

OpenArm Transaction MessagingMediator

1. MessagingMediator

This class outputs the information for a given transaction as JMS messages.

2. MessagingMediator Properties

Regardless of **how** it is configured, the MessagingMediator has the following properties, which must be set:

- *queueConnectionFactoryName* : the name of the JMS QueueConnectionFactory that should be used
- *queueName* : the name of the JMS queue that the Mediator will send messages to. The Mediator will try to create this queue, on the fly, if it does not exist.
- *jndiInitialContextProperties* : an instance of a java.util.Properties object, containing values for various JNDI/JMS properties, such as Context.INITIAL_CONTEXT_FACTORY or Context.PROVIDER_URL. This is implementation dependent, and you may or may not need to use this property. The value set here will be passed as the `Hastable environment` parameter to the constructor of `InitialContext`, used internally in the Mediator to get started with JMS.
- *transactionLengthInterval* : the time interval that you want as a threshold that must be exceeded before a message is sent
- *updateLengthInterval* : the same sort of threshold for updates
- *isBlockedEventTrap* : a boolean flag that determines whether or not blocking events should be forwarded as messages

3. Dependency Injection Configuration

This class is accompanied by a so-called Mediator Configuration class -- `MessagingMediatorConfiguration` -- which you can use to configure the Mediator programmatically. The properties of the configuration class correspond to the properties of the Mediator. Here is an example, which configures the Mediator to use ActiveMQ as the JMS implementation:

```
return new OpenArmConfiguration() {
```

```
public Map getMediatorConfigurations() {
    final Map result = new HashMap();
    final Properties jndiInitialContextProperties = new Properties();
    jndiInitialContextProperties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
        "org.activemq.jndi.ActiveMQIni
jndiInitialContextProperties.setProperty(Context.PROVIDER_URL, "tcp://1
result.put("net.m2technologies.open_arm.transport.transaction.messaging
        new MessagingMediatorConfiguration("QueueConnectionFactory",
                                           2000,
                                           5000,
                                           true,
                                           "myQ",
                                           jndiInitialContextPropert
    return result;
}; }
```

The sample above also shows the default values that this class will use for each of these attributes, if you do not provide a properties file, or if you omit one or more of the attributes.

4. Declarative Configuration

The (deprecated) declarative, file-based configuration mechanism is not supported for this Mediator.