

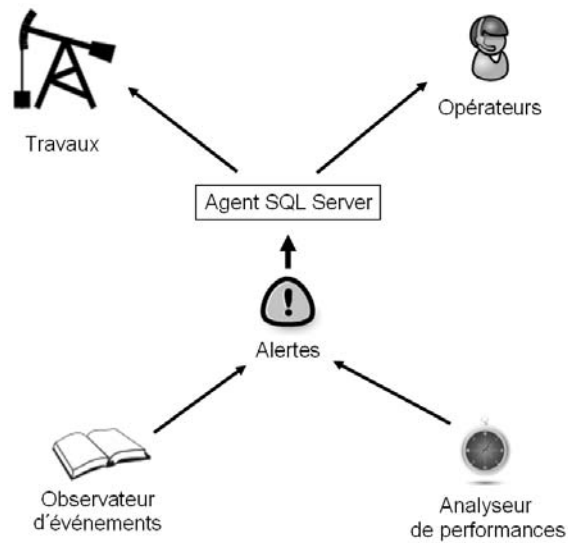
## Chapitre 5

# Tâches planifiées

### 1. Introduction

SQL Server donne la possibilité d'automatiser les tâches administratives. Il n'est bien sûr pas possible d'automatiser toutes les tâches mais les tâches planifiées représentent un bon complément à l'optimisation faite par défaut par SQL Server. De plus, avec ces tâches prédéfinies, l'administrateur possède un rôle d'anticipateur, ce qui lui donne plus de possibilités pour en tirer le meilleur tant au niveau des performances que de la fiabilité.

La gestion des tâches planifiées, des alertes et des opérateurs sont des services rendus par l'agent SQL Server. Ce service doit être démarré afin que ces éléments soient gérés. L'agent SQL Server travaille avec l'Observateur d'événements pour la gestion des erreurs SQL Server, l'Analyseur de performances pour la gestion des alertes sur des conditions de performances, et la base MSDB afin de connaître la réponse à appliquer face à une alerte, ou bien les tâches planifiées à exécuter.



### Principe de fonctionnement

Face à une alerte, l'agent peut réagir en exécutant un travail et/ou en prévenant un opérateur afin que ce dernier soit au courant du problème qui vient de surgir. Bien entendu, l'exécution d'une tâche peut conduire au déclenchement de nouvelles alertes et ainsi de suite.

D'autres tâches planifiées vont être exécutées par le service SQL Server Agent, non pas en réponse à une erreur mais sur une base de temps. Par exemple, une reconstruction des index peut être planifiée une fois par semaine dans la nuit du samedi au dimanche. L'agent SQL Server permet de réaliser une administration préventive des problèmes qui peuvent se poser lors de l'exploitation courante d'un serveur de bases de données.

## 2. Configuration des services

Comme l'exécution automatique de travaux administratifs repose sur le service SQL Server Agent, il est important que ce dernier soit correctement configuré.

### Remarque

*La configuration du service MSSQL Server a été abordée lors de l'installation.*

## 2.1 La sécurité de SQL Server Agent

Le service SQL Server Agent permet la gestion de nombreux éléments. Si le service doit posséder des droits élevés sur le serveur pour être capable de réaliser correctement toutes les tâches qui lui sont assignées, l'utilisation de ce service doit être contrôlée au plus juste. Ce contrôle est assuré par les trois rôles de base de données définis sur la base msdb :

- **SQLAgentUserRole** : peuvent créer leurs propres travaux
- **SQLAgentReaderRole** : peuvent en plus lister tous les travaux du serveur
- **SQLAgentOperatorRole** : ont tous les droits sur la gestion des travaux, alertes et opérateurs

L'appartenance à ces rôles n'est nécessaire que pour les utilisateurs non-membres du rôle de serveur sysadmin.

Par exemple, si un utilisateur se connecte à la console graphique SQL Server Management Studio sans être membre de l'un de ces trois rôles, alors l'outil ne présentera tout simplement pas le nœud relatif à SQL Server Agent. Ainsi, l'utilisateur n'est pas capable de modifier, ni même de connaître le travail réalisé au niveau de l'automatisation de tâches. Le même niveau de sécurité est défini au niveau Transact SQL.

## 2.2 Configuration de la messagerie

La gestion des mails avec SQL Server passe par la messagerie de base de données. Cette fonctionnalité peut servir dans le cadre des bases de données utilisateurs afin d'envoyer des mails liés aux applications utilisateurs correspondantes ou bien dans le cadre de l'agent SQL Server pour des notifications de réussite et/ou échec aux administrateurs.

Le service de mail de base de données utilise le protocole standard SMTP pour envoyer les mails. Il ne repose pas sur MAPI, ce qui rend facultatif l'installation d'un client de messagerie comme Outlook. Au travers de ce protocole, le service de mail de base de données prend en charge l'envoi de mails au format HTML.

La messagerie de base de données n'est pas active par défaut, aussi il faut l'activer soit avec l'assistant de configuration au travers de SQL Server Management Studio, soit par l'intermédiaire de scripts Transact SQL en faisant appel à des instructions et des procédures stockées spécifiques. Ce paramétrage permettra de créer des profils de messagerie associés à un ou plusieurs comptes de messagerie.

Compte tenu du caractère ponctuel de cette action de configuration, seul le mode graphique est présenté dans cet ouvrage.

Il existe deux types de profils :

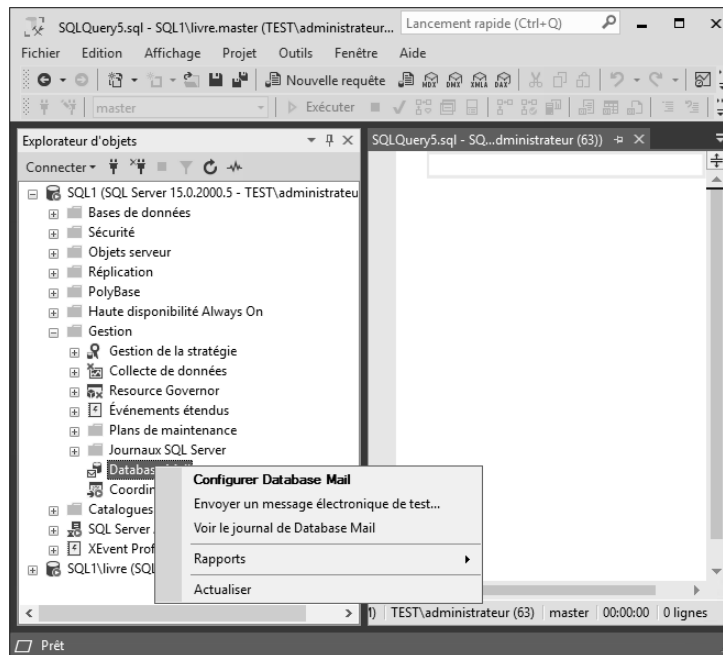
- Les profils publics, qui sont accessibles à tous les utilisateurs qui appartiennent au rôle **DatabaseMailUserRole** sur la base msdb. C'est parce qu'un utilisateur est membre de ce rôle qu'il peut envoyer des messages électroniques.
- Les profils privés, également définis sur la base msdb, mais cette fois-ci chaque utilisateur doit être précisément identifié afin de pouvoir utiliser le profil.

### 2.2.1 Configuration depuis SQL Server Management Studio

L'assistant de configuration de la messagerie de base de données va permettre de réaliser simplement et en étant guidé l'une des actions suivantes :

- Configurer la messagerie de base de données.
- Gérer les comptes de la messagerie de base de données.
- Gérer les profils de sécurité.
- Gérer les paramètres système.

L'assistant est lancé depuis SQL Server Management Studio en sélectionnant **Configurer Database Mail** dans le menu contextuel associé au nœud **Gestion - Database Mail** de l'instance SQL Server sur laquelle la configuration doit être faite.



Le premier écran de l'assistant (après l'écran d'accueil) permet de sélectionner l'action que l'on souhaite réaliser avec l'assistant. La première étape consiste à configurer Database Mail.

Pour que la messagerie puisse fonctionner, il faut activer le composant en sélectionnant **Oui** à la question posée. La création de profils peut commencer.

Assistant Configuration de Database Mail - SQL1

**Nouveau profil**  
Spécifiez le nom, la description, les comptes et la priorité de basculement du profil.

Nom de profil :

Description :

Il se peut qu'un profil soit associé à plusieurs comptes SMTP. Si un compte échoue lors de l'envoi d'un message électronique, le profil utilise le compte suivant dans la liste des priorités. Spécifiez les comptes associés au profil, puis déplacez les comptes pour définir la priorité de basculement.

Comptes SMTP :

Pri...	Nom du com...	Adresse de messagerie
--------	---------------	-----------------------

Ajouter...  
Supprimer  
Monter  
Descendre

Aide < Précédent Suivant > Terminer >> Annuler

### *Définition d'un premier profil*

En utilisant le bouton **Ajouter**, il est possible de définir un ou plusieurs comptes de courrier.

Nouveau compte Database Mail

Spécifiez le nom, la description et les attributs de votre compte SMTP.

Nom du compte : Livre

Description :

Serveur de courrier sortant (SMTP)

Adresse de messagerie : sql@test.local

Nom complet :

Répondre au courrier :

Nom du serveur : localhost Numéro du port : 25

Ce serveur nécessite une connexion sécurisée (SSL).

Authentification SMTP

Authentification Windows à l'aide d'informations d'identification du service Moteur de base de données

Authentification de base

Nom d'utilisateur :

Mot de passe :

Confirmer le mot de passe :

Authentification anonyme

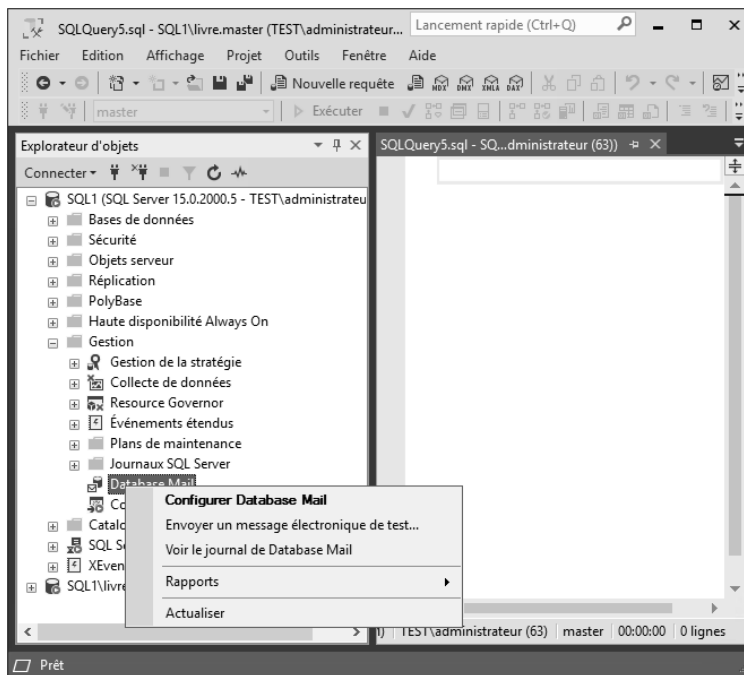
OK Annuler Aide

### Définition d'un compte mail

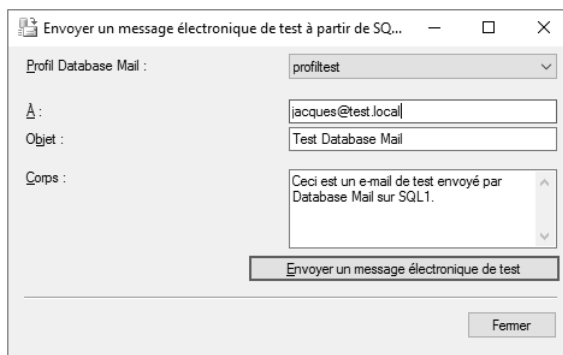
Par la suite, l'assistant propose simplement de rendre public le profil, c'est-à-dire accessible à l'ensemble des utilisateurs du serveur. Enfin, l'assistant se termine par un écran de synthèse qui résume les différentes opérations demandées. La validation de cette synthèse entraîne la création du profil.

### 2.2.2 Tester le service

Seuls les utilisateurs membres du rôle de serveur **sysadmin** ou bien du rôle de base de données **databaseMailUserRole** défini sur la base **msdb**, peuvent envoyer des mails. Il est facile de tester le profil, en sélectionnant **Envoyer un message électronique de test** depuis le menu contextuel associé au nœud **Database Mail** depuis l'explorateur d'objets de SQL Server Management Studio.



Un écran permet alors de préciser le profil à utiliser ainsi que le destinataire du message de test.



La messagerie est maintenant opérationnelle et il est possible d'envoyer des mails à l'aide de la procédure stockée `sp_send_dbmail`.