

# Unit 10.Dead letter queues

## What this unit is about

This unit describes dead letter queues.

## What you should be able to do

After completing this unit, you should be able to:

- Explain dead letter queues
- Describe the dead letter queue handler
- Describe strategies for using dead letter queues

## How you will check your progress

- Checkpoint questions
- Machine exercises

## References

SC34-6941	WebSphere MQ Script (MQSC) Command Reference
SC34-6931	WebSphere MQ Intercommunication
SC34-6933	WebSphere MQ Queue Manager Clusters

## Unit objectives

---

After completing this unit, you should be able to:

- Explain dead-letter queues
- Describe the dead-letter queue handler
- Describe strategies for using dead-letter queues

© Copyright IBM Corporation 2008

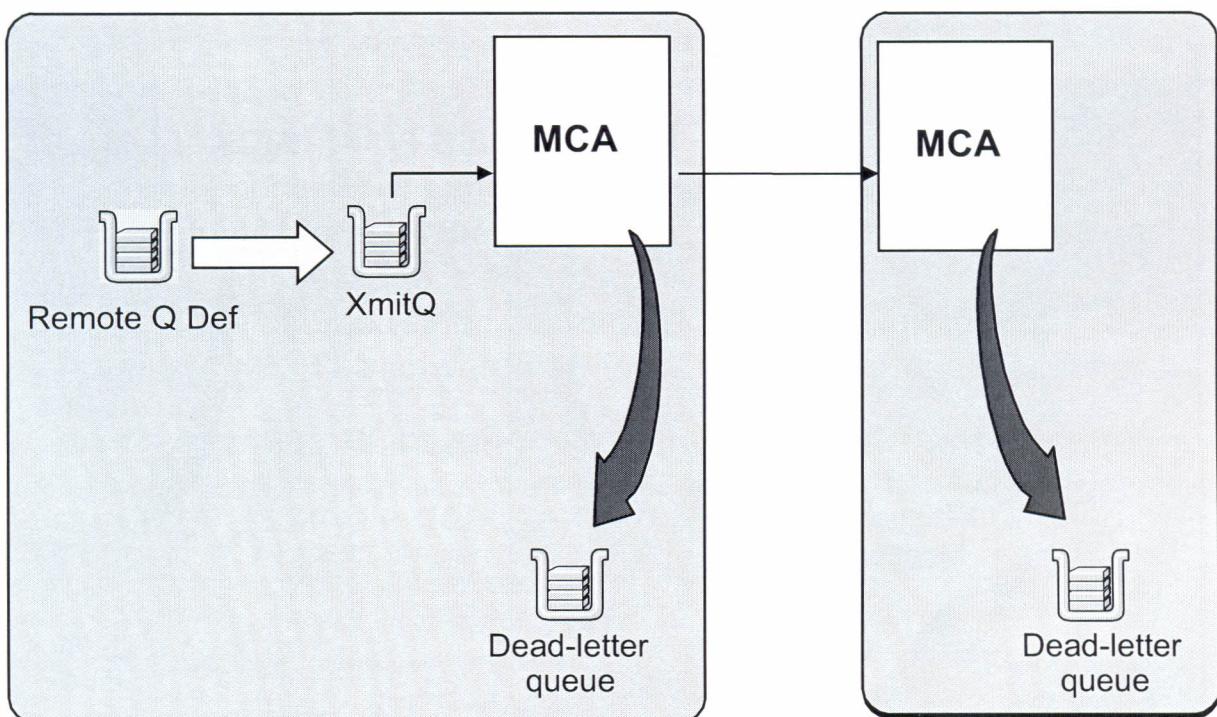
---

Figure 10-1. Unit objectives

WM203 / VM2032.0

### **Notes:**

## Dead-letter queue (1 of 2)



© Copyright IBM Corporation 2008

Figure 10-2. Dead-letter queue (1 of 2)

WM203 / VM2032.0

### Notes:

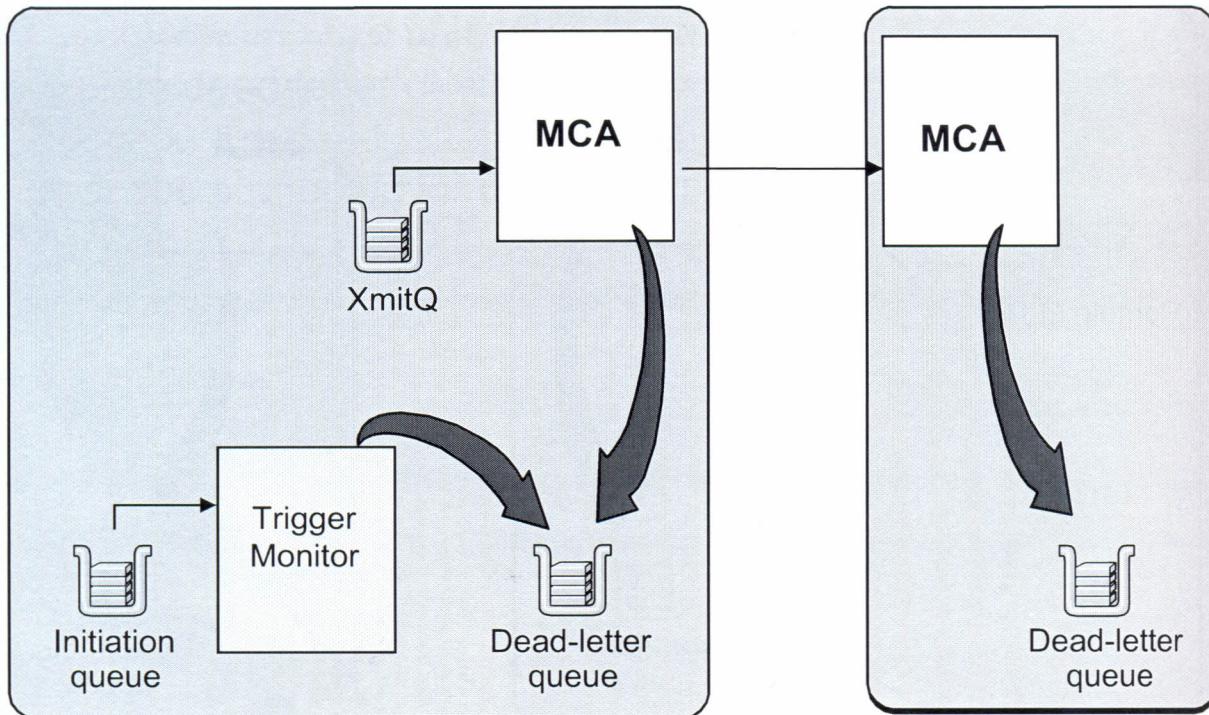
When a problem relating to a message is detected asynchronously, an exception report is generated if one has been requested and the report is sent to the specified reply-to queue. The *Feedback* field in the message descriptor of the report message indicates the reason for the report. The original message is put on the dead letter queue unless `MQRO_DISCARD_MSG` is requested as a report option.

If a message cannot be delivered, it is put on the dead letter queue at the receiving end of a message channel, if one is defined. Message-retry at the receiving end of a channel may be useful if the problem is only temporary.

If `CONVERT (YES)` is specified in the channel definition at the sending end of a message channel and a message cannot be converted, the message is put on the dead letter queue at the sending end.

If a message cannot be put on the dead letter queue, the channel is stopped and the message remains on the transmission queue. A fast nonpersistent message, however, is discarded in these circumstances and the channel remains open.

## Dead-letter queue (2 of 2)



© Copyright IBM Corporation 2008

Figure 10-3. Dead-letter queue (2 of 2)

WM203 / VM2032.0

### Notes:

If a trigger message is created but cannot be put on the initiation queue (for example, because the queue is full or the length of the trigger message is greater than the maximum message length specified for the initiation queue), the trigger message is put instead on the dead-letter (undelivered message) queue.

If the put operation to the dead-letter queue cannot complete successfully, the trigger message is discarded and a warning message is placed in the error log.

## Dead-letter queue handler

`runmqdlq < Rules_Table`

- Rules table contains a set of rules
  - Each rule consists of pattern matching keywords and an action
  - For each message on the dead-letter queue, each rule whose pattern matches the message is attempted in turn
  - A message can be retried, forwarded, discarded, or ignored
  - A message can be forwarded with, or without, the dead letter header
- Can have multiple instances, each with a different rules table
- Source of a sample dead-letter queue handler is supplied

```
DESTQM(QM17) ACTION(FWD) FWDQ(&DESTQ) FWDQM(QM17A)
MSGTYPE(MQMT_REPORT) FEEDBACK(MQFB_EXPIRATION) +
ACTION(DISCARD)
REASON(MQRC_Q_FULL) ACTION(RETRY)
DESTQ(XYZ*) ACTION(FWD) FWDQ(XYZ_DEADQ) FWDQM('')
```

© Copyright IBM Corporation 2008

Figure 10-4. Dead-letter queue handler

WM203 / VM2032.0

### Notes:

The purpose of a dead letter queue handler is to provide an automated way of handling messages on the dead letter queue.

## Using dead-letter queues

- Create a dead-letter queue on all queue managers
  - Use message-retry on message channels for transient conditions
  - Consider "return to sender"

```
MQRO_PASS_MSG_ID +
MQRO_PASS_CORREL_ID +
MQRO_EXCEPTION_WITH_FULL_DATA +
MQRO_DISCARD_MSG
```

- Use a dead-letter queue handler
  - Trigger when message arrives on the dead-letter queue
  - Possibly attempt further retries
  - If unsuccessful, forward to application dead-letter queue associated with:
    - Destination queue
    - Application specified by the *PutAppName* field in the message descriptor
- Do not allow an application dead-letter queue to become full

```
ACTION(FWD) FWDQ(REALLY.DEAD.QUEUE) HEADER(YES)
```

© Copyright IBM Corporation 2008

Figure 10-5. Using dead-letter queues

WM203 / VM2032.0

### Notes:

Remember that creating means to define the queue and to tell the queue manager the name of its dead letter queue.

## Checkpoint questions

1. What will happen if a dead-letter queue does not exist and a message cannot be delivered? What is the one exception to this rule?
2. What is the purpose of a dead-letter queue handler?
3. From a dead-letter queue handler rules table:

DESTQM(A1) action(fwd) fwdq(Q1) fwdqm(A2)

- a. Where will a message destined for queue manager A1 be found?
- b. How can the rule be modified so that the message will be stripped of its dead letter header?

© Copyright IBM Corporation 2008

Figure 10-6. Checkpoint questions

WM203 / VM2032.0

### Notes: