Subscription Recovery Policy

The subscription recovery policy allows you to go back in time when you subscribe to a topic.

For example imagine you are processing a price feed; you're using a federated network and either a network glitch occurs or someone kills the broker you're talking to. If you reconnect to another broker in the cluster you may have lost messages.

So we support a timed or fixed size recovery buffer so that if you reconnect to another broker within some time period (depending on volume & RAM this could be 30 seconds to 5 minutes), then any messages you missed during the downtime are redelivered before new messages are delivered to you.

For more information see Retroactive Consumer

Last image caching

Its often common in financial market data type worlds to want to know the latest price of say IBM stock along with get all the future updates to the price. Historically you often had a request-reply snapshot quote service for the latest price, then you subscribe to a topic for updates. The issue is the client then has 2 APIs / middlewares to deal with - often quite different things - plus you have an ordering issue (race condition) - the update could beat the last price request so you can get out of order (going back in time to an old price, which could be very old).

One of our *subscription recovery policy* implementations is called **Last Image Subscription Policy** which will ensure that when you subscribe to a topic (say PRICES.NASDAQ.IBM), you will receive the last image (the last message that was sent on that topic) plus any updates which occur in the future, with ordering to ensure that the last image is always first before any new messages arrive.

A common problem in market data type situations is that you may have a cache of last image prices, then a feed of new price changes; if you request the last price from the cache and subscribe to new prices; depending on how you do it you can either, miss an update or receive a newer update before the old last image arrives (so either get out of order messages).

Note that you can configure the subscription recovery policy, and most other policies on different destinations or wildcards. So you may use last image policy for prices on topics only and use a buffered fixed size policy for other notifications on different topics etc.

Summary of Available Recovery Policies

Policy Name	Sample Configuration	Description
FixedSizedSubscriptionRecoveryPolicy	<fixedSizedSubscriptionRecoveryPolicy maximumSize="1024"/>	Keep a fixed amount of memory in RAM for message history which is evicted in time order.
FixedCountSubscriptionRecoveryPolicy	<fixedCountSubscriptionRecoveryPolicy maximumSize="100"/>	Keep a fixed count of last messages.
LastImageSubscriptionRecoveryPolicy	<lastImageSubscriptionRecoveryPolicy/>	Keep only the last message.
NoSubscriptionRecoveryPolicy	<noSubscriptionRecoveryPolicy/>	Disables message recovery.
QueryBasedSubscriptionRecoveryPolicy	<queryBasedSubscriptionRecoveryPolicy query="JMSType = 'car' AND color = 'blue'"/>	Perform a user specific query mechanism to load any message they may have missed. Details on message selectors are available here: http://java.sun.com/j2ee/1.4/docs/api/javax/jms/Message.html
TimedSubscriptionRecoveryPolicy	<timedSubscriptionRecoveryPolicy recoverDuration="60000" />	Keep a timed buffer of messages around in memory and use that to recover new subscriptions. Recovery time is in milliseconds.
RetainedMessageSubscriptionRecoveryPolicy	<retainedMessageSubscriptionRecoveryPolicy />	Keep the last message with ActiveMQ.Retain property set to true