Support de présentation

SOMMAIRE

1 - PREAMBULE

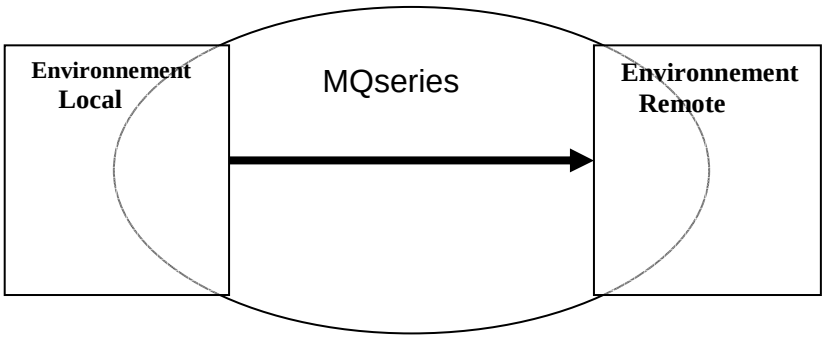
Ce document a été élaboré à l'aide de Michel M. -expert MQ Series pour le cours Client Serveur du CNAM.
Des erreurs ont pu se glisser dans le texte. N'hésitez pas à me communiquer vos remarques ou vos corrections.
Bonne lecture.

2 - QU'EST-CE QUE MQSERIES ?

2.1 - Définition

MQseries est un Middleware orienté message (M.O.M) développé par IBM permettant à 2 applications hébergées sur des plates-formes hétérogènes de communiquer.

Le but principal est de fournir un moyen simple de communiquer entre 2 applications
Une API fournie pour de nombreux langages permet de déposer et de lire des messages .
MQseries se charge du transport entre les 2 plates-formes au fil de l'eau.



- ⇨ **dépôt d'information (sans attente de réponse)**
- ⇨ **conversation (question avec attente de réponse)**

2.2 - Exemples d'utilisation

- **Application Mainframe générant des fax → serveur NT d'émission de fax**
- **Application Mainframe <→ serveur NT pour EDI vers extérieur**
- **Application Mainframe → mise à jour de base de données d'un serveur X**
- **Serveur UNIX avec application HR → application Mainframe pour fichier du personnel**
- **WEBSPHERE MQ Integrator est un logiciel IBM d'EAI pour mettre à jour différentes plates-formes applicatives**

2.3 - Plates-formes et versions

Toutes les plates-formes sont supportées dont :

- **Mainframe IBM : VSE (2.1) ; OS/390 (5.2)**
- **UNIX AIX (5.2), HP-UX(5.2), SOLARIS(5.2), LINUX (5.2) …**
- **PC WINDOWS NT (5.2) ; WINDOWS (2.1) ; OS/2 (5.1)**
- **Divers OS/400 (5.2), Open VMS (5.2)**

2.4 - Autres M.O.M du marché

ORACLE AQ , MICROSOFT MSMQ, BEA MQ … mais pas présents sur toutes les OS

3 - OBJETS / RESSOURCES

3.1 - QUEUE MANAGER

Gère les queues (il ne doit pas perdre de message) et les autres ressources grâce à des agents qui traitent chaque type de ressource.

Un message MQ peut faire jusqu'à 4 Mb et il est composé de :

message descriptor (324 caractères) renseigné par l'application et le Queue Manager)

data de l'application (de longueur fixe ou variable)

3.2 - QUEUES

Queue Remote : est une queue appartenant à un autre Queue Manager, dans laquelle on souhaite déposer un message

Queue Locale de transmission : est une queue où transitent les messages de toutes les queues remotes d'un Queue Manager avant d'être transmis à celui-ci.

Il faut donc une queue de transmission par Queue Manager qui doit recevoir des messages (une par Channel d'émission).

Queue Locale d'application : est une queue qui contient les messages envoyés par des Queue Manager connectés ou écrit par une application locale (ou client).

Quand un message est écrit dans une Queue locale d'application, MQseries peut déclencher un process par un system de TRIGGER.

En général, ce process lira les messages non traités.

DEAD LETTER QUEUE : est une queue locale utilisée par le QM pour déposer les messages qu'il ne peut pas écrire dans la Queue destinatrice (messages reçus d'un QM pour une Queue locale inexistante, stoppée, fichier de la Queue plein….)

LOG : est une Queue utilisée par le QM pour écrire des messages systèmes d'information (ex : QM démarré ..) et d'anomalie

ALIAS : permet de coder dans les applications un nom de Queue différent de celui défini dans le QM (l'administrateur de MQ doit définir le relation ALIAS – nom réel)

3.3 - CHANNEL

Lien entre les Queue Manager

- Channel d'émission (SENDER)
- Channel de réception (RECEIVER)

Le nom du Channel doit être identique (c'est le nom du Channel d'émission de l'un et le nom du Channel de réception du QM partenaire)

Le protocole réseau peut-être :

SNA (en LU 6.2 avec l' APPC)

TCP/IP (avec le port 1414 de préférence)

3.4 - MQI

Interface simple entre les applications et MQseries

Ensemble de sous-programmes appelés par 'CALL' dans de nombreux langages de programmation (COBOL, PL/1, C, C++, Visual Basic, java)

3.5 - Autres définitions

Qdepth est le nombre de messages non encore traités dans une queue locale (Qlocal normale) ou non encore transmis (Qlocal de transmission)

QSN (Queue Sequence Number) est le numéro de séquence attribué à un message lors de son écriture dans une Queue (note : le message et son MSN disparaissent en cas de Backout, le QSN passe à zéro lors de la recréation de la Queue).

MSN (Message Sequence Number) est le numéro de séquence du message transmis sur un Channel . La valeur est conservée par le channel émetteur et le channel récepteur associé . Elle est vérifiée à chaque envoi de message pour un contrôle de séquence par le récepteur .

Les Agents : certaine taches de MQseries sont effectuées par des agents de MQseries (exemple MCA : Message Channel Agent)

4 - MQ CLIENT

Possibilité de faire tourner des applications MQ sur certaines plates-formes sans y installer un Queue Manager et tous ses fichiers

MQclient est une licence gratuite

Le MQclient possede le MQI pour développer des applications (qui ressemblent à celles développées localement) et dialoguer avec le Queue Manager

Le MQclient communique avec un Queue Manager uniquement en TCP/IP

Le Queue Manager ne peut pas déclencher de process (trigger) sur un MQclient

La disponibilité de MQseries sur cette plate-forme dépend de la disponibilité

de son Queue Manager

réseau de la connexion au Queue Manager

5 - MQ INTERFACE

MQI

appels 'majeurs'

⇨ **MQCONN (QMGR,Hconn,CC,RC)** ⟶ Queue Manager

⟵

⇨ **MQOPEN (Hconn,ObjDesc,Options,Hobj,CC,RC)**

OUTPUT → pour faire du MQPUT
INPUT → MQGET détruit le message
BROWSE → MQGET ne détruit pas le message
(le QDEPTH n'est pas modifié)

⇨ **MQPUT (Hconn,Hobj,MSGDesc,PUTOpts,Length,Buffer,CC,RC)**

Message Descriptor	DATA

⇨ **MQGET (Hconn,Hobj,MSGDesc,GETOpts,Length,Buffer,DataLength,CC,RC)**

⇦ **MQCLOSE (Hconn,Hobj,Options,CC,RC)**

⇦ **MQDISC (Hconn,CC,RC)**

⇨ **MQPUT1 (Hconn,Hobj,MSGDesc,PUTOpts,Length,Buffer,CC,RC)**
MQPUT1 équivalent à **MQOPEN + MQPUT + MQCLOSE**

Note : CC = Condition Code
RC= Return Code

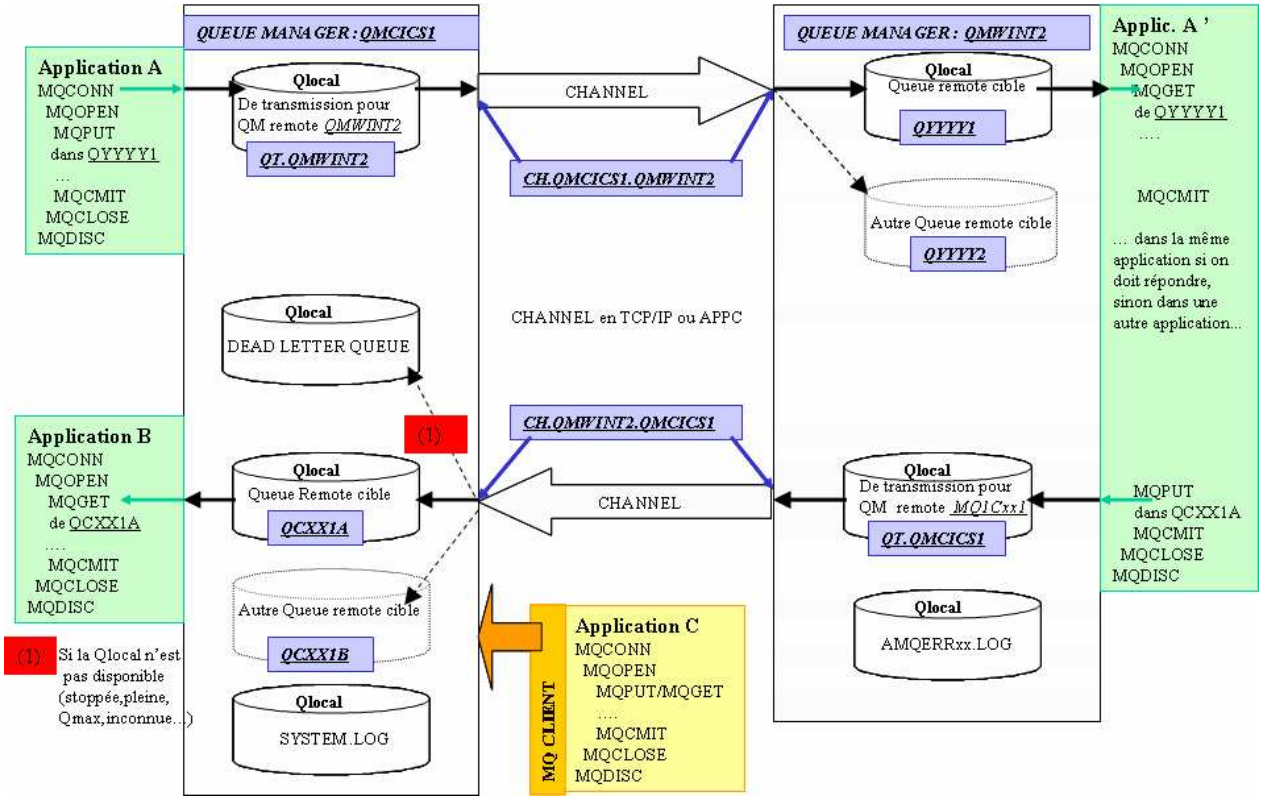
appels 'mineurs'

⇨ **MQINQ** pour interroger des objets
⇨ **MQSET** pour modifier des objets

appels 'mineurs' pour gérer l'intégrité

⇨ **MQBEGIN** pour début de tache
⇨ **MQCMIT** pour valider les actions (écriture, maj, suppression)
⇨ **MQBACK** pour faire marche arrière sur les actions

6 - ANNEXE


QUEUE MANAGER : QMCICS1
QUEUE MANAGER : QMWINT2
Application A
MQCONN
MQOPEN
MQPUT
dans QYYYY1
...
MQCMIT
MQCLOSE
MQDISC
Qlocal
De transmission pour
QM remote QMWINT2
QT.QMWINT2
CHANNEL
CH.QMCICS1.QMWINT2
Qlocal
Queue remote cible
QYYYY1
Autre Queue remote cible
QYYYY2
Applic. A '
MQCONN
MQOPEN
MQGET
de QYYYY1
....
MQCMIT
... dans la même application si on doit répondre, sinon dans une autre application...
Qlocal
DEAD LETTER QUEUE
CHANNEL en TCP/IP ou APPC
(1)
CH.QMWINT2.QMCICS1
Application B
MQCONN
MQOPEN
MQGET
de QCXX1A
....
MQCMIT
MQCLOSE
MQDISC
Qlocal
Queue Remote cible
QCXX1A
CHANNEL
Qlocal
De transmission pour
QM remote MQ1Cxx1
QT.QMCICS1
MQPUT
dans QCXX1A
MQCMIT
MQCLOSE
MQDISC
Autre Queue remote cible
QCXX1B
Qlocal
AMQERRxx.LOG
(1) Si la Qlocal n'est pas disponible (stoppée,pleine, Qmax,inconnue...)
Qlocal
SYSTEM.LOG
MQ CLIENT
Application C
MQCONN
MQOPEN
MQPUT/MQGET
....
MQCMIT
MQCLOSE
MQDISC

7 - SUIVI DES EVOLUTIONS DU DOCUMENT

Date	Version	Objet de la modification
16/12/2001	001	Version Initiale
20/12/2001	002	Version Finale

Tableau 1 : Tableau de suivi des évolutions