CREATING PUBLISH-SUBCRIBE ENVIRONMENT IN AIX

TABLE OF CONTENTS

Definitions of different terms

Before going into the details of pub-sub environment creation, we will discuss brief definitions of different terms which need to be known and clarified before the exercise.

JMS provider

An implementation of the JMS interface for a Message Oriented Middleware (MOM). Providers are implemented as either a Java JMS implementation or an adapter to a non-Java MOM.

JMS client

An application or process that produces and/or receives messages.

JMS producer

A JMS client that creates and sends messages.

JMS consumer

A JMS client that receives messages.

JMS message

An object that contains the data being transferred between JMS clients.

JMS queue

A staging area that contains messages that have been sent and are waiting to be read. Note that, contrary to what the name queue suggests, messages don't have to be delivered in the order sent. If the message driven bean pool contains more than one instance then messages can be processed concurrently and thus it is possible that a later message is processed sooner than an earlier one. A JMS queue guarantees only that each message is processed only once.

JMS topic

A distribution mechanism for publishing messages that are delivered to multiple subscribers.

After these basic definitions, the next question which may arise in our mind should be about the publish/subscribe model. So lets see some more details.

Communication Models

The JMS API supports two models:

- Point-to-Point or queuing model
- Publish and subscribe model

Point-to-Point model

In the **point-to-point or queuing model**, a sender posts messages to a particular queue and a receiver reads messages from the queue. Here, the sender knows the destination of the message and posts the message directly to the receiver's queue. It is characterized by the following:

- Only one consumer gets the message
- The producer does not have to be running at the time the consumer consumes the message, nor does the consumer need to be running at the time the message is sent
- Every message successfully processed is acknowledged by the consumer

What is Publish/Subscribe model?

The publish/subscribe model supports publishing messages to a particular message topic. Subscribers may register interest in receiving messages on a particular message topic. In this model, neither the publisher nor the subscriber know about each other. A good analogy for this is an anonymous bulletin board. The following are characteristics of this model:

- Multiple consumers can get the message
- There is a timing dependency between publishers and subscribers. The publisher has to create a subscription in order for clients to be able to subscribe. The subscriber has to remain continuously active to receive messages, unless it has established a durable subscription. In that case, messages published while the subscriber is not connected will be redistributed whenever it reconnects.

How to create a pub/sub environment in AIX

Following are the quick steps to create a pub/sub environment in AIX platform.

Step 1: Create a Queue Manager QM1

```
DEFINE QMGR(QM1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE)
```

Step 2: Create Websphere MQ Objects (Related to the Client Application)

```
DEFINE QLOCAL(DEMO.QUEUE1)
DEFINE QLOCAL(DEMO.BestPrice)
DEFINE QMODEL(DEMO.MODEL1)
```

Define a Server Connection Channel

```
DEFINE CHANNEL(DEMO.JAVA.CHANNEL) CHLTYPE(SVRCONN) DESCR('Server Connection
Channel for JMS Demo')
```

Step 3: Publish - Subscribe Setup

Start the MQ Broker

```
strmqbrk -m QM1
```

to verify MQ Broker (Publish/Subscribe) is running,

```
dspmqbrk -m QM1
```

Step 4: Create JMS System Queues (Script)

For WebSphere MQ JMS Publish/Subscribe implementation to work correctly, a number of system queues must be created. To create the objects, enter:

```
runmqsc QM1 < <mq_install_path>\java\bin\MQJMS_PSQ.mqsc
```

(I've attached the script in Appendix A)

Step 5: Create JMS Administered Objects (file system)

Make necessary changes to <mq_install>/java/bin/JMSAdmin.config file by specifying the correct Context Factory and the JNDI Provider url. For this example, we have used the File based JNDI Provider.

-Run the JMSAdmin Tool

```
<mq_install>/java/bin/JMSAdmin -v
```

This will start the JMSAdmin tool. Now you may create or manage your desired JMS objects using this JMSAdmin tool.

For help in defining the JMS Administered objects (See Appendix B - *JMS Admin Tool Script*)

Appendix A - JMS System Queues Creation Script

```
************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Administration Queue                         */
*                                                                      */
************************************************************************/
* Define a local queue
DEFINE QLOCAL('SYSTEM.JMS.ADMIN.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - admin queue') +
        DEFPSIST(YES)  +
        NOSHARE

************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Subscriber Status Queue                      */
*                                                                      */
************************************************************************/
    DEFINE QLOCAL('SYSTEM.JMS.PS.STATUS.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - PS status queue') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED)

************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Report Queue                                 */
*                                                                      */
************************************************************************/
    DEFINE QLOCAL('SYSTEM.JMS.REPORT.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - Report queue') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED)
```

```
************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Subscribers Model Queue                      */
*                                                                      */
*   Create model queue used by subscribers to create a permanent       */
*   queue for subsciptions                                             */
*                                                                      */
************************************************************************/
*   General reply queue                                                */
    DEFINE QMODEL('SYSTEM.JMS.MODEL.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - Model queue') +
        DEFTYPE(PERMDYN) +
        SHARE DEFSOPT(SHARED)

************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Default Non-Durable Shared Queue             */
*                                                                      */
*   Create local queue used as the default shared queue by             */
*   non-durable subscribers                                            */
*                                                                      */
************************************************************************/
**   Create a local queue
    DEFINE QLOCAL('SYSTEM.JMS.ND.SUBSCRIBER.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - PS ND shared queue') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED) +
        MAXDEPTH(100000)
```

```
************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Default Non-Durable Shared Queue for         */
*   ConnectionConsumer functionality                                   */
*                                                                      */
*   Create local queue used as the default shared queue by             */
*   non-durable connection consumers                                   */
*                                                                      */
************************************************************************/
**   Create a local queue
    DEFINE QLOCAL('SYSTEM.JMS.ND.CC.SUBSCRIBER.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - PS ND CC shared q') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED)  +
        MAXDEPTH(100000)

************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Default Durable Shared Queue                 */
*                                                                      */
*   Create local queue used as the default shared queue by durable     */
*   subscribers                                                        */
*                                                                      */
************************************************************************/
**   Create a local queue
    DEFINE QLOCAL('SYSTEM.JMS.D.SUBSCRIBER.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - PS D shared queue') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED)  +
        MAXDEPTH(100000)
```

```
************************************************************************/
*                                                                      */
*   JMS Publish/Subscribe Default Durable Shared Queue for             */
*   ConnectionConsumer functionality                                   */
*                                                                      */
*   Create local queue used as the default shared queue by durable     */
*   connection consumers                                               */
*                                                                      */
************************************************************************/
**   Create a local queue
    DEFINE QLOCAL('SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE') REPLACE +
        DESCR('Websphere MQ - JMS Classes - PS D CC shared q') +
        DEFPSIST(YES)  +
        SHARE DEFSOPT(SHARED)  +
        MAXDEPTH(100000)
```

APPENDIX B - JMS Admin Tool Scripts for Pub-Sub Environment

1. Create a Context and Move to it:

```
DEFINE CTX(DEMO)
CHANGE CTX(DEMO)
```

You will get a prompt like as follows:

```
InitCtx/DEMO>
```

2. Define a Server ConnectionFactory

```
DEFINE CF(DemoServerCF) QMANAGER(QM1) TRAN(BIND) TEMPMODEL(DEMO.MODEL1)
```

3. Define a Client ConnectinFactory

```
DEFINE CF(DemoClientCF) QMANAGER(QM1) TRAN(CLIENT) TEMPMODEL(DEMO.MODEL1)
HOSTNAME(172.26.150.2) PORT(1430) CHANNEL(DEMO.JAVA.CHANNEL)
```

(If the listener for a port is not defined at qmanager)

```
define listener(LISTENER.FOR.JMS) TRPTYPE(TCP) PORT(1430)
START LISTENER(LISTENER.FOR.JMS)
```

4. Define a QueueConnectionFactory

```
DEFINE QCF(DemoClientQCF) TRAN(CLIENT)
```

5. Topic Connection Factory

```
DEFINE TCF(DemoTCF)TRANSPORT(CLIENT) QMANAGER(QM1) HOST(172.26.150.2)
PORT(1430)
```

5. Define a JMS Queue

```
DEFINE Q(Queue1) QUEUE(DEMO.QUEUE1) QMANAGER(QM1)
```

6. Define Topic

```
DEFINE T(BestPriceRequest) TOPIC(DEMO/BestPriceRequest)
DEFINE T(BestPriceReply) TOPIC(DEMO/BestPriceReply)
```

References

- Wikipedia
  URL: http://en.wikipedia.org/wiki/Java_Message_Service

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10