

Sybase 12.5.3a et MQ Series

Sybase ASE 12.5.3a sera prêt à dialoguer directement avec Websphere MQ Series.

Septembre 2005

Documentation technique # 27



Introduction

- Sybase Adaptive Server Enterprise 12.5.3a sera livré Q3 2005 avec un support complet de Websphere MQ Series.
- Cette présentation est un bref aperçu des nouvelles fonctionnalités de Sybase 12.5.3a permettant de communiquer directement avec IBM MQ Series à travers l'option « Real Time Messaging ».

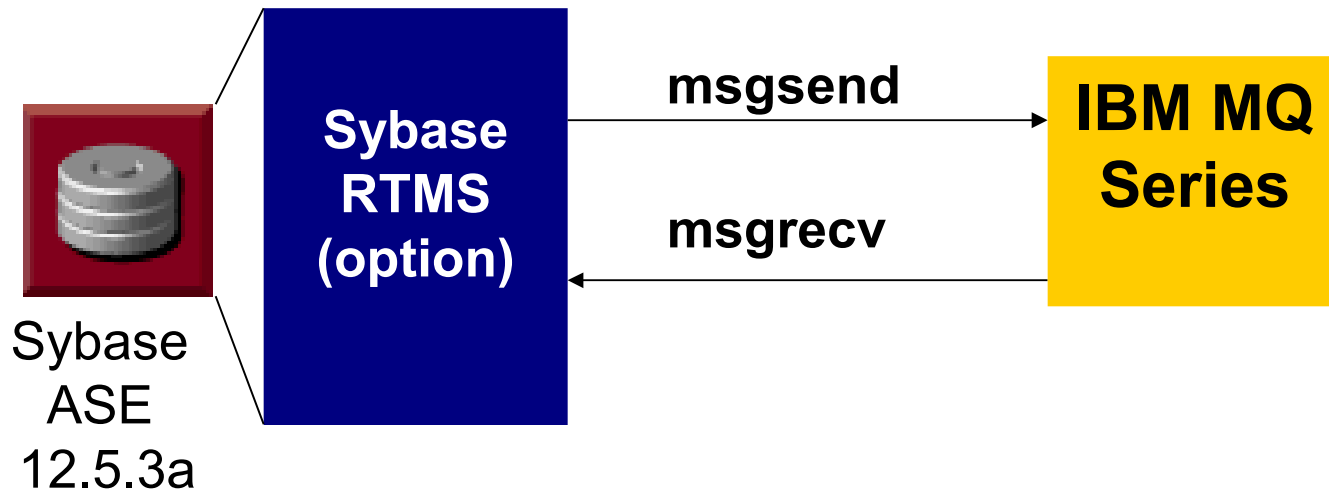
Historique : ASE 12.5.2 et RTMS

- L'option RTMS (Real Time Messaging Services) a été introduite avec Sybase Adaptive Server Enterprise 12.5.2.
- Avec ASE 12.5.2, seul TIBCO JMS (Java Messaging Services) est supporté.
- Cette option requiert une licence.

Historique : ASE 12.5.3a et MQ Series

- L'option RTMS (Real Time Messaging Services) avec Sybase 12.5.3a a été améliorée pour supporter les systèmes de messages TIBCO JMS, TIBCO RDV daemon et MQ Series.
- Sybase 12.5.3a sera capable d'envoyer et de recevoir des messages vers/provenant de MQ Series (même au sein d'une transaction).

Architecture : ASE 12.5.3a et MQ Series



Configuration ASE 12.5.3a

- De la mémoire doit être configurée au sein d'Adaptive Server Enterprise pour les services de messages (messaging services).
- Un ou plusieurs engines (online engines) doivent être dédiés aux communications entre ASE et MQ Series.
- Le rôle messaging_role doit être attribué aux logins autorisés à utiliser les services de messages.

Configuration MQ

- Le client MQ doit être installé sur la machine hébergeant Adaptive Server.
- Adaptive Server charge dynamiquement les bibliothèques partagées de MQ si la variable `LD_LIBRARY_PATH` référence les chemins vers le client MQ.
- Les autorisations MQ nécessaires doivent être données au user Unix qui démarre le serveur Sybase ASE.

Commandes T-SQL pour MQ : msgsend

- La commande Transact-SQL msgsend est utilisée pour envoyer les messages d'un serveur Sybase ASE vers une queue MQ :

```
select msgsend('hello world',  
              'ibm_mq:chnl1/tcp/host1(1234)?qmgr=QM,  
              queue=DEFAULT.QUEUE' message header 'priority=2')
```

- Beaucoup d'options sont offertes dans la commande msgsend.

Commandes T-SQL pour MQ : msgrecv

- La commande Transact-SQL msgrecv est utilisée pour lire les messages provenant d'une queue MQ :

```
select msgrecv(  
  'ibm_mq:chnl1/tcp/host1(1234)?qmgr=QM,  
  queue=DEFAULT.QUEUE'  
  option 'timeout=30s')
```

Commandes T-SQL pour MQ : @@msgproperties

- La variable globale @@msgproperties stocke les propriétés relatives au message reçu pour analyser les éventuels échecs :

```
select @@msgproperties

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  <msgproperties
    MQPSUserId="&apos;user2 &apos;"
    MQPSReasonText="&apos;MQRCCF_DUPLICATE_IDENTITY&apos;"
    MQPSReason="3078"
    MQPSCompCode="2"
  </msgproperties>
```

Commandes T-SQL pour MQ : msgpropvalue

- La commande Transact-SQL msgpropvalue permet de lire les propriétés relatives au message reçu, propriétés stockées dans la variable globale @@msgproperties :

```
if (msgpropvalue('MQPSCmpCode', @@msgproperties) != "0")
begin
    print " Register/subscriber failed"
end
```

Comportement transactionnel : comportement par défaut

- Par défaut, toutes les opérations messaging (msgsend, msgrecv etc...) sont annulées si la transaction est annulée ou en échec.
- En revanche, par défaut, un échec d'une opération messaging msgsend ou msgrecv n'affecte pas la transaction parente.

Comportement transactionnel : comment contrôler ?

- Le comportement transactionnel est contrôlé par la commande `set transactional messaging` :

```
set transactional messaging [ none | simple | full ]
```

Comportement transactionnel : le mode NONE

- Avec le mode « None » les opérations Messaging et database sont indépendantes :

```
set transactional messaging none
begin tran
    msgsend (...)
    insert (...)
```

- msgsend est exécutée même si la commande insert échoue.

Comportement transactionnel : le mode SIMPLE

- Avec le mode «simple» (mode par défaut), les opérations database influent sur les opérations messaging, mais la réciproque n'est pas vraie :

```
set transactional messaging simple
begin tran
    msgsend (...)
    insert (...)
```

- La commande insert n'est pas annulée si la commande msgsend échoue, en revanche msgsend n'est pas exécutée si la commande insert est annulée (roll back).

Comportement transactionnel : le mode FULL

- Avec le mode «full», si l'opération messaging échoue, la transaction est annulée (roll back) et vice-versa si la transaction échoue, l'opération messaging est annulée :

```
set transactional messaging full
begin tran
    msgsend (...)
    insert (...)
```

- La commande insert est annulée si msgsend échoue et réciproquement si l'insertion échoue, msgsend est annulée.

Conclusion

- Il ne s'agit dans cette présentation que d'un survol des nouveautés Sybase 12.5.3a avec MQ, une documentation bien plus complète sera proposée dès que Sybase 12.5.3a est disponible.