

## WebSphere MQ : Gestion de la Dead Letter Queue

---

*Ce document traite du fonctionnement de la Dead Letter Queue, des moyens pour gérer celle-ci et du recyclage des messages.*

### **Envoi d'un message par une application**

Lorsqu'une application utilisateur pose un message dans une queue, le Queue Manager vérifie la conformité du message, et en particulier sa « délivrabilité », c'est à dire la possibilité de le livrer dans la queue destinataire.

Si la queue destinataire est locale, le Queue Manager vérifie simplement que cette queue existe et que celle-ci peut accepter des messages (ni pleine, ni « *Put Disabled* »).

Si la queue destinataire est distante (remote), le Queue Manager local vérifie qu'il est capable d'acheminer ce message vers le Queue Manager distant gérant la queue destinataire.

Si la queue locale ne peut pas accepter le message, ou si aucun chemin n'est trouvé le Queue Manager distant, le Queue Manager local « refuse » le dépôt du message par l'application et lui renvoie un code retour (MajCod = 2 – Fail, et ReasonCode = explication du problème rencontré).

### **Livraison d'un message vers un site éloigné**

Le Queue Manager A reçoit, de la part d'une application, un message à destination d'une queue gérée par le Queue Manager B. Le Queue Manager A connaît un chemin vers le Queue Manager B, et la *Transmission Queue* correspondante est capable de recevoir des messages à transmettre.

Le Queue Manager A accepte donc le message et renvoie à l'application un code retour OK (MajCod = 0 ReasonCode = 0000), il est donc « responsable » du message.

Dans l'immense majorité des cas, le Queue Manager A transmettra sans problème le message au Queue Manager B, qui le livrera dans la queue destinataire.

### **La Dead Letter Queue**

Dans quelques rares cas, le message ne pourra pas être livré et sera dirigé vers la « Dead Letter Queue » (DLQ) :

Lorsque le message arrive sur le Queue Manager B, ce dernier tente de poser le message dans la queue de destination. Si cette queue est pleine, ou si elle est « fermée » (*put disabled*), il n'est pas possible d'y mettre le message. Mais il n'est également plus possible d'avertir l'application que son message est refusé, puisque le Queue Manager A a accepté ce message avec un Code Retour OK.

## WebSphere MQ : Gestion de la Dead Letter Queue

---

C'est dans un cas comme celui-ci que le Queue Manager B va déposer le message dans sa DLQ.

Si aucune DLQ n'est définie pour ce Queue Manager, le message n'est pas perdu pour autant, dans ce cas il reste dans la *Transmission Queue* du Queue Manager A et le channel correspondant se ferme.

L'arrivée d'un message applicatif dans une DLQ est donc un événement important, qui doit le plus rapidement possible être rapporté à un processus de surveillance.

Ce rapport est en général effectué via un « *DLQ Monitor* », programme qui est en écoute sur la DLQ, et qui déclenche un traitement lors de l'arrivée d'un message. Il existe également des mécanismes système qui préviennent l'opérateur (message dans QSYSOPR, observateur d'événements Windows, ...).

### **Arrivée d'un message en Dead Letter Queue**

Lorsqu'un Queue Manager ne peut pas délivrer un message dans la queue de destination demandée, il va écrire ce message dans la DLQ. Un « *Dead Letter Header* » (MQDLH) est ajouté au début du message, qui indique le nom de la queue où aurait du être posé le message, la raison de l'échec ainsi que l'heure et la date de ce dernier.

L'écriture d'un message en DLQ peut également être effectuée par une application, si elle vient de lire un message qu'elle n'est pas capable de traiter. Dans ce cas, l'application devra prévoir de rajouter au message le *Dead Letter Header*.

Dans les deux cas, ce header pourra être exploité par le *DLQ Monitor*, pour informer le processus de surveillance de la cause de non livraison (queue pleine, fermée, endommagée, introuvable).

### **Recyclage des messages en Dead Letter Queue**

Sauf exception, les messages arrivés en DLQ sont destinés à être « recyclés » vers la queue où ils auraient du normalement être livrés, une fois corrigé le problème qui a entraîné l'échec de livraison.

Ce dernier point est très important : si le Queue Manager n'a pas pu poser un message dans une queue applicative, il y a peu de chances qu'un ré-essai automatique (immédiat ou différé) soit plus efficace car la cause du rejet initial (queue pleine, fermée, ...) sera probablement encore là.

L'arrivée d'un message en DLQ doit donc déclencher les étapes suivantes :

- Signalisation de l'événement (*DLQ Monitor* par exemple)

## WebSphere MQ : Gestion de la Dead Letter Queue

---

- Analyse de la raison du rejet (queue pleine, fermée, ...) et détermination de la cause (volumétrie inhabituelle, sous-dimensionnement d'une queue, application d'envoi qui boucle, application de réception arrêtée ou pas assez rapide, ...)
- Correction de la cause (re-dimensionnement de la queue, relance de l'application, changement des paramètres d'exécution de l'application de réception, ...)
- Recyclage des messages.

### **Le Dead Letter Queue Handler**

Un *Dead Letter Queue Handler* (DLQ Handler) est un programme qui gère la DLQ, au sens où il va tenter de recycler les messages qui y ont été déposés.

Il est tout à fait possible de développer soit même un *DLQ Handler*, c'était d'ailleurs la seule option il y a quelques années.

Depuis la version 5.1 (4.21 sur l'AS/400), IBM a commencé à livrer un *DLQ Handler* sur les différentes plates-formes WebSphere MQ.

Il s'agit du *runmqdlq* sur Unix & Windows, ou de *STRMQMDLQ* sur AS/400. Son fonctionnement est détaillé dans la brochure « System Administration Guide » de WebSphere MQ.

Ce programme utilitaire qui permet l'automatisation de ce recyclage. Les actions à effectuer sur les messages sont définies dans une table d'action, où l'on peut préciser très finement l'action à réaliser. Pour chaque critère, il est possible de préciser l'action souhaitée (*retry*, *forward*, *discard*, *ignore*).

### **Risques de l'automatisation du DLQ Handler**

Il peut être tentant d'automatiser le processus de recyclage des messages en DLQ, laissant tourner en permanence le *DLQ Handler*, afin qu'il recycle au fur et à mesure les messages. Ce n'est pas forcément une bonne idée, car si un message tombe en DLQ à l'instant « t », c'est qu'il y a une raison pour cela. Et il est assez peu probable qu'à l'instant « t + 0.1 seconde » cette raison ait disparu !

De plus, si dans une suite de message certains passent par la DLQ avant d'être recyclés, il est fort probable qu'ils arriveront à destination en ordre dispersé, au moins par rapport à ceux qui sont passés directement.

### **Conclusion**

L'arrivée d'un message en DLQ veut dire qu'un Queue Manager ne **sait pas** ou ne **peut pas** traiter ce message. Cela signifie en général qu'une action **manuelle** est nécessaire pour corriger la situation.

## WebSphere MQ : Gestion de la Dead Letter Queue

---

Il faut donc être très prudent avant de mettre en place des mécanismes automatisant le traitement et la reprise des messages arrivant en DLQ.

Fin du document