

Unit 17. Message tracking

What this unit is about

This unit covers the way in which a queue manager can track a message using the trace route facility. Application program report options are covered, as well as the use of the trace route utility.

What you should be able to do

After completing this unit, you should be able to:

- Describe how WebSphere MQ tracks a message that has been tagged for tracing
- Explain to programmers how to initiate message tracking

How you will check your progress

- Checkpoint questions
- Lab exercise

References

WebSphere MQ V7 Information Center

Unit objectives

After completing this unit, you should be able to:

- Describe how WebSphere MQ keeps track of a message that has been tagged for tracing
- Explain to programmers how to initiate message tracking

© Copyright IBM Corporation 2008

Figure 17-1. Unit objectives

WM203 / VM2032.0

Notes:

17.1.Lost messages

Lost messages topic objectives

After completing this topic, you should be able to:

- List places where messages can be held up in transit to their intended destination
- Describe situations where messages are not or no longer available in normal processing

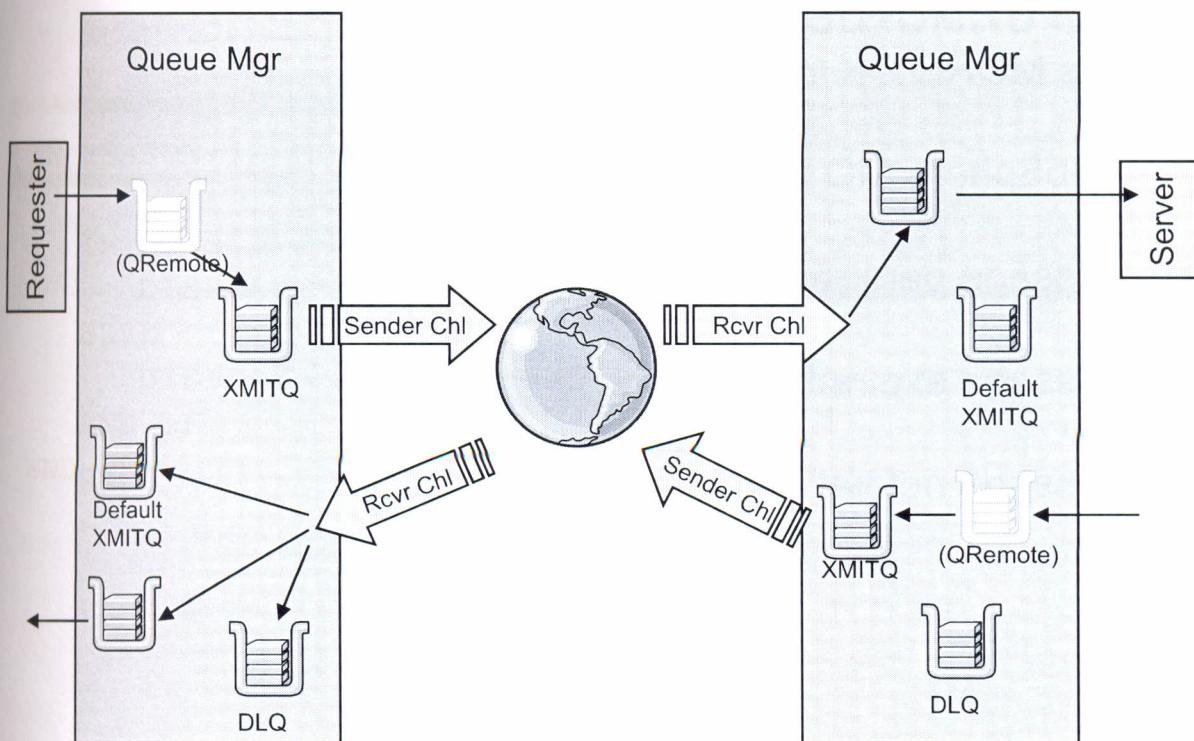
© Copyright IBM Corporation 2008

Figure 17-2. Lost messages topic objectives

WM203 / VM2032.0

Notes:

Where is the message?



© Copyright IBM Corporation 2008

Figure 17-3. Where is the message?

WM203 / VM2032.0

Notes:

There are many places a message might end up after a successful MQPUT or MQPUB. Particularly when it must cross systems.

Some things to remember: WebSphere MQ does not lose messages by design. But messages can go amiss for various reasons.

Missing message checklist

1. MQPUT/MQ PUB return code:
was MQRC OK ?
2. WebSphere MQ network reasons
3. Message persistence
4. Message expired
5. Message not yet committed

© Copyright IBM Corporation 2008

Figure 17-4. Missing message checklist

WM203 / VM2032.0

Notes:

Step one - Checking return codes

```
IF REASON IS NOT EQUAL TO MQRC-NONE
```

```
    IF REASON IS EQUAL TO MQRC_Q_FULL  
        DISPLAY 'Queue contains max number of messages'  
    ELSE  
        DISPLAY 'MQPUT ended with reason code ' REASON  
    END-IF  
END-IF.
```

© Copyright IBM Corporation 2008

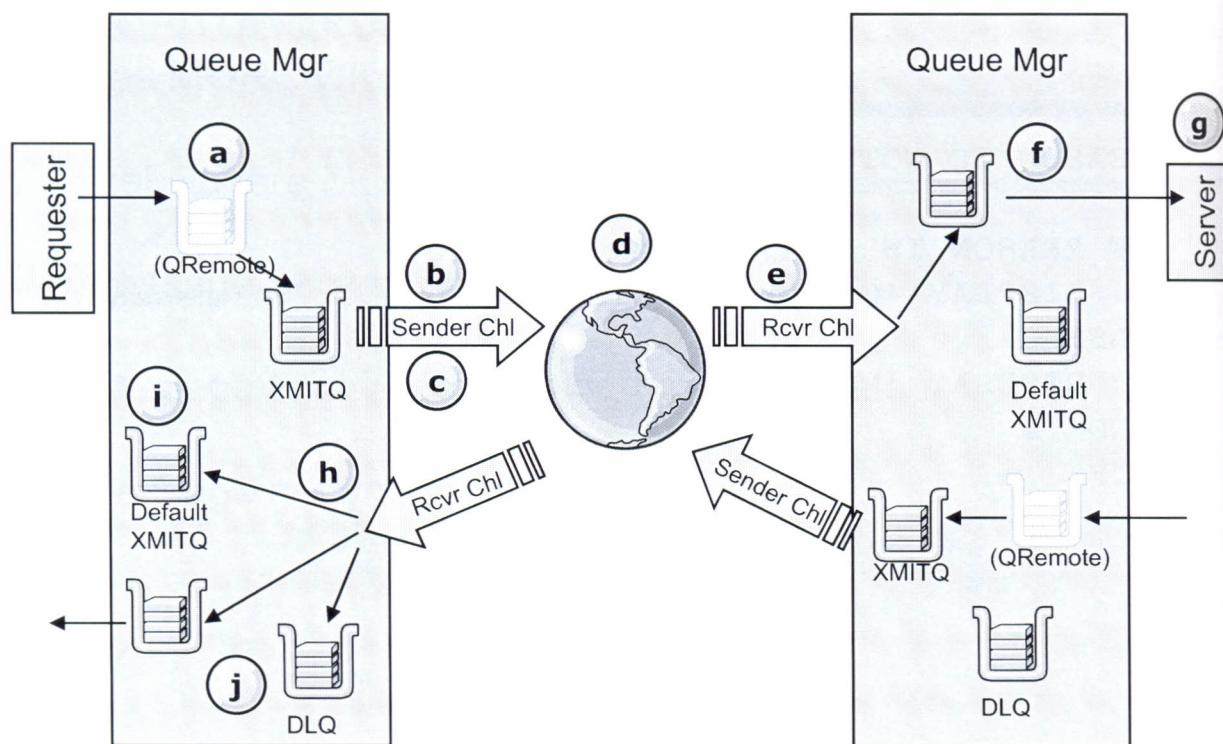
I2032.0 Figure 17-5. Step one - Checking return codes

WM203 / VM2032.0

Notes:

Best practice is for applications to always check the MQI call return codes for errors and react appropriately.

Step two – WebSphere MQ network reasons



© Copyright IBM Corporation 2008

Figure 17-6. Step two – WebSphere MQ network reasons

WM203 / VM2032.0

Notes:

Potential failure points:

1. Incoming message put to the transmit queue (xmitq) but xmitq is put/get disabled, or not defined with usage xmitq.
2. Sender channel is not running or was not triggered to start, or cannot start because xmitq is “get” disabled or at MAXCHL limit.
3. The channel initiator is not running.
4. The channel failed to start due to a network problem.
5. Receiver channel could not be started because listener is not running, or receiver failed to put to a target queue because target queue is put disabled or full. Maybe the channel initiator or event the queue manager is not running.
6. The server program could not get the message because the target queue is “get” disabled.
7. The server program is not started or failed to be triggered.

8. The receiver channel received a message for another remote queue manager, but there is no transmission queue defined for it, so the receiver channel put the message to the default transmit queue.
9. There is no channel defined to serve the default transmit queue.
10. The receiver channel received an inbound message for an unknown local target queue, or the local target queue was full, and so was put to the dead letter queue (DLQ).

The same failure points are present in both directions. This list is not exhaustive.

Step three - Message persistence

DEFPSIST(YES) ?

The screenshot shows a Windows application window titled "Message browser". Inside, it displays message details for a queue named "ANDREW.TEST.PERSISTENT". The window includes fields for "Queue Manager Name" (Canberra1) and "Queue Name". A table lists two messages:

Position	Persistence	Message data	Put application name	Put date
1	Persistent	Test Mesesage 1 (default)	re MQ\java\jre\bin\javaw.exe	Jun 2
2	Not persistent	[header]□□* Input parameters for MQPut2 program *□□*□... s\Downloads\jh03\rfhutil.exe Jun 2		

© Copyright IBM Corporation 2008

Figure 17-7. Step three - Message persistence

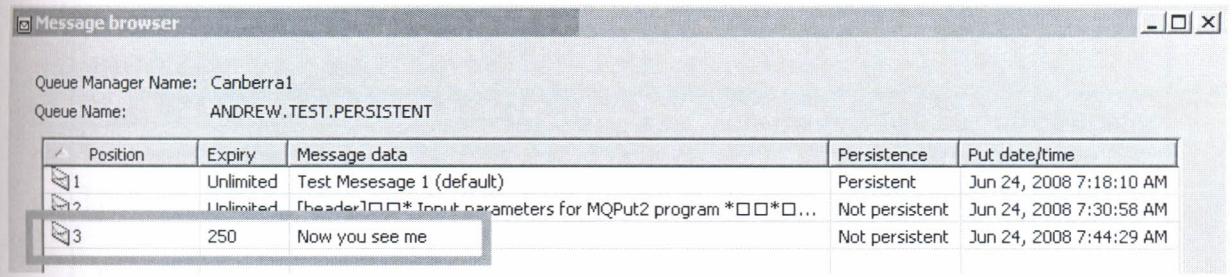
WM203 / VM2032.0

Notes:

Queues having DEFPSIST(YES) does not mean every message on that queue is persistent. The putting application can override this property in the MQMD.

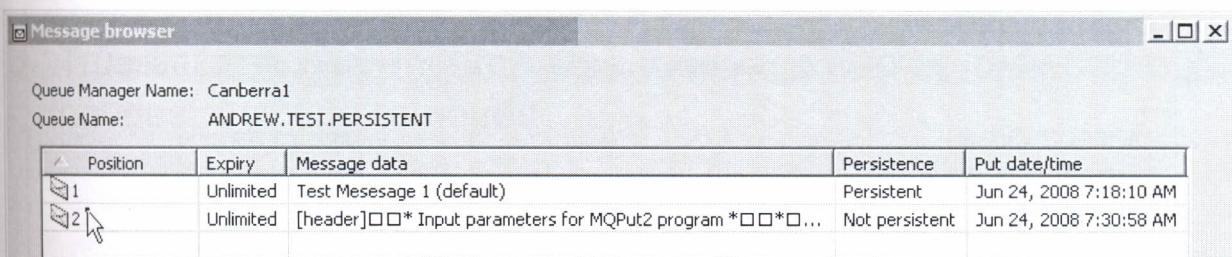
Use the message browse facility of WebSphere MQ Explorer to examine the persistence of individual messages on a queue.

Step four - Message expired



Queue Manager Name: Canberra1
Queue Name: ANDREW.TEST.PERSISTENT

Position	Expiry	Message data	Persistence	Put date/time
1	Unlimited	Test Mesesage 1 (default)	Persistent	Jun 24, 2008 7:18:10 AM
2	Unlimited	[header]□□* Input parameters for MQPut2 program *□□*□...	Not persistent	Jun 24, 2008 7:30:58 AM
3	250	Now you see me	Not persistent	Jun 24, 2008 7:44:29 AM



Queue Manager Name: Canberra1
Queue Name: ANDREW.TEST.PERSISTENT

Position	Expiry	Message data	Persistence	Put date/time
1	Unlimited	Test Mesesage 1 (default)	Persistent	Jun 24, 2008 7:18:10 AM
2	Unlimited	[header]□□* Input parameters for MQPut2 program *□□*□...	Not persistent	Jun 24, 2008 7:30:58 AM

© Copyright IBM Corporation 2008

Figure 17-8. Step four - Message expired

WM203 / VM2032.0

Notes:

Messages can expire, but not by default. Applications can explicitly set a life span of a message. This is typically the case with time sensitive data like quotations. The expiry time shown here is in 1/10ths of second and represents the amount of time remaining (time to live).

Step five - Uncommitted messages

```
display qstatus(testq1) type(queue) all
AMQ8450: Display queue status details.
  QUEUE (TESTQ1)
  CURDEPTH (4)                                     TYPE (QUEUE)
  OPPROCS (1)                                     IPPROCS (0)
  UNCOM (YES)                                     QTIME ( , )

QUEUE (TESTQ1)                                     TYPE (HANDLE)
APPLTAG (c:\WMQ\bin\amqsput.exe)                  APPLTYPE (USER)
:
OUTPUT (YES)                                       PID (10880)
QMURID (0.0)                                       SET (NO)
TID (1)
URID (XA_FORMATID[ ] XA_GTRID[ ] XA_BQUAL[ ])
URTYPE (QMGR)
USERID (andrew@localhost)
```

© Copyright IBM Corporation 2008

Figure 17-9. Step five - Uncommitted messages

WM203 / VM2032.0

Notes:

Uncommitted messages are not available to other applications until the unit of work is complete.

A display of the current depth of a queue includes the uncommitted messages, even though they are not available.

Use: `display qstatus(TESTQ1) type(handle) all`

To discover information about the processes attached to this queue.

Checkpoint questions

1. Why might a message might be delayed on a transmission queue?
 - a. Queue Manager is at MAXCHL
 - b. Transmission queue is full
 - c. Transmission queue is GET(DISABLED)
 - d. Channel initiator is not running
2. True or false: Persistent messages do not expire.
3. True or false: An uncommitted message can be browsed but not destructively removed.

© Copyright IBM Corporation 2008

Figure 17-10. Checkpoint questions

WM203 / VM2032.0

Notes:

Lost messages topic summary

Having completed this topic, you should be able to:

- List places where messages can be held up in transit to their intended destination
- Describe situations where messages are not or no longer available in normal processing

© Copyright IBM Corporation 2008

Figure 17-11. Lost messages topic summary

WM203 / VM2032.0

Notes:

17.2.Activity tracing

Activity tracing topic objectives

After completing this topic, you should be able to:

- Describe the use and purpose of activity messages
- Describe the use and purpose of the display trace route utility

© Copyright IBM Corporation 2008

Figure 17-12. Activity tracing topic objectives

WM203 / VM2032.0

Notes:

Activity reports

- Can be used to determine the route of message
- Some terms:
 - Activities: work performed on a message consisting of one or more operations
 - Operation: single pieces of work such as an MQGET
- Specified using MQRO_ACTIVITY bit in MQMD
- Applications may choose to generate an activity report when it sees the flag
 - One “application” that chooses this is the MCA
- Activity reports take the form of PCF data

© Copyright IBM Corporation 2008

Figure 17-13. Activity reports

WM203 / VM2032.0

Notes:

Activity reports are used to identify the route a message has taken through a queue manager network. As a message is routed through a queue manager network, various applications perform *activities* on behalf of the message. These activities are used to determine the message route.

Activities are discrete actions performed on behalf of a message by an application, and consist of operations, which are single pieces of work that are performed by an application.

The following are examples of activities:

- A message channel agent (MCA) sending a message from a transmission queue down a channel.
- An MCA receiving a message from a channel and putting it on its target queue.
- An application getting a message from a queue, and putting a reply message in response.
- The WebSphere MQ publish/subscribe engine processing a message.

Activities consist of one or more *operations*. Operations are single pieces of work that are performed by an application. For example, the activity of an MCA sending a message from a transmission queue down a channel, consists of the following operations:

- Getting a message from a transmission queue (a *Get* operation).
- Sending the message down a channel (a *Send* operation).

In a publish/subscribe network, the activity of a message being processed by the WebSphere MQ publish/subscribe engine can consist of the following multiple operations:

- Putting a message to a topic string (a *Put* operation).
- Zero or more operations for each of the subscribers that are considered for receipt of the message (a *Publish* operation, a *Discarded Publish* operation, or an *Excluded Publish* operation).

Using activity reports

- An application creates a message with the activity report option set and a reply queue specified
- When the message is processed by an MCA an activity report is generated
 - Even on its own the MCA can generate useful information
 - Customer applications can be enhanced to generate extra activity reports if there is something worthwhile to inform the originating application.
- Provided there is a route to the reply queue all the reports can be collated by the originating application
- By piecing together all the reports the route the message took can be determined, along with some measures of the performance of the route

© Copyright IBM Corporation 2008

Figure 17-14. Using activity reports

WM203 / VM2032.0

Notes:

Trace route utility

- Dedicated messages called trace-route messages may be put into the network
- A trace-route message is for special format
- Behavior of trace-route message is controlled by data in the message itself
- Depending on message and qmgr settings, as activities are performed activity data is either reported or accumulated
- Trace route message is discarded just before it reaches its final destination queue.

© Copyright IBM Corporation 2008

Figure 17-15. Trace route utility

WM203 / VM2032.0

Notes:

Dedicated messages called trace-route messages may be put into the network.

- Key difference is that the body of the message can be modified
- Allows reliable ordering of activities, optional delivery of the message, loop detection/prevention and the specifying of the accumulation that should take place

A trace-route message is one where:

- MQMD Format is either MQFMT_ADMIN or MQFMT_EMBEDDED_PCF
- MQCFH Type, Version and Command fields are MQCFT_TRACE_ROUTE, MQCFH_VERSION_3 and MQCMD_TRACE_ROUTE

Behavior of trace-route message is controlled by data in the message itself.

- Some attributes change as message flows through the queue manager network.

Depending on message and queue manager settings, as activities are performed:

- An activity report can be generated

- Activity information can be recorded (accumulated) in the message itself
- A reply message can be generated containing all accumulated activities when the trace-route message reaches the end of its route

The trace route message is not a real application message so it is discarded before it reaches its final destination so no application sees it.

DSPMQRTE

- The **dspmqrte** command is supplied with WebSphere MQ
- Generates trace route messages
- The **dspmqrte** command will gather responses, interpret and display results
- User added properties will also be displayed
- Can be used with WebSphere MQ on z/OS

© Copyright IBM Corporation 2008

Figure 17-16. DSPMQRTE

WM203 / VM2032.0

Notes:

The **dspmqrte** command is supplied with distributed WebSphere MQ. It generates Trace Route messages.

The **dspmqrte** command gathers responses and:

- Orders them correctly
- Identifies whether information is missing
- Displays known parameters in an easy to read fashion.

User added properties are also displayed, although not as intelligently as recognized properties.

Although not supplied with WebSphere MQ for z/OS, **dspmqrte** can be client connected to any queue manager and so can be used with z/OS by any customer with the Client Attach Facility (CAF).

Dspmqrte sample output (1 of 2)

```
C:\Program Files\WebSphere MQ\bin>dspmqrte -q fred -m AJGMQ1 -v outline
AMQ8653: DSPMQRTE command started with options '-q fred -m AJGMQ1 -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'AJGMQ2', queue m
nager 'AJGMQ1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
ApplName: 'ebSphere MQ\bin\dspmqrte.exe'

Operation:
OperationType: Put
QMgrName: 'AJGMQ1'
QName: 'fred'
ResolvedQName: 'AJGMQ2'
RemoteQName: 'fred'
RemoteQMgrName: 'AJGMQ2'
-----
Activity:
ApplName: 'ebSphere MQ\bin\runmqch1.exe'

Operation:
OperationType: Get
QMgrName: 'AJGMQ1'
QName: 'AJGMQ2'
ResolvedQName: 'AJGMQ2'

Operation:
OperationType: Send
QMgrName: 'AJGMQ1'
RemoteQMgrName: 'AJGMQ2'
ChannelName: 'AJGMQ1.TO.AJGMQ2'
ChannelType: Sender
XmitQName: 'AJGMQ2'
```

© Copyright IBM Corporation 2008

Figure 17-17. Dspmqrte sample output (1 of 2)

WM203 / VM2032.0

Notes:

The first activity shows dspmqrte placing the trace route on remote queue "fred" on queue manager AJGMQ1.

The second activity shows the sender channel process **runmqch1** taking the message off the transmission queue and sending the message to queue manager AJGMQ2 along channel named AJGMQ1.TO.AJGMQ2

Dspmqrte sample output (2 of 2)

```
Activity:  
ApplName: 'ebSphere MQ\bin\amqrmpa.exe'
```

```
Operation:  
OperationType: Receive  
QMgrName: 'AJGMQ2  
RemoteQMgrName: 'AJGMQ1  
ChannelName: 'AJGMQ1.TO.AJGMQ2  
ChannelType: Receiver
```

```
Operation:  
OperationType: Discard  
QMgrName: 'AJGMQ2  
Feedback: UnknownObjectName
```

```
AMQ8652: DSPMQRTE command has finished.
```

© Copyright IBM Corporation 2008

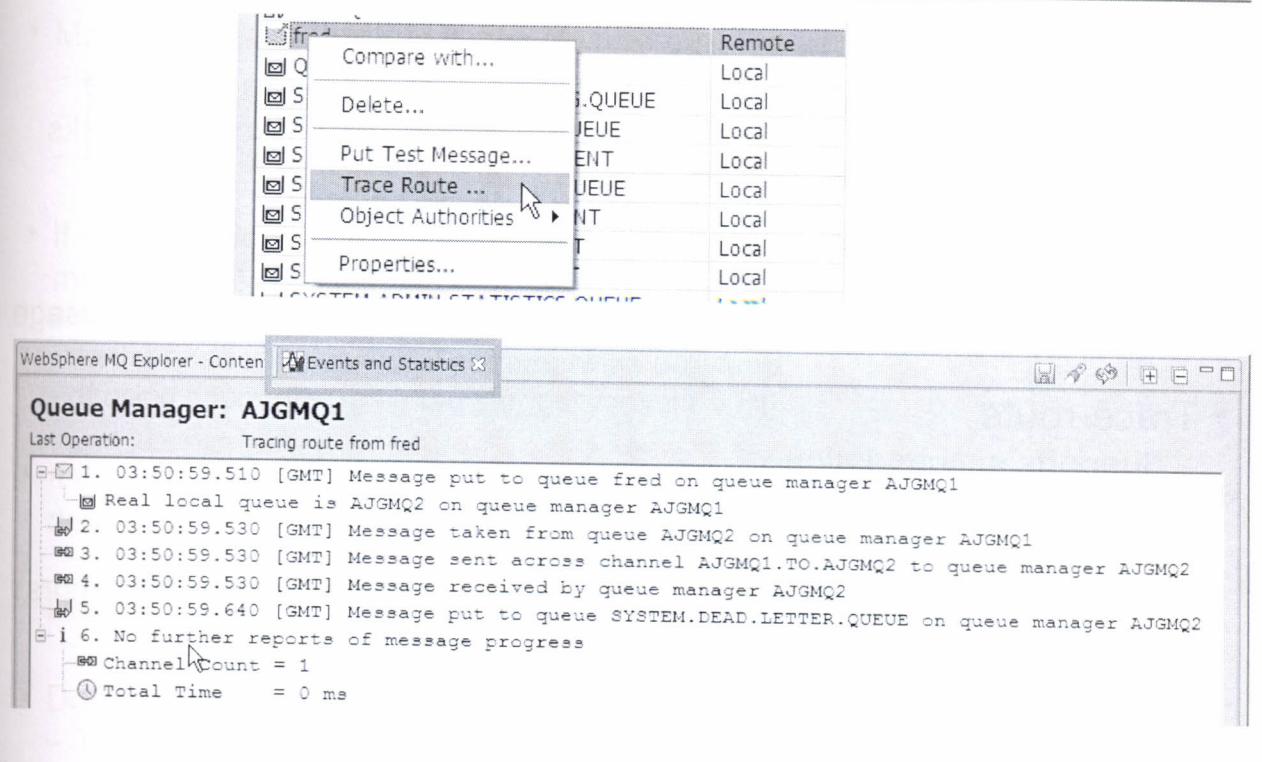
Figure 17-18. Dspmqrte sample output (2 of 2)

WM203 / VM2032.0

Notes:

The third activity shows the channel receiver process receiving the message off the channel and attempting to deliver the message. The second operation shows Discard, which is not unexpected, however the feedback code of UnknownObjectName shows that the target queue was not found.

Example using MSOP



Notes:

MSOP is the statistics and events plug-in. Note the results for MSOP are shown in a separate tab on the Eclipse window.

Comparing activity recording and trace-route

- Both
 - Could be used to determine the last known location of a message
 - Can be used to check the configuration of queue manager networks
- Activity
 - Can be used without adding to, or modifying message data
 - Activity messages are processed with little increased processor usage
- Trace-route
 - Supports accumulation
 - Report ordering can be easily determined
 - Does not require return paths
 - Knows if data is missing (such as from passing through earlier qmgrs)
 - Some increased processor usage for parsing and modifying PCF group

© Copyright IBM Corporation 2008

Figure 17-20. Comparing activity recording and trace-route

WM203 / VM2032.0

Notes:

Hints and tips for trace route

- Missing information does not always mean a problem has occurred.
 - It may be that a qmgr attribute on one computer is set to QUEUE, rather than MSG
 - Path to reply queue may not be defined or available
- If user cannot put to reply queue activity, information seems to be missing.
 - Disabling PASS_DISCARD_AND_EXPIRY should lead to responses being sent to the dead letter queue if not otherwise delivered
 - If using dspmqrte, raise the verbosity level (-v all)
- If the last operation seen is a put to a xmit queue then a channel may be down
- Default dspqmrt e options should show the last operation as DISCARD
- Do not put trace-route messages to distribution lists
 - This combination is not supported, although activity reports do work with dist lists

© Copyright IBM Corporation 2008

Figure 17-21. Hints and tips for trace route

WM203 / VM2032.0

Notes:

Checkpoint questions

1. True or false: An activity is a single action like a MQPUT.
2. True or false: An activity report data requires custom code to interpret and format the raw data.
3. True or false: Activity reports must be handled correctly by applications.
4. True or false: Trace route messages must be handled correctly by applications.
5. True or false: Trace route can be run as a WebSphere MQ client
6. True or false: Trace route is most effective when tracing messages across channels

© Copyright IBM Corporation 2008

Figure 17-22. Checkpoint questions

WM203 / VM2032.0

Notes:

Activity tracing topic summary

Having completed this topic, you should be able to:

- Describe the use and purpose of activity messages
- Describe the use and purpose of the display trace route utility

© Copyright IBM Corporation 2008

Figure 17-23. Activity tracing topic summary

WM203 / VM2032.0

Notes:

Unit summary

Having completed this unit, you should be able to:

- Describe how WebSphere MQ keeps track of a message that has been tagged for tracing
- Explain to programmers how to initiate message tracking

© Copyright IBM Corporation 2008

Figure 17-24. Unit summary

WM203 / VM2032.0

Notes: