be external components that need to be redeployed when the Server goes back to the Active state.

The components to be undeployed are listed in Fiorano files. The ExternalDeploymentLists parameter helps specify lists of files (separated by commas) containing external components that are to be redeployed when necessary.

Valid values:

The default value for the ExternalDeploymentLists parameter is null.

A comma separated list of .lst files is a valid value for this parameter.

Given a relative path for a file name the files are searched for in the directory: <current-profile>/deploy/. If an absolute path is given files are searched in the directory.

Chapter 11: Dispatcher Configuration

This chapter explains the parameters that can be configured for the FioranoMQ Dispatcher which is used for dispatching incoming connections for load-balancing purposes. These are the various parameters that are configured under the **FioranoMQ→Fiorano→Dispatcher→*** tree node in the Dispatcher Profile in Offline configuration using Studio.

For more information on FioranoMQ Dispatcher, refer to **Chapter 10 Dispatcher** of *FioranoMQ Handbook*.

FioranoJobManager2

Presented below are the details for the parameters present in **Fiorano->JobManager->Impl2->FioranoJobManager2->config**.

MinimumJobWorkers

The MinimumJobWorkers parameter is used to specify the minimum number of job workers to be added when the server is started and remain in the system at any given point of time. This parameter can be increased or decreased depending on the load on each server.

Valid values:

The default value is for the MinimumJobWorkers parameter is **4**.

The range of integer values considered valid by the MinimumJobWorkers are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

Example:

If the server is expected to handle a large number of queues/topics simultaneously, the MinimumJobWorkers parameter may be set at a higher number so as to ensure that this would not result in idle JobWorkers. Similarly, where the load of the server is expected to be little, the number can be decreased.

MaximumJobWorkers

The MaximumJobWorkers parameter allows the User to set the limit on the number of JobWorkers allowed.. The MaximumJobWorkers parameter can be increased or decreased depending on the expected load on each server.

Valid values:

The default value of the MaximumJobWorkers parameter is **40**.

The range of integer values considered valid by the MaximumJobWorkers are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

Example:

Allowing unchecked addition of JobWorkers could result in too many threads of JobWorkers being created thus hampering the performance of the system. Depending on the requirements the value for number of JobWorkers allowed can be increased or decreased.

Dispatcher

Presented below are the details for are the parameters present in **Fiorano->Dispatcher->config**.

Name

This parameter is used to specify the name of the dispatcher.

Valid values:

The default value of the Dispatcher parameter is **FioranoMQ Dispatcher**.

Any String may be used as a valid value for this parameter.

PingTimeoutAttribute

The PingTimeoutAttribute parameter describes the time in milliseconds after which the server is pinged.

The dispatcher has to make sure that the server it is connected to is alive, and this is achieved through pinging the Server intermittently. This intermittent time interval is specified by this parameter.

Valid values:

The default value of the PingTimeout Attribute is **30000**.

The range of integer values considered valid by the PingTimeoutAttribut are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

InitialTimeoutAttribute

The InitialTimeoutAttribute parameter describes the initial timeout in milliseconds in connecting with a server. The dispatcher will time out if it cannot connect to the server within this time.

Valid values:

The default value of the InitialTimeoutAttribute is **2000**.

The range of integer values considered valid by the InitialTimeoutAttribute are -2^{31} to $2^{31}-1$.

All values less than zero retrieve the same result.

Chapter 12: Repeater Configuration

This chapter explains the parameters that can be configured for the FioranoMQ Repeater that is used for exchanging information between different MQ Servers connected in a particular network topology such as hub-spoke, mesh or bus-based. These are the various parameters that are configured under the **FioranoMQ→Fiorano→Repeater→*** tree node in the Repeater Profile in Offline configuration using Studio.

For more information on FioranoMQ Repeater, refer to **Chapter 11 Repeater** of *FioranoMQ Handbook*.

FioranoRepeaterManager

Presented below are the details for the parameters present in **Fiorano->Repeater->RepeaterManager->FioranoRepeaterManager->config**.

Name

This parameter is used to specify the name of the Repeater Instance.

Valid values:

The default value of the Dispatcher parameter is **Repeater**.

Any String may be used as a valid value for this parameter.

DurableSubscriptionInfoFileName

The DurableSubscriptionInfoFileName parameter is used to specify the name of the XML file that contains information related to durable subscriptions.

Valid values:

The default value of the DurableSubscriptionInfoFileName is **rp_durable.xml**.

Any valid xml filename is an acceptable parameter.

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`. If an absolute path for a filename then that file is used.

PingInterval

The PingInterval parameter describes the time out in milliseconds after which the connection to the Server is pinged.

The repeater makes sure that the Server it connected to is still alive by pinging it intermittently. The time interval between pings is specified by this parameter.

Valid values:

The default value for the PingInterval parameter is **10000**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

AvoidLoopback

The AvoidLoopback parameter is used to specify whether cyclic links are valid in the repeater.

Loopback occurs when there are a set of links connecting FioranoMQ Servers so as to allow the formation of a "cycle". If a message were to originate at a particular FioranoMQ Server then the message would traverse the entire cycle and return to the originating Server. If AvoidLoopback is enabled the message will not be transferred further. If AvoidLoopback is not enabled the message stays within the cycle indefinitely.

Valid values:

The default value of the AvoidLoopback parameter is **yes**.

- **yes** - Loopback is avoided.
- **no** - Loopback is not avoided.

TransactionSize

The TransactionSize parameter is used to specify the number of messages that will amount to a complete transaction.

Valid values:

The default value of the TransactionSize parameter is **0**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

TransactionTimeout

In a transaction a group of messages together are seen as one transaction. If a message which is a part of the transaction is not received by the subscriber within the TransactionTimeoutPeriod timeout period specified in milliseconds, the transaction commits the messages received as a completed transaction.

Valid values:

The default value of the TransactionTimeout parameter is **10**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

SourceServer

Presented below are the details for the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->SourceServer**.

Name

This parameter is used to specify the name of the source of the FioranoMQ Server.

Valid values:

The default value of the Dispatcher parameter is **LocalhostSourceServer**.

Any String may be used as a valid value for this parameter.

ConnectionInfo

Presented below are the details for the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->SourceServer->ConnectionInfo**.

ProtocolType

The ProtocolType parameter describes the underlying protocol to be used to connect to the source of the FioranoMQ Server.

Valid values:

The default value of the ProtocolType Server is **TCP**.

JMXProtocol

The JMXProtocol parameter describes the underlying communication protocol that is used for to connect the JMX connection to the Source FioranoMQ Server.

Valid values:

The default value of the JMXProtocol is **rmi**.

JMXPort

The JMXPort parameter specifies the JMX port hosting the Source FioranoMQ Server, to which a JMX connection is made.

Valid values:

The default value of this parameter is **1858**.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

All values less than zero retrieve the same result.

ServerURL

The ServerURL parameter specifies the URL of the FioranoMQ Server that will be a part of a link. The FioranoMQ Server identified ten serves as the source Server in the link.

Valid values:

The default value of the ServerURL is **http://localhost:1856**.

Format: http://hostname:port

UserName

The UserName parameter specifies the login name used by the repeater to connect to the source FioranoMQ Server.

Valid values:

The default value of the UserName parameter is **anonymous**.

Any String may be used as a valid value for this parameter.

Password

The Password parameter specifies the login name used by the repeater to connect to the source FioranoMQ Server.

Valid values:

The default value of the password parameter is **anonymous**.

Any String may be used as a valid value for this parameter.

TopicConnectionFactory

The TopicConnectionFactory parameter specifies the topic connection factory used by the repeater to connect to the source FioranoMQ Server.

Valid values:

The default value of the TopicConnectionFactory parameter is **primaryTCF**.

Any String may be used as a valid value for this parameter.

ServerSecurityManager

The ServerSecurityManager parameter specifies the Security Manager used by the source FioranoMQ Server.

Valid values:

The default value of the ServerSecurityManager parameter is **null**.

Any String may be used as a valid value for this parameter.

ProxyURLAttribute

The ProxyURLAttribute parameter specifies the URLs of the proxy Servers that are to be used. Multiple backup URLs may be specified as a semicolon-separated string of URLs.

Valid values:

The default value of the ProxyURLAttribute is **null**.

ServerSecurityManager parameter LinkTopicInfo

Presented below are the details for the parameters present in **Fiorano->Repeater->FioranoRepeaterManager->Link->LinkTopicInfo**.

IsDurable

The IsDurable parameter specifies whether the link between the source and the target Server is durable. A durable link ensures that no messages are lost across the repeater in case of network failure.

Valid values:

The default value of the IsDurable parameter is **false**.

- **true** - Link is durable
- **false** - Link is not durable

ConnectionMode

The ConnectionMode parameter specifies whether the same JMS connection will be used for replicating the data across JMS Servers or if a separate connection will be created to be used for each link.

Valid values:

The default value of the ConnectionMode parameter is **Shared**.

- Shared - In this mode the repeater will use the same JMS connection for replicating the data across JMS Servers.
- Exclusive - In this mode the repeater will use separate JMS connections for replicating data across servers. In the 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 a destination. A single connection is used for all the multiple topics.

Type

The Type parameter specifies if a link should be permanently connected to the target Server or if the link should when a subscriber exists.

Valid values:

The default value for the Type parameter is **SUBSCRIBER EXISTS**.

- SUBSCRIBER EXISTS - The link with the target server that is used for replication does not exist unless there is a subscriber present on that target server.
- ALWAYS - The link that the repeater has with the server will be a permanent link.

ReplyOn

The ReplyOn parameter specifies the destination on which the repeater listens for replies to requests it forwards.

Valid values:

The default value is **null**.

SourceTopicName

The SourceTopicName parameter specifies the name of the topic on which subscriptions are made on source Server of the link.

Valid values:

The default value of the SourceTopicName is **primaryTopic**.

TargetTopicName

The TargetTopicName parameter specifies the name of the topic on the target server where messages are replicated.

Valid values:

The default value of the TargetTopicNameParameter is **secondaryTopic**.

MessageSelector

The MessageSelector parameter specifies the selector that is set on a link between the Servers that ensures only the required messages are exchanged between Servers.

Valid values:

The default value of the MessageSelector parameter is **null**.

Only messages where header and property values match those of the selector will be readable. A message selector matches a message if the selector arrives as **true** when the message header values and property values are substituted by their corresponding identifiers by 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

This chapter explains the parameters that can be configured for the FioranoMQ Bridge which enables inter-operability between FioranoMQ and different messaging queue vendors. These are the various parameters that are configured under the **FioranoMQ→Fiorano→Bridge→*** tree node in the Bridge Profile in Offline configuration using Studio.

For more information on FioranoMQ Bridge, refer to **Chapter 12 Bridge** of *FioranoMQ Handbook*.

ConnectionInfo

Presented below are the details for the parameters present in **Fiorano->Bridge->FioranoBridge->FioranoConnectionManager->config-><LinkName>-><ServerName>->ConnectionInfo**.

ServerType

The ServerType parameter is needed to mention the type of the Server being used.

Default Value:

JMS - The default type of server.

ProviderUrl

The ProviderURL value specifies where the Server is currently running.

The ProviderUrl parameter allows the User to configure the bridge that links 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 InitialContextFactory parameter specifies 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 BridgeUser parameter is the User Name which the bridge uses to connect to the Server.

Default Value:

Anonymous - The default bridge user name.

BridgePassword

The BridgePassword is the password which the bridge uses to connect to the Server.

Default Value:

Anonymous - The default password.

QCF

The QCF parameter specifies the connection factory with which the connection is to be established.

Default Value:

primaryQCF - The default connection factory used for forming the connection.

Protocol

The Protocol parameter specifies 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 parameter is the User Name used to connect to the Server.

Default Value:

ayrton - The default username.

SecurityCredentials

The SecurityCredentials parameter is the Password used to connect to the Server.

Default Value:

Senna - The default password.

ServerSecurityManager

The ServerSecurityManager parameter is the Security Manager of the server linked by a bridge.

Default Value:

Null - No server Security Manager.

QueueInfo

Presented below are the details for 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 QueueName parameter is the name of the queue present in the target/source Server to be linked by the bridge. This parameter 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

This chapter explains the various parameters that can be configured for connectors used in the FioranoMQ Server (JMX & RMI) for handling remote connection requests to access/modify server configurations at runtime. These are the various parameters that are configured under the **FioranoMQ→Fiorano→jmx→connector→JMSBasedJMXConnector2** and **FioranoMQ→Fiorano→jmx→connector→RMI BasedJMXConnector** tree nodes in Offline configuration through Studio.

For more information on JMX Connectors, refer **Chapter 3 Connection Management** of *FioranoMQ Concepts Guide*.

JMS Based Connector Configuration

Presented below are the details for the parameters present in **JMX->Connector->JMSBasedJMXConnector**.

InterceptorClassName

The InceptorClassName parameter is the Class Name that is used to intercept all jmx calls. When JMX calls arrive at the container, this parameter is used to intercept and process these calls.

Default Value:

fiorano.jmx.connector.FioranoJMXI nterceptor - The name of the class used to intercept all jmx calls.

SecurityProtocol

The SecurityProtocol is the secure protocol used if the Connector Server the Server is connected to is running on a secure protocol. Example: SUN_SSL, NO_SECURITY.

Valid values:

- NO_SECURITY - No Security protocol is used.
- SUN_SSL - Sun_SSL is the security protocol that is used.

Default Value:

- NO_SECURITY - No security protocol is used by default.

Protocol

The Protocol parameter is the transport protocol used to connect to the JMS Server. This parameter allows the user to choose between available protocols such as LPC, TCP, and HTTP.

Valid values:

- TCP - TCP is the transport protocol that is used.
- HTTP - HTTP is the transport protocol that is used.
- LPC - LPC is the transport protocol that is used.

Default value:

LPC - LPC is the default protocol used to connect to the JMS Server.

SecurityManagerClass

The SecurityManagerClass parameter is the Security Manager class implements IFMQSecurityManager.

Valid values:

NO_SECURITY - No Security Manager class is being used.

fiorano.jmx.connector.fmq.security.JSSESecurityManager - The Security Manager class name for Sun_SSL.

Default value:

NO_SECURITY – If the No Security Protocol is used, no the SecurityManagerClass will be used by default. However if a SecurityProtocol is used, say SUN_SSL, then the corresponding SecurityManagerClass, such as `fiorano.jmx.connector.fmq.security.JSSESecurityManager` must be used.

Dependencies:

SecurityProtocol

QueueConnectionFactory

The QueueConnectionFactory parameter is used by the 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 being used.

ServerPort

The ServerPort is the port of the JMS Server to which the ConnectorServer connects.

The ServerPort allows the User to configure the server to run on different ports. When a User has to run two or more servers on the same machine, changing the ServerPort value is mandatory since two servers cannot run from the same port.

Default value:

1856 - The default port number of the JMS server.

ServerAddress

The ServerAddress parameter is the IP of the FioranoMQ Server to which the ConnectorServer connects.

Note: ServerAddress should be **localhost/LocalIP/MachineName** when the LPC QueueConnectionFactory is being used.

Default Value:

localhost - The default ServerAddress of the JMS Server.

Username

The UserName parameter is the User Name that is used to connect to the JMS Server.

Default Value:

Admin – Default username.

Password

Password is the parameter that serves as the password used to connect to the JMS Server.

ConnectQueueName

The ConnectQueueName parameter is the Connect Queue Name on which the Connector Server listens for connection requests.

Default Value: **JMX_CONNECT_QUEUE**

RequestQueueName

The RequestQueueName parameter is the Request Queue Name on which the Connector Server listens for jmx queries

Default Value: **JMX_SERVICE_QUEUE**

NotifQueueName

The NotifQueueName parameter is the Notification Queue Name on which the Connector Server sends notifications.

Default Value: **JMX_SERVICE_QUEUE**

RMI Based Connector Configuration

Presented below are the details for the parameters present in **JMX->Connector->RMIBasedJMXConnector**.

RMI ServerPort

The RMI ServerPort parameter is the port on which the Mx4J RMI Connector Server connects.

This parameter allows the User to configure the RMI port of the Server. When a User runs two or more Servers on the same machine, changing the RMI ServerPort value is mandatory as two Servers cannot run on the same RMI port.

Default Value: **1858** - The default RMI port of the Server.

InterceptorClassName

The InterceptClassName parameter is the class Name that is used to intercept all jmx calls

Default Value: **fiorano.jmx.connector.FioranoJMXInterceptor** - The name of the class used to intercept all jmx calls.

RmiPortExportObjects

Port on which Rmi Registry will export the Mx4J RMI Connector stubs. If set to 0, stubs will be exported at a random port at runtime.

Default Value: 0

Hostname

Specifies host name on which the registry would export objects. The system property 'java.rmi.server.hostname' is set to this value before starting the rmi registry.

Note: It can take comma separated values too. Example: 192.168.1.2, 10.0.0.1

Default Value: null

RmiServerSocketFactoryClassName

This parameter denotes the name of server socket factory class to be used by rmi runtime. This class implements 'java.rmi.server. RMI ServerSocketFactory' interface.

Default Value: fi orano. rmi . serverfac. def. Fi oranoRMI ServerSocketFactory

RmiClientSocketFactoryClassName

This parameter denotes the name of client socket factory class to be used by rmi runtime. This class implements 'java.rmi.server. RMI ClientSocketFactory' interface.

Default Value: fi orano. rmi . cl i entfac. def. Fi oranoRMI Cl i entSocketFactory

JMX Engine Configuration

Presented below are the details for the parameters present in **Fiorano->jmx->engine->JMXEngine->ClientJMXEngine**.

SecurityProtocol

The SecurityProtocol is the secure protocol used if the Connector Server the Server is connected to is running on a secure protocol. Example: SUN_SSL,NO_SECURITY.

Valid values:

- NO_SECURITY - No Security protocol is used.
- SUN_SSL - Sun_SSL is the security protocol that is used.

Default Value:

NO_SECURITY - No security protocol is used by default.

Protocol

The Protocol parameter is the transport protocol used to connect to the JMS Server. This parameter allows the user to choose between available protocols such as LPC, TCP, and HTTP.

Valid values:

- TCP - TCP is the transport protocol that is used.
- HTTP - HTTP is the transport protocol that is used.
- LPC - LPC is the transport protocol that is used.

Default value:

LPC - LPC is the default protocol used to connect to the JMS Server.

SecurityManagerClass

The SecurityManagerClass parameter is the Security Manager class implements IFMQSecurityManager.

Valid values:

NO_SECURITY - No Security Manager class is being used.

fiorano.jmx.connector.fmq.security.JSSESecurityManager - The Security Manager class name for Sun_SSL.

Default value:

NO_SECURITY – If the No Security Protocol is used, no the SecurityManagerClass will be used by default. However if a SecurityProtocol is used, say SUN_SSL, then the corresponding SecurityManagerClass, such as fiorano.jmx.connector.fmq.security.JSSESecurityManager must be used.

Dependencies:

SecurityProtocol

QueueConnectionFactory

The QueueConnectionFactory parameter is used by the 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 being 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

The ServerAddress parameter is the IP of the FioranoMQ Server to which the ConnectorServer connects.

Note: ServerAddress should be **localhost/LocalIP/MachineName** when the LPC QueueConnectionFactory is being used.

Default Value:

localhost - The default ServerAddress of the JMS Server.

Username

The UserName parameter is the User Name that is used to connect to the JMS Server.

Default Value:

Admin – Default username.

Password

Password is the parameter that serves as the password used to connect to the JMS Server.

ConnectQueueName

The ConnectQueueName parameter is the Connect Queue Name on which the Connector Server listens for connection requests.

Default Value: **JMX_CONNECT_QUEUE**

RequestQueueName

The RequestQueueName parameter is the Request Queue Name on which the Connector Server listens for jmx queries

Default Value: **JMX_SERVICE_QUEUE**

NotifQueueName

The NotifQueueName parameter is the Notification Queue Name on which the Connector Server sends notifications.

Default Value: **JMX_SERVICE_QUEUE**

Monitoring Manager Configuration

Presented below are the details for the parameters present in **Fiorano->jmx->notifications->MonitoringManager**.

ClockTickForMonitoring

The ClockTickForMonitoring parameter specifies the time interval between monitoring instances in milliseconds. The ClockTickForMonitoring indicates the number of milliseconds after which the timer for monitoring is reset. This parameter sets the time interval in milliseconds between successive monitoring attempts.

Default Value:

5000 - This is equivalent to five seconds and indicates the default idle time after which the monitoring timer is reset.

Chapter 15: Advanced Configuration

This chapter provides information about parameters that are generally configured along with the parameters described in all earlier chapters. All the miscellaneous modules, such as Route Manager, Directory Service, RDBMS Manager, Resource Manager, etc. are described in this chapter.

Route Manager Configuration

Presented below are the details for the parameters present in **Fiorano->etc->RouteManager**.

MaxTopicBuffer

The MaxTopicBuffer parameter is the upper limit for the message buffer size in bytes.

This parameter allows the user to specify the buffer size in bytes of the messages in the queue. Depending on the density 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

Presented below are the details for the parameters present in **Fiorano->etc->RdbmsManager**.

URL

The URL parameter is the URL of the database where messages are stored. Please refer to the RDBMS Server documentation for formatting this parameter.

The URL parameter allows the User to set/change the URL of the Database Server.

Default Value:

jdbc:hsqldb:../profiles/FioranoMQ/run/RDBMS/FMQ – This is the default URL of the FioranoMQ RDBMS.

JdbcDriver

The JdbcDriver is the parameter that is used for RDBMS based Data Storage. Please refer to the Database Driver documentation for formatting this parameter.

Default Value:

org.hsqldb.jdbcDriver - The default driver used for RDBMS based Data Storage.

Password

The Password parameter is the password used by a RDBMS User – This is the password used for connecting to the RDBMS Server.

Username

The Username parameter is the Database User Name used for connecting to the RDBMS Server.

Default Value:

Sa

MaxConnections

The MaxConnections parameter allows a maximum number of Database connections to be created.

The MaxConnections parameter enables restricting the total number of Database connections created. Having too many database connections could affect the performance of the Server.

Default Value:

200 - A maximum of 200 Database connections are allowed at any given point of time.

DefaultConnections

The DefaultConnections parameter is the number of Database connections created upon starting the Server.

If the User wants to increase the number of Database connections created upon startup, the value of this parameter should be increased.

Default Value:

1 - By default only 1 Database connection is created at startup.

MaxNoOfRowsFetch

The MaxNoOfRowsFetch parameter specifies the maximum number of rows fetched at a time from the database when a query entered.

When a large number of records needs to be fetched increasing the number of entries fetched per query statement require fewer queries to be entered as more records would be fetched per query.

Default Value:

100 - The maximum number of rows fetched at a shot from the database is 100.

EnableReconnect

The EnableReconnect parameter which is a boolean indicates whether reconnection needs to be enabled.

The EnableReconnect parameter allows the User to decide whether the Server needs to reconnect to the RDBMS Server when the connection is lost. Where reconnection is not necessary or should not be performed this parameter should be set to False.

Valid values:

True - Enables reconnection attempts to the RDBMS Server when connection is lost.

False - Disables reconnection attempts to the RDBMS Server when connection is lost.

Default Value:

True - Enables reconnection attempts to the RDBMS Server when connection is lost.

ReconnectTimeInterval

The ReconnectTimeInterval parameter is the time interval in milliseconds between RDBMS reconnection attempts.

The ReconnectTime Interval parameter allows the User to specify the interval between successive reconnection attempts when the connection to the RDBMS Server is down. This value can be increased when the time taken resetting the RDBMS Server is long.

Default Value:

10000 - Time in milliseconds between RDBMS reconnection attempts (which is equivalent to ten seconds).

WaitTime

The WaitTime parameter is the waiting time if no additional connections are available in the connecting pool. Zero indicates an indefinite waiting time.

The WaitTime parameter allows the User to configure the length of time the server waits to reconnect when the connection pool is empty.

Default Value:

0 – Indicates as infinite/indefinite waiting time.

ConnectionTimeoutInterval

The ConnectionTimeoutInterval parameter indicates the connection timeout interval in milliseconds.

The ConnectionTimeoutInterval parameter allows the user to configure the connection timeout interval which that is the length of time after when the connection times out if no activity occurs during that period of time.

Default Value:

10000 - Connection timeout interval is by default set to ten seconds.

PropertiesFilename

The PropertiesFilename parameter is the JDBC properties file name. This file contains database specific details for various SQL Types. This parameter needs to include the name of the file jdbc_*.cfg available at %FIORANO_HOME%/fmq/profiles/%PROFILE%/conf directory.

Default Value:

jdbc_hsql.cfg – This is the default JDBC properties File Name.

DbConnectivityQuery

The DbConnectivityQuery parameter represents the query that should be executed to check the health of a connection.

The DbConnectivityQuery parameter allows the User to configure the query to be used to test the health/validity of a connection.

Default Value:

select * from tab - The default query used to check if the connection is alive.

EnableRdbms

The EnableRdbms parameter enables/disables the Database Manager. Setting this flag to true is a prerequisite for using an RDBMS Server as a back-end message store.

EnableRdbms allows the User to enable/disable the RDBMS message store feature.

Valid Values:

True - Enables the RDBMS message store.

False - Disables the RDBMS message store.

Default Value:

True - Enables the RDBMS message store.

Resource Manager Configurations

Presented below are the details for the parameters present in **Fiorano->etc->ResourceManager**.

MaxMemoryThresholdAllowed

The MaxMemoryThresholdAllowed parameter is the maximum memory threshold that is allowed to be used. The memory threshold 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%.

The MaxMemoryThresholdAllowed parameter allows the User to restrict memory usage preventing excessive memory usage by the Server processes.

Default Value:

0.8 - The default memory threshold is set to 80%.

TimeInterval

The TimeInterval parameter is the time interval after which the monitoring thread runs once again. The time interval is measured in milliseconds.

The TimeInterval parameter allows 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, this parameter should be set to a high value. However, if monitoring is to be done over a short period of time, this parameter should be set to a smaller value.

Default Value:

10000 - A monitoring thread runs once every 10000 milliseconds (which is equivalent to ten seconds).

EnableMonitoringThread

The EnableMonitoringThread parameter specifies whether the monitoring thread is to be enabled.

The EnableMonitoringThread parameter allows the User to enable/disable the monitoring thread. If the User does not want the monitoring thread to run at periodical intervals, this parameter should be set to False or set to No in Studio.

Valid Values:

True/Yes (in Studio) – Enables the running of the monitoring thread at periodic intervals.

False/No (in Studio) - Disables the running of the monitoring thread at periodic intervals.

Default Value:

True/Yes (In Studio) - Enables the monitoring thread to be run a periodic intervals.

EnableVerboseLogging

The EnableVerboseLogging parameter specifies whether to enable/disable verbose logging. If set to false, logs are maintained in the 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].

The EnableVerboseLogging parameter allows the User to choose the format of the logs that are being monitored.

Valid Values:

- **True/Yes (in Studio)** - Enables Verbose Logging.
- **False/No (in Studio)** - Disables Verbose Logging.

Default Value:

True/Yes (in Studio) - Enables Verbose logging

EnableThreadDumpInfo

The EnableThreadDump parameter specifies whether to enable/disable thread dump information given the memory in the Server falls below the given threshold.

A User can choose whether to enable the thread dump information using the EnableThreadDump parameter. When the server memory falls below a certain threshold, this parameter should be set to True/Yes (in Studio). If set to True/Yes (in Studio) thread information is dumped into a file so that the User does not loose data.

Valid Values:

- **True/Yes (in Studio)** – Enables dumping thread information into a file so as not to loose data when the Server is low on memory.
- **True/No (in Studio)** - Disables dumping thread info information into a file so as not to loose data when the Server is low on memory.

Default Value:

True/Yes (in Studio) - Enables the dumping thread info information into a file so as not to loose data when the Server is low on memory.

PingManager Configurations

Presented below are the details for the parameters present in **Fiorano->etc->PingManager**.

PingTimeout

The PingTimeout parameter indicates the ping timeout time in milliseconds. The PingTimeout parameter the length of time after which a connection that is disconnected.

This parameter allows the User to configure the length of time after which it should stop trying to connect to a connection that is dead.

Default Value:

60000 - The number of milliseconds after which a connection that is dead is disconnected.

JobManager Configurations

MinimumJobWorkers

The MinimumJobWorkers parameter specifies the minimum number of JobWorkers present at any given time in the system.

The MinimumJobWorkers parameter allows the User to configure the minimum number of JobWorkers that may be added as the Server starts and be present in the system at any given point of time. This parameter can be increased or decreased depending upon the load on each server. If the server is to handle a large number of queues/topics simultaneously, this parameter should be set at a higher number. If the load of the server is expected to be small, this parameter can be decreased so as not to have idle JobWorkers in the system.

Default Value:

4 - The Minimum number of JobWorkers Available in the system is four.

MaximumJobWorkers

The MaximumJobWorkers parameter specifies the maximum number of JobWorkers present at any given time in the system.

The MaximumJobWorkers parameter allows the User to configure the maximum number of JobWorkers that may be added as the Server starts and be present in the system at any given point of time. This parameter allows the user to set the limit on the number of JobWorkers allowed in the JobManager. Allowing unchecked additions of JobWorkers could result in too many JobWorkers threads being created, thus hampering the performance of the system. Depending on the scenario, this value of this parameter can be increased or decreased.

Default Value:

40 - The Maximum number of JobWorkers available in the system is 40.

JobsPerWorker

The JobsPerWorker parameter indicates the 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 larger than the number of job workers present, then new JobWorkers are added. The maximum number of job workers that may be present at any given time cannot exceed 19The value of the maximum job workers allowed in the system cannot be changed.

Default Value:

4 -The default number of jobs assigned per job worker at any point of time is 4.

ObjectManager Configurations

Presented below are the details for the parameters present in **Fiorano->etc->runTimeObjectManagers->ObjectMaanger/gmsObjectManager**.

HashCodeReuseAlgoUsed

The HashcodeReuseAlgoUsed parameter indicates whether the Reuse HashCode Algorithm is used creating Object Handles.

Valid Values:

- **True/Yes (in Stubio)** – HashcodeReuse Algorithm is used.
- **False/No (in Studio)** - HashcodeReuse Algorithm is not used.

Default Value:

True/Yes (in Studio) - The hash code reuse algo is used in the creation of Object Handles.

ThreadManager Configurations

Presented below are the details for Following are the parameters present in **Fiorano->etc->ThreadManager**.

DelayBetweenAttempts

The DelayBetweenAttempts parameter is the delay in milliseconds between two successive attempts to create a thread when the initial thread creation of a critical operation thread has failed.

If, the creation of a critical operation thread fails, the DelayBetweenAttempts parameter along with the MaximumAttempts parameter ensures that further attempts to create the thread do not affect the performance of the Server.

Default Value:

100 – There is a 100 milliseconds delay between successive attempts to create a thread.

Dependencies:

MaximumAttempts

MaximumAttempts

The MaximumAttempts parameter specifies the maximum number of tries to be attempted to create a thread.

If the creation of a thread results in a failure, after the specified time interval, called the DelayBetweenAttempts parameter, another attempt is made to create the thread. The number of attempts to be made in creating a thread are specifies (and controlled) by this parameter.

Default Value:

8 – The maximum number of attempts to be made to create a thread, before reporting failure of thread creation, is eight.

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

Presented below are the details for the parameters present in **Fiorano->etc->ClientLogManager->DefaultLogManager**.

Logging into FioranoMQ is achieved with the help of Log4J. Logging in using Log4J makes it possible to enable logging at runtime without modifying the application binaries.

LogDir

The LogDir parameter is the Base Directory where logs are stored.

The LogDir parameter allows the User to specify the directory in which the logs are to be stored.

Default Value: ../profiles/FioranoMQ/run/logs/ - The directory where logs are stored in FioranoMQ.

Encoding

The Encoding parameter identifies the Name of the character encoding that is used.

The Encoding parameter allows the User to specify the character encoding to be used when saving logs. This parameter should be set before logs are written and saved.

Default Value:

UTF-8 - The default character encoding is UTF-8.

Level

The Level parameter is the log level that is used by newly constructed handler objects.

The Level parameter allows the User to increase or decrease the log level. Increasing the log level allows more detailed information on any errors or warnings that occur to be displayed. Decreasing the log level briefs error information messages to be displayed.

Valid Values:

- **-1 to 6 and 10** - Any integer between Zero and six. 10 is the highest integer that may be used. -1.0 is the lowest integer that may be used. All log entries are logged at a particular log level. Each logger has an associated logging level which may be anyone of the following values:
- -1 INHERIT - Inherits log level from its parent.
- QUIET - No logged information.
- FATAL - Severe errors that might cause premature termination.
- ERROR - Other runtime errors or unexpected conditions.
- WARN - Use of deprecated APIs, poor use of API, almost errors, other runtime situations that are undesirable or unexpected but not necessarily wrong.
- INFO - Interesting runtime events at startup/shutdown.
- DEBUG - Detailed information on **flow** through the system.
- TRACE - More detailed information.
- 10 ALL - All logged information.

Default Value:

2 The default log level is two.

FileSizeLimit

The FileSizeLimit parameter specifies the maximum size of each log file where 0 => infinite. This parameter is used by the File Handler.

The FileSizeLimit parameter allows the user to limit the size of the log file. Restricting storage space by specifying the maximum limit of a log ensures that memory does not run out.

Default Value:

0 – Zero = a file size with no size limit.

FileCount

The FileCount parameter specifies the maximum number of log files that may be created. This parameter is used by the File Handler.

The FileCount parameter allows the User to specify the number of log files that may be created.

Default Value:

1 - By default only 1 log file may be created.

AppendToFile

The AppendToFile parameter specifies the append mode of the File Handler. By default, the appended mode of the FileHandler is enabled.

The AppendToFile parameter allows the User to specify whether to append data to log files. If set to false, 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 specified. After the specified log file, the earliest log file is over written with current log entries. The log files keep rotating. If set to true, log entries are appended to the end of the current log file.

Valid Values:

- True/Yes (in Studio) - Enables Appending to file.
- False/No (in Studio) - Disables Appending to file.

Default Value:

True/Yes (in Studio) - Enables Appending to file.

Formatter

The Formatter parameter specifies 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

Presented below are the details for the parameters present in **Fiorano->etc->MQDefaultObjCreator**.

AutoUpdationAllowedForCFs

The AutoUpdationAllowedForCFs is the parameter that sets the AutoUpdate property for FioranoMQ. The default the ConnectionFactory's parameter is based on the AutoUpdationAllowedForCFs flag. To set this parameter, set the boolean option in ConnectionFactory indicating whether updation is allowed in Connection Factories (CF) during startup.

Valid Values:

- True/Yes (in Studio) – Enables AutoUpdation for default connection factories.
- False/No (in Studio) - Disables AutoUpdation for default connection factories.

Default Value:

True/Yes (In Studio) - Enables the AutoUpdate property for FioranoMQ default CFs.

BackupServerIp

The BackupServerIP parameter specifies the backup server IP. This IP is configured in the backup URL of the default Connection Factories.

The BackupServerIP parameter allows the user to set a backup server IP for the default Connection Factories.

Default Value:

No Value - The BackupServerIP parameter is left blank by default. The User can set the Backup Server IP by specifying a value for this parameter.

Dependencies:

BackupServerPort

BackupServerPort

The BackupServerPort parameter specifies the Backup Server port. This port is configured in the backup URL for setting the default Connection Factories.

The BackupServerPort parameter allows the User to set a Backup Server port for the default Connection Factories.

Default Value:

-1 - This is the default value of the BackupServerPort parameter but it is an invalid port number and is therefore ignored. To have a valid backupUrl both the BackupServerIP and the BackupServerPort should be valid.

Dependencies:

BackupServerIp

Timer Service Configurations

Presented below are the details for the parameters present in **Fiorano->etc->TimerService**.

Name

The Name parameter is the name of the timer service.

Default Value:

FioranoTimerService – is the default name of the timer service.

ClockTick

The ClockTick parameter is the time period in milliseconds after which scheduling of processes are checked.

The ClockTick parameter is the time interval in milliseconds between successive ticks of the timer's clock. Increasing this value results in fewer scheduling of processes checks.

Default Value:

500 – is the time in milliseconds after which process are checked for scheduling.

JNDI Configurations

NativeFileNamingManager

Presented below are the details for the parameters present in **Fiorano->jndi->NamingManager->File->NativeFileNamingManager**.

DefragmentationLowerThreshold

The DefragmentationLowerThreshold parameter specifies the minimum number of deleted entries for triggering auto-defragmentation of the AdminObjectManager

It is necessary to remove the deleted entries from the cache to prevent the cache from becoming from large. However, cache compaction for each deleted entry should be prevented as this process is quite costly. The minimum number of entries that should be deleted to trigger defragmentation is set by this parameter.

Default Value:

10 - Defragmentation is triggered only when there are a minimum of ten deleted entries.

Dependencies:

DefragmentationUpperThreshold. DefragmentationPercentage.

DefragmentationUpperThreshold

The DefragmentationUpperThreshold parameter is the maximum number of deleted entries after which auto-defragmentation of AdminObjectManager is guaranteed/triggered during start up.

When the number of deleted entries in the cache becomes more than the DefragmentationUpperThreshold parameter, the defragmentation process is triggered. The value should be greater than the DefragmentationLowerThreshold. On the other hand, having this parameter set to a very high value could result in a waste of memory.

Default Value:

15 - Defragmentation is triggered only when there are 15 deleted entries.

DefragmentationLowerThreshold. DefragmentationPercentage.

DefragmentationPercentage

The DefragmentationPercentage parameter specifies the threshold percentage for auto-defragmentation during startup when the number of deleted entries fall between the thresholds mentioned above.

When the number of deleted entries becomes more than the DefragmentationLowerThreshold set at 40, defragmentation is not triggered immediately. It is triggered either if the number of deleted entries is more than the DefragmentationUpperLimit, or more than 40 or if the percentage of deleted entries present in the cache becomes more than the value set in the DefragmentationPercentage parameter.

Default Value:

20 - The percentage of deleted entries in the cache should be more than 20% to trigger defragmentation.

Path

The Path parameter is the root path for this Naming Manager.

Default Value:

The directory considered the current directory will become the default root path for the NativeFileNamingManager.

CacheNamingManager

Presented below are the details for the parameters present in **Fiorano->jndi->CacheNamingManager**.

TTL

The TTL parameter is the duration in milliseconds when the administrative object remains available within the memory.

The TTL parameter allows the User to determine how long the administrative object entries remain in the cache. Using this parameter the length of time these entries may remain available can be increased. If the data attached to these objects are needed only for a short time the value of this parameter can be reduced.

Default Value:

60000 - is the time in milliseconds when the admin objects remain within the memory. This value is equivalent to 60 seconds.

XMLFileNamingManager

Presented below are the details for the parameters present in **Fiorano->jndi->XMLFileNamingManager**.

Path

The Path parameter is the absolute or the relative path of the xml file used in persisting information of administrative objects.

Default Value:

The current directory is considered the default path of the Path parameter.

Filename

The Filename parameter is the ServiceProvider instance of the profile.

Default Value:

admin.xml – is the default xml filename

LDAPNamingManager

Fiorano->jndi->LDAPNamingManager

LdapProviderUrl

The LdapProviderUrl parameter is the URL of the LDAP Server from where information related to administrative objects are read.

The LdapProviderUrl 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

The LdapBasedn parameter is the Base Domain Name within the LDAP Server under which all information related to administrative objects is stored.

The LdapBasedn parameter allows the User to set the Base Domain from which information related to the administrative objects may be retrieved.

Default Value:

o=fiorano, c=US – is the default Base DomainName within the LDAP Server.

Principal

The Principal parameter is the User Name that NamingManager uses to connect to the LDAP Server for accessing/storing information related to administrative objects.

The Principal parameter allows the User to set authentication details needed to connect to the LDAP Server.

Default Value:

o=fiorano, c=US - The default username with which to connect to the LDAP Server.

Credentials

The Credentials parameter is the password connected to a specific used to login to a specific LDAP Server.

LdapInitialCtxFactory

The LdapInitialCtxFactory parameter is the Class Name of the initial context factory which is used to connect with specific LDAP Servers.

Default Value:

com.sun.jndi.ldap.LdapCtxFactory - The default initial context factory class used for connecting to the LDAP Server.

LdapPollInterval

The LdapPollInterval parameter is the TimeInterval after which Naming Manager tries to reconnect to LDAP Server in the event of a connection breakdown.

The LdapPollInterval parameter allows the User to set the time interval after the length of which an attempt to connect to the LDAP Server is made if the connection breakdowns. This value coupled with the LdapPrimaryServerReconnectAttempts set at 64, determines the total amount of time after which no more attempts are made to connect to the Primary Server.

Default Value:

4000 – is the default time interval between successive attempts to connect to the LDAP Server.

Dependencies:

LdapPrimaryServerReconnectAttempts.

LdapPrimaryServerReconnectAttempts

The LdapPrimaryServerReconnectAttempts parameter specifies the number of reconnect attempts to the Primary Server that Naming Manager makes before trying to connect to the Secondary Server.

The LdapPrimaryServerReconnectAttempts parameter allows the User to decide on the number of attempts to be made to connect to the Primary Server before connecting to the Secondary Server. This value coupled with the LdapPollInterval set at 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 BackupLdapProviderUrl parameter is the URL of the Backup LDAP Server which Naming Manager tries to connect to when the Primary LDAP Server is not available.

The BackupLdapProviderUrl parameter allows the User to setup the Backup LDAP Server to which the Naming Manager can connect when the Primary LDAP Server is down. After trying to connect to the Primary Server, set by the LdapPrimaryServerReconnectAttempts at 64, attempts, an attempt to establish a connection with the Server running at a URL specified by this parameter will be made.

Dependencies: LdapPrimaryServerReconnectAttempts.

BackupLdapBasedn

The BackupLdapBasedn parameter specifies the Base Domain Name in the Backup LDAP Server under which all information related to administrative objects is stored.

The BackupLdapBasedn parameter allows the User to set the Base Domain from which information related to the administrative objects may be retrieved.

Default Value:

o=fiorano, c=US – is the default Base Domain Name within the Backup LDAP Server.

BackupPrincipal

The BackupPrincipal parameter specifies the User Name used by the NamingManager to connect to the backup LDAP Server for accessing/storing information relating to administrative objects.

The BackupPrincipal parameter allows the User to set authentication details so it can connect to the backup LDAP Server.

Default Value:

`o=fiorano, c=US` – is the default username that is used to connect to the backup LDAP Server.

BackupCredentials

The BackupCredentials is the password with specified login details that the User uses to connects to a specified backup LDAP Server.

BackupLdapInitialCtxFactory

The BackupLdapInitialCtxFactory is the parameter that specifies the Class Name of the initial context factory that is used to open connections with a specified backup LDAP Server.

Default Value:

`com.sun.jndi.ldap.LdapCtxFactory` – is the default initial context factory class used to connect to the backup LDAP Server.

BackupLdapPollInterval

The BackupLdapPollInterval parameter specifies the TimeInterval after which Naming Manager tries to reconnect to backup LDAP Server in the event of a connection breakdown.

The BackupLdapPollInterval parameter allows the user to set the time interval after which an attempt to connect to the backup LDAP Server is made in the event of a connection breakdown.

Default Value:

`4000` – is the default time interval between successive attempts to connect to the backup LDAP Server.

TopicCFBaseDn

The TopicCFBaseDn parameter specifies is the Base Domain name within the LDAP Server under which all topic connection factories are stored.

The TopicCFBaseDn parameter allows the User to set the domain under which the Topic connection factories are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which the topic connection factories are stored.

QueueCFBaseDn

The QueueCFBaseDn parameter specifies the Base Domain name within the LDAP Server under which all queue connection factories are stored.

The QueueCFBaseDn parameter allows the User to set the domain under which the queue connection factories are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which the queue connection factories are stored.

AdminCFBaseDn

The AdminCFBaseDn parameter specifies the Base Domain Name within the LDAP Server under which all admin connection factories will be stored.

The AdminCFBaseDn parameter allows the User to set the domain name under which the administrative connection factories are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which the administrative connection factories are stored.

UnifiedCFBaseDn

The UnifiedCFBaseDn parameter specifies the Base Domain name within the LDAP Server under which all unified connection factories are stored.

The UnifiedCFBaseDn parameter allows the User to set the domain under which the unified connection factories are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which the unified connection factories are stored.

TopicsBaseDn

The TopicsBaseDn parameter specifies the Base Domain name within the LDAP Server under which all topics are stored.

The TopicsBaseDn parameter allows the User to set the domain under which all Topics are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which the topics are stored.

QueueBaseDn

The QueueBaseDn parameter specifies the Base Domain name within the LDAP Server under which all queues are stored

The QueueBaseDn parameter allows the User to set the domain under which all Queues are stored.

Default Value:

o=fiorano, c=US – is the default domain name under which queues are stored.

RDBMSNamingManager

Presented below are the details for the parameters present in **Fiorano->jndi->RDBMSNamingManager**.

URL

The URL parameter specifies the URL of database server from where information related to administrative objects are read.

The URL parameter allows the User to set/change the URL of the database server.

Default Value:

jdbc:hsqldb:Database – is the default URL of the database server.

Password

The Password parameter specifies the password using which a User with specified login details connects to the database Server that is specified.

Username

The Username parameter specifies the User Name which the Naming Manager uses to connect to the RDBMS for accessing/storing information related to administrative objects.

DatabaseDriver

The database driver parameter specifies the database driver Class Name used to open connect with the specified RDBMS server.

Default Value:

org.hsqldb.jdbcDriver - is the default database driver class name.

SQLStatementFile

The SQLStatementFile parameter specifies the Name of the properties file that is used to define the Name-Value properties specific to a particular database.

The SQLStatementFile parameter allows the user to set/change the properties file.

Default Value:

nm_sqls.properties – is the default properties file used.

ConnectionLostErrorMessage

The ConnectionLostErrorMessage parameter is the error message that is posted when a connection is lost.

The ConnectionLostErrorMessage parameter allows the User to set custom messages to be displayed when connection to the RDBMS Server is lost.

Default Value:

DB_CONNECTION_LOST – is the default connection lost message.

EnableReconnect

The EnableReconnect parameter specifies whether the Naming Manager has to try to reconnect to the database Server in case of connection breakdown.

The EnableReconnect parameter allows the User to decide whether the Naming Manager should make an attempt to reconnect to the RDBMS Server when the connection is lost. Where reconnection is not necessary or should not be performed this parameter should be set to False.

Valid Values:

True - Enables reconnection attempts with the Database Server when the connection is down.

False - Disables reconnection attempts with the Database Server when the connection is down.

Default Value:

True - Enables reconnection attempts with the Database Server when the connection is down.

DatabaseConnectionTimeout

The DatabaseConnectionTimeout parameter specifies the length of time that the Naming Manager will attempt to connect to the Database Server. This parameter value may be increased or decreased as per requirements.

Default Value:

10000 - is the length of time in milliseconds the Naming Manager attempts to connect to the Database Server.

InitializeDB

The InitializeDB parameter specifies whether to initialize the Database for accessing/storing information related to administrative objects.

Valid values:

True - Enables initializing the Database

False - Disables initializing the Database

Default Value:

True - Enables initializing the Database.

Connection Consumer Configuration

FileDBManager5

Presented below are the details for the parameters present in **Fiorano->mq->connection-consumer->FileDBManager->FileDBManager5**.

Path

The Path parameter allows consumer configuration to be stored in the file based data storage in a specified path.

Valid values:

The default value of the Path parameter is **CONN_CONSUMER**.

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

DeletedThresholdPercent

The DeletedThresholdPercent specifies the entries to be deleted from a cache that stores and manages the list of all the tables pertaining to consumer related information in the file-storage based table. After repeated modification of the table, old entries in the table are marked as deleted. This value provides a threshold percentage of deleted entries after which the Cache Compaction Process is started so that no deleted entries remain in the table. Since cache compaction involves CPU intensive File operations, the DeletedThresholdPercent should be used with DeletedThresholdCount to ensure it is not invoked too often.

Valid values:

The default value of the DeletedThresholdPercent parameter is **50**. This indicates that when the ratio of deleted entries to total entries is equal to or greater than 0.5, the Cache compaction process will be invoked.

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

Note:

All values less than zero or equal to zero retrieve the same result.

All values greater than or equal to 100 retrieve the same result.

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:

The range of integer values considered valid by the PingInterval are -2^{31} to $2^{31} - 1$.

All values less than or equal to zero retrieve the same result.

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. It is best to choose a value that is high enough to trigger a cache compaction.

Dependencies

DeletedThresholdPercent.

DbTableNamePrefix

The DbTableNamePrefix parameter allows Database Names usually prefixed with a string to be modified.

Valid values:

Any String. Default value is #.

Example

If the present state of the Database is to be saved to be used again in the future, modifying the prefix to some other relevant string and changing it back to the original string when it is to be used may be done using the DbTableNamePrefix parameter.

Directory Service

Presented below are the details for the parameters present in **Fiorano->etc->DirectoryService**.

EnableCluster

The EnableCluster parameter specifies whether Cluster Manager is to be started along with the FioranoMQ Server.

Default Value: Yes.

EnableProfileMonitoring

The EnableProfileMonitoring parameter defines whether the configuration changes made to the profiles residing inside the Management Server are to be monitored.

If enabled, all the configurational changes are logged automatically to the file \$DBPATH/logs/profile-monitor.log.

Default Value: no

DirectoryServiceAddress

The DirectoryServiceAddress parameter sets the IPAddress on which the Cluster Manager runs.

Default Value: localhost

DirectoryServicePort

The DirectoryServicePort parameter sets the port on which the Cluster Manager runs.

Default Value: 10389

SecurityPrincipal

The SecurityPrincipal parameter sets the Security Principal required for creating a connection with the Cluster Manager.

Default Value: Uid=admin,ou=system.

SecurityCredentials

The SecurityCredentials parameter sets the Security Credentials required for creating a connection with the Cluster Manager.

Default Value: secret.

AuthenticationMode

The AuthenticationMode parameter sets the authentication mode to be used to connect with the Cluster Manager.

Default Value: simple.

InstancePath

The InstancePath parameter sets the path of the Instance upon which the Cluster Manager will run.

If set to null, the default value set for the working directory is \$FIORANO_HOME/DSInstance/\$PROFILE_NAME.

Default Value: null

SynchPeriodMillis

The SynchPeriodMills parameter sets the synchronization time for the Directory Service in milliseconds.

Default Value: 15000