

git_comments:

1. * * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * <p> * <http://www.apache.org/licenses/LICENSE-2.0> * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
2. * * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * <p> * <http://www.apache.org/licenses/LICENSE-2.0> * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
3. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
4. * * Test Address queue parameters are correctly read.
5. All msgs should now go to the next consumer only, without any errors or exceptions
6. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
7. * * Exclusive Test
8. All msgs should go to the first consumer
9. Last message only should go to the consumer
10. * * LVQ Test
11. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
12. Last message only should go to the other consumer as well.
13. Last message only should go to the consumer.
14. Set the consumer window size to 0 to not buffer any messages client side.
15. * * Creates a temporary queue with a filter. * * @param address the queue will be bound to this address * @param routingType the delivery mode for this queue, MULTICAST or ANYCAST * @param queueName the name of the queue * @param filter only messages which match this filter will be put in the queue * @param maxConsumers how many concurrent consumers will be allowed on this queue * @param purgeOnNoConsumers whether to delete the contents of the queue when the last

- consumer disconnects * @param exclusive if the queue is exclusive queue * @param lastValue if the queue is last value queue * @throws ActiveMQException in an exception occurs while creating the queue
16. * * Creates Shared queue. A queue that will exist as long as there are consumers or is durable. * * @param address the queue will be bound to this address * @param routingType the delivery mode for this queue, MULTICAST or ANYCAST * @param queueName the name of the queue * @param filter whether the queue is durable or not * @param durable if the queue is durable * @param maxConsumers how many concurrent consumers will be allowed on this queue * @param purgeOnNoConsumers whether to delete the contents of the queue when the last consumer disconnects * @param exclusive if the queue is exclusive queue * @param lastValue if the queue is last value queue * @throws ActiveMQException in an exception occurs while creating the queue
 17. * * Creates a non-temporaryqueue. * * @param address the queue will be bound to this address * @param routingType the delivery mode for this queue, MULTICAST or ANYCAST * @param queueName the name of the queue * @param filter only messages which match this filter will be put in the queue * @param durable whether the queue is durable or not * @param autoCreated whether to mark this queue as autoCreated or not * @param maxConsumers how many concurrent consumers will be allowed on this queue * @param purgeOnNoConsumers whether to delete the contents of the queue when the last consumer disconnects * @param exclusive whether the queue should be exclusive * @param lastValue whether the queue should be lastValue * @throws ActiveMQException
 18. * * Creates a non-temporary queue. * * @param address the queue will be bound to this address * @param routingType the delivery mode for this queue, MULTICAST or ANYCAST * @param queueName the name of the queue * @param filter only messages which match this filter will be put in the queue * @param durable whether the queue is durable or not * @param autoCreated whether to mark this queue as autoCreated or not * @param maxConsumers how many concurrent consumers will be allowed on this queue * @param purgeOnNoConsumers whether to delete the contents of the queue when the last consumer disconnects * @param exclusive whether the queue should be exclusive * @param lastValue whether the queue should be lastValue * @throws ActiveMQException
 19. * * Update a queue. * * @param name name of the queue * @param routingType the routing type used for this address, {@code MULTICAST} or {@code ANYCAST} * @param maxConsumers the maximum number of consumers allowed on this queue at any one time * @param purgeOnNoConsumers delete this queue when the last consumer disconnects * @return a textual summary of the queue * @throws Exception
 20. * *
 21. * * Creates Shared queue. A queue that will exist as long as there are consumers or is durable. * * @param address the queue will be bound to this address * @param routingType the delivery mode for this queue, MULTICAST or ANYCAST * @param queueName the name of the queue * @param filter whether the queue is durable or not * @param durable if the queue is durable * @param maxConsumers how many concurrent consumers will be allowed on this queue * @param purgeOnNoConsumers whether to delete the contents of the queue when the last consumer disconnects * @param exclusive if the queue is exclusive queue * @param lastValue if the queue is last value queue * @throws ActiveMQException in an exception occurs while creating the queue
 22. * * Queue parameters;
 23. no-op

git_commits:

1. **summary:** This closes #1832
message: This closes #1832

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ARTEMIS-853 Support for exclusive consumers
body: Support exclusive consumer Allow default address level settings for exclusive consumer Allow queue level setting in broker.xml Add the ability to set queue settings via Core JMS using address. Similar to ActiveMQ 5.X Allow for Core JMS client to define exclusive consumer using address parameters Add tests

github_pulls_comments:

1. @gtully ping. I will add docs in another PR later. Like wise will sort OpenWire to be able to set client side after(if) this gets merged. Wanted to sort supporting this on core (client and config) as first step, as a single change, to avoid it becoming a mammoth PR.
2. **body:** @michaelandrepearce to make the feature work is trivial in queueimpl; to make it dynamic is a lot of code. I would have been shooting for an address setting and pulling that value in queueImpl on creation. An all or nothing approach. But what you have is very comprehensive :-). I just wonder if there really is a need to have the consumer create an lvq or exclusive queue. If there is, then this nails it. For an existing openwire client the value can come from a consumer info or from a destination parameter but mostly it is the effect that is needed, not the fact that it is dynamic, ie: Because it effects every consumer having a single consumer control it is a bit imbalanced and not typical I think.
label: code-design
3. @gtully yes there is a use case for the client part here in our org, we have dynamic non durable shared consumers, where one pair of shared consumers needs it to be exclusive, and the other pair don't, thus why. and why i added that in the test case. I'd rather do this fully, than partial and have to come back to it, thus it being comprehensive. @mtaylor or anyone(PMC) with access, can someone look at the build system, ive tried 10 times, and keep getting build fails with the below which seems infra related. ERROR: Error fetching remote repo 'origin' hudson.plugins.git.GitException: Failed to fetch from git://github.com/apache/activemq-artemis.git
4. @michaelandrepearce Hey Michael. I think this was a temp failure. There is an Artemis PR job running now that doesn't show the git clone problem. The CI is shared and there are a bunch of jobs queued so we might have to wait a while to run again.
5. @gtully you happy if this is merged (I can do or I can't leave for you) or you still reviewing? Just waiting on this before I continue with the other bits as mentioned
6. @michaelandrepearce @gtully @mtaylor If everything is fine I will going to merge it, ok?
7. @franz1981 seems no objections (i addressed gtully's comment on rationalising fyi) , you still ok to merge?
8. @michaelandrepearce i don't have the chance to merge it yet :(Anyway it seems that there aren't any objections so I suppose it is ok 🍑
9. @michaelandrepearce I think the logic should have been updated on QueueImpl for the exclusive consumer... You should just had the pos always 0. or have a different logic without using counters at all. The code became non consistent. the pos increases but then there is a statement to always get position 0 if exclusive.
10. We do the same as if message group which doesn't increase the pos. If it's exclusive we don't increment the pos like with message group We can enhance this if needed. Ping me on IRC tomorrow? happy to make further enhancements if wanted.

github_pulls_reviews:

jira_issues:

1. **summary:** Support for exclusive consumers
description: Artemis should support a consumer feature where a single consumer receives all messages, even when multiple consumers are present. This capability maintains message ordering while allowing a HA consumer. ActiveMQ 5.x supports this, as does IBM MQ, Tibco EMS, etc.

jira_issues_comments:

1. [~jbertram] / [~martyntaylor] wasn't this added for the new addressing. MaxConsumes=1?
2. if MaxConsumes=1 did not error with an exception and retained the consumer as pending it would be like the exclusive consumer. With the exclusive consumer feature, the second or low priority consumer is valid, just not getting any messages. Only when the first consumer goes away does dispatch fallback to the second consumer. MaxConsumes=1 could work but the client will then have to deal with retries and timeout and error logs filling up etc. The original (way back) 5.x use case came from having a HA cluster of active ejb servers with mdb's pointing to the broker, only one should get messages.
3. 5.x has: allConsumersExclusiveByDefault destination policy[1]; if we implement that on a queue, then we cover 90% of the use cases. It does not need to be dynamic, ie: initiated by the first exclusive

consumer... mixing exclusive and not exclusive is not the norm. [1] <http://activemq.apache.org/per-destination-policies.html>

4. Github user michaelandrepearce opened a pull request: <https://github.com/apache/activemq-artemis/pull/1832> ARTEMIS-853 Support for exclusive consumers Support exclusive consumer Allow default address level settings for exclusive consumer Allow queue level setting in broker.xml Add the ability to set queue settings via Core JMS using address. Similar to ActiveMQ 5.X Allow for Core JMS client to define exclusive consumer using address parameters Add tests You can merge this pull request into a Git repository by running: `$ git pull https://github.com/michaelandrepearce/activemq-artemis ARTEMIS-853` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/activemq-artemis/pull/1832.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1832 ---- commit 87acc92e71fe63b6a252bab38d4aba68242a4c04 Author: Michael André Pearce <michael.andre.pearce@...> Date: 2018-01-30T17:43:14Z ARTEMIS-853 Support for exclusive consumers Support exclusive consumer Allow default address level settings for exclusive consumer Allow queue level setting in broker.xml Add the ability to set queue settings via Core JMS using address. Similar to ActiveMQ 5.X Allow for Core JMS client to define exclusive consumer using address parameters Add tests ----
5. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @gtully ping.
6. Github user gtully commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @michaelandrepearce to make the feature work is trivial in queueimpl; to make it dynamic is a lot of code. I would have been shooting for an address setting and pulling that value in queueImpl on creation. An all or nothing approach. But what you have is very comprehensive :-)) I just wonder if there really is a need to have the consumer create an lvq or exclusive queue. If there is, then this nails it. For an existing openwire client the value can come from a consumer info or from a destination parameter but mostly it is the effect that is needed, not the fact that it is dynamic, ie: Because it effects every consumer having a single consumer control it is a bit imbalanced and not typical I think.
7. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @gtully yes there is a use case for the client part here in our org, thus why. I'd rather do this fully, than partial. @mtaylor or anyone with access, can someone look at the build system, ive tried 10 times, and keep getting build fails with the below which seems infra related. ERROR: Error fetching remote repo 'origin' hudson.plugins.git.GitException: Failed to fetch from <git://github.com/apache/activemq-artemis.git>
8. Github user mtaylor commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @michaelandrepearce Hey Michael. I think this was a temp failure. There is an Artemis PR job running now that doesn't show the git clone problem. The CI is shared and there are a bunch of jobs queued so we might have to wait a while to run again.
9. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @gtully you happy if this is merged? Just waiting on this before I continue with the other bits as mentioned
10. Github user franz1981 commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @michaelandrepearce @gtully @mtaylor If everything is fine I will going to merge it, ok?
11. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @franz1981 seems no objections, you still ok to merge?
12. Github user franz1981 commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @michaelandrepearce i don't have the chance to merge it yet :(Anyway it seems that there aren't any objections so I suppose it is ok 🍀
13. Commit 47c9a90dcb16b70aeef3a09c89400669add083f5 in activemq-artemis's branch refs/heads/master from [~michael.andre.pearce] [<https://git-wip-us.apache.org/repos/asf?p=activemq-artemis.git;h=47c9a90>] ARTEMIS-853 Support for exclusive consumers Support exclusive consumer Allow default address level settings for exclusive consumer Allow queue level setting in broker.xml Add the ability to set queue settings via Core JMS using address. Similar to ActiveMQ 5.X Allow for Core JMS client to define exclusive consumer using address parameters Add tests
14. Commit 5dae8ca941d94bb6ae6b7f20876b8afb7cc953f in activemq-artemis's branch refs/heads/master from [~michael.andre.pearce] [<https://git-wip-us.apache.org/repos/asf?p=activemq-artemis.git;h=5dae8ca>] ARTEMIS-853 Support for exclusive consumers Rationalise and re-use URISupport.
15. Github user asfgit closed the pull request at: <https://github.com/apache/activemq-artemis/pull/1832>
16. Commit dc41f3ca491e96e199290a225fdaa07ac05d66df in activemq-artemis's branch refs/heads/master from [~michael.andre.pearce] [<https://git-wip-us.apache.org/repos/asf?p=activemq-artemis.git;h=dc41f3c>

-] ARTEMIS-853 Support for exclusive consumers Support exclusive consumer Allow default address level settings for exclusive consumer Allow queue level setting in broker.xml Add the ability to set queue settings via Core JMS using address. Similar to ActiveMQ 5.X Allow for Core JMS client to define exclusive consumer using address parameters Add tests
17. Commit 38c45c92142cd6a3b9f49f2d436d7a0665309c00 in activemq-artemis's branch refs/heads/master from [~michael.andre.pearce] [<https://git-wip-us.apache.org/repos/asf?p=activemq-artemis.git;h=38c45c9>] ARTEMIS-853 Support for exclusive consumers Rationalise and re-use URISupport.
 18. Github user clebertsuconic commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> @michaelandrepearce I think the logic should have been updated on QueueImpl for the exclusive consumer... You should just had the pos always 0. or have a different logic without using counters at all. The code became non consistent. the pos increases but then there is a statement to always get position 0 if exclusive.
 19. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1832> We do the same as if message group which doesn't increase the pos if it's exclusive just like it doesn't if exclusive. We can enhance this if needed. Ping me on IRC tomorrow? happy to make further enhancements if needed.
 20. GitHub user michaelandrepearce opened a pull request: <https://github.com/apache/activemq-artemis/pull/1856> ARTEMIS-853 Support for exclusive consumers Alternative QueueImpl - Seperated logic without counters You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/michaelandrepearce/activemq-artemis> ARTEMIS-853-2 Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/activemq-artemis/pull/1856.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1856 ---- commit 87ecda7ed5b09774c23f3de16bff2b20971db015 Author: Michael André Pearce <michael.andre.pearce@...> Date: 2018-02-07T21:07:44Z ARTEMIS-853 Support for exclusive consumers Alternative QueueImpl - Seperated logic without counters ----
 21. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1856> @clebertsuconic as per your comment to look at an alternative QueueImpl with it having separate flow without counters for exclusive. IMO i don't think this is a better approach, but i want to offer it to you. There is an amount of logic that is almost similar but not re-usable or abstract-able (ive done what i could). This is why i followed the same logic that MessageGroups does when a group already is assigned, which is to override the consumer and then don't update the pos, as it made for a lot smaller change, less intrusive change following a pattern already done with msg groups. Anyhow if you prefer this then im not dead opposed to this approach just don't think its better, but i offer it for your review.
 22. GitHub user michaelandrepearce opened a pull request: <https://github.com/apache/activemq-artemis/pull/1858> ARTEMIS-853 - Support for exclusive consumers Fix coverity findings. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/michaelandrepearce/activemq-artemis> ARTEMIS-853-coverity Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/activemq-artemis/pull/1858.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1858 ---- commit b5288982ca11475f177793b5c4b3abdc21154b6c Author: Michael André Pearce <michael.andre.pearce@...> Date: 2018-02-08T23:51:04Z ARTEMIS-853 - Support for exclusive consumers Fix coverity findings. ----
 23. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1858> @jdanekrh these are the coverity fixes i believe are related to the commit you commented on, as noted the others i believe are due to other commits/pr's that simply you're picking up in the report, or as you noted its fine and i'd agree. I will merge as soon as PR build is green.
 24. Github user clebertsuconic commented on the issue: <https://github.com/apache/activemq-artemis/pull/1856> I will need to look on this on Monday. Will be traveling for a few days.
 25. Commit 1ddeb065f1875cca920e5408a0f606de952d8b19 in activemq-artemis's branch refs/heads/master from [~michael.andre.pearce] [<https://git-wip-us.apache.org/repos/asf?p=activemq-artemis.git;h=1ddeb06>] ARTEMIS-853 - Coverity fixes Fix coverity findings.
 26. Github user asfgit closed the pull request at: <https://github.com/apache/activemq-artemis/pull/1858>
 27. GitHub user michaelandrepearce opened a pull request: <https://github.com/apache/activemq-artemis/pull/1880> ARTEMIS-853 - Add documents Exclusive Queue documents added to detail behaviour and how to configure. Also update docuents for last-value queue to cover additional ways to configure You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/michaelandrepearce/activemq-artemis> ARTEMIS-853-Doc Alternatively you can

review and apply these changes as the patch at: <https://github.com/apache/activemq-artemis/pull/1880.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #1880 ---- commit 03f4a52f1f1de52f4827321ede96437ff57c2a03 Author: Michael André Pearce <michael.andre.pearce@...> Date: 2018-02-20T08:39:37Z ARTEMIS-853 - Add documents Exclusive Queue documents added to detail behaviour and how to configure. Also update documents for last-value queue to cover additional ways to configure ----

28. Github user asfgit closed the pull request at: <https://github.com/apache/activemq-artemis/pull/1880>
29. Github user michaelandrepearce commented on the issue: <https://github.com/apache/activemq-artemis/pull/1856> I'm closing this as no further comments.
30. Github user michaelandrepearce closed the pull request at: <https://github.com/apache/activemq-artemis/pull/1856>