

Calculating the delay time for a message in SYSTEM.DDELAY.LOCAL.QUEUE in an IBM MQ queue manager

<https://www.ibm.com/support/pages/node/7151033>

Date last updated: 10-May-2024

Angel Rivera
IBM MQ Support

<https://www.ibm.com/products/mq/support>
Find all the support you need for IBM MQ

+++ Objective +++

In your MQ JMS Client application you are using the JMS Delay feature in IBM MQ, which temporarily stores the message in the internal queue SYSTEM.DDELAY.LOCAL.QUEUE and now you want to check how much time left is in the delayed message before it is released from that internal queue.

+++ ANSWER +++

Caveat! There are many steps involved!!

The message that is placed in the internal queue will have a DDH (Delivery Delay) header.

Our colleague Mahesh P. mentioned that this is the internal structure for the DDH header:

```
typedef struct _DDH {
    MQCHAR4  strucId;          /* "DDH " */
    MQLONG   version;         /* 1 */
    /* Time in milliseconds until delivery delay completes relative to */
    /* OS epoch. */
    MQUINT64 delayComplete;
    MQUINT64 Expiry;          /* Appl expiry time */
    MQCHAR48 qName;           /* Target queue name */
    MQCHAR48 qmgrName;        /* Target queue manager name */
}
```

Thus, the Expiry time (inside the SYSTEM.DDELAY.LOCAL.QUEUE) starts from the 9th byte in the amqsbcbg output.

In my Windows test system, I have an MQ JMS sample that explicitly invokes the Delivery Delay

Directory: C:\angel\coding\MQ\JMS2-DeliveryDelay
File: JmsProducerDeliveryDelay

C:\angel\coding\MQ\JMS2-DeliveryDelay> time

The current time is: 13:07:57.20 (ET)

I ran the test sample:

C:\angel\coding\MQ\JMS2-DeliveryDelay> runmqjms JmsProducerDeliveryDelay -m QMWIN -d Q1

Enter some text to be sent in a message <ENTER to finish>:TEST-DELIVERY-DELAY

Sent message:

JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 2
JMSDeliveryDelay: 300000
JMSDeliveryTime: 1699117992379
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120514d57494e202020202020c630466501e70240
JMSTimestamp: 1699117692379
JMSCorrelationID: null
JMSDestination: queue:///Q1
JMSReplyTo: null
JMSRedelivered: false
JMSXAppID: JmsProducerDeliveryDelay
JMSXDeliveryCount: 0
JMSXUserID: xx
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20231104
JMS_IBM_PutTime: 17081239

TEST1

Enter some text to be sent in a message <ENTER to finish>:

SUCCESS

Notice the following lines:

JMSDeliveryMode: 2
JMSDeliveryDelay: 300000
JMSDeliveryTime: 1699117992379

...

JMSXAppID: JmsProducerDeliveryDelay
JMSXDeliveryCount: 0

JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20231104
JMS_IBM_PutTime: 17081239

Let's use this Epoch Unix Converter:
<https://www.epochconverter.com/>

... and enter the decimal for JMSDeliveryTime:
1699117992379

The result is:

Assuming that this timestamp is in milliseconds:
GMT: Saturday, November 4, 2023 5:13:12.379 PM
Your time zone: Saturday, November 4, 2023 1:13:12.379 PM GMT-04:00 DST

Because the MQ JMS Client application does not show explicitly the:
JMSDeliveryTime: 1699117992379
... then the question is:
Is it possible to look at the JMSDeliveryTime from amqsbcg?

The answer is:

Yes, but it requires some extra steps.

Let's issue:

`amqsbcg SYSTEM.DDELAY.LOCAL.QUEUE QMWIN > ambsbcg.txt`

Then get the 1st line of the Payload/Contents, which has the DDH header.

```
**** Message ****
length - 701 of 701 bytes
00000000: 4444 4820 0100 0000 123F 5EBC 5609 0600 'DDH .....?^.V ..'
                                *****
Byte position:                9 .. 12
```

The time stamp starts in the 9th byte.

Note:

For more details on the swapping of the bytes, please see:

<https://www.ibm.com/support/pages/node/7151034>

How to do a manual byte swapping from the output of IBM MQ amqsbcg in Windows and Linux x86 to find out the DLQ reason code

Because it is an Intel system, it is necessary to swap the bytes and remove the spaces:

```
123F 5EBC 5609 0600 => 00060956BC5E3F12
                        ++      ++
                        ++      ++
                    ++      ++
                ++      ++
            ++      ++
        ++      ++
    ++      ++
++      ++
```

Finally, let's add the prefix that indicates that it is an Hexadecimal number: 0x
`0x00060956BC5E3F12`

Thus, the Hex of the swapped bytes is shown below, by using the following Hex to Decimal Converter:

<https://www.rapidtables.com/convert/number/hex-to-decimal.html>

Let's enter the hexadecimal number
`0x00060956BC5E3F12`

The Decimal number is:
`1699117992394514`

Now let's use the Epoch Unix converter:
<https://www.epochconverter.com/>

Enter the decimal:
`1699117992394514`

The result is:

`Assuming that this timestamp is in microseconds (1/1,000,000 second):
GMT: Saturday, November 4, 2023 5:13:12.394 PM
Your time zone: Saturday, November 4, 2023 1:13:12.394 PM GMT-04:00 DST`

OK. Now we have the time when the message will be released from the SYSTEM.DDELAY.LOCAL.QUEUE queue.

But what is the actual Delivery Delay time?

We can compare it with the time stamp when the JMS Client application did the actual put into the queue manager.

This can be found in the output of amqsbcbg, in the following lines.

```
00000190: 4A6D 7350 726F 6475 6365 7244 656C 6976 'JmsProducerDeliv'  
000001A0: 6572 7944 656C 6179 2020 2020 3230 3233 'eryDelay 2023'  
000001B0: 3131 3034 3137 3038 3132 3339 2020 2020 '110417081239  '
```

Let's put the text that is shown split in 3 lines above, into a single long text:

```
JmsProducerDeliveryDelay 2023110417081239
```

Let's add some separators for easier reading:

```
2023-11-04 17:08:12:39
```

The time stamp is: 2023-11-04 17:08 GMT or 1:08 PM GMT

Thus the difference between the Put application time:

```
2023-11-04 17:08 GMT or 1:08 PM GMT-04:00 DST  
****
```

... and the JMSDeliveryTime

```
Saturday, November 4, 2023 1:13:12.394 PM GMT-04:00 DST  
****
```

That is between 1:08 and 1:13 is 5 minutes.

+++ end +++