

Les emails avec SQL Server 2005

La gestion des emails avec la SQL Server 2005



Après avoir rédigé le précédent article concernant :

- [Les emails avec SQL Server 2000 et MSDE \(FR\)](#)

Il fallait traiter des modifications apportées avec la nouvelle mouture de SQL Server. Nous verrons donc comment dans effectuer ces envois depuis le nouveau moteur SQL Server.

Introduction

Lors du développement de la nouvelle version du moteur relationnel de Microsoft, une plainte classique des utilisateurs de SQL Server 2000 était de devoir installer un client Outlook afin de paramétrer l'envoi des emails. Ainsi les administrateurs système comprenaient mal le besoin de l'application cliente sur un serveur.

De plus, de nombreuses possibilités ont été ouvertes avec l'intégration du Framework .NET 2.0 dans le moteur, de ce fait pourquoi ne pas utiliser la possibilité classique offerte aux développeurs WEB :

- [L'utilisation de l'envoi d'email via le protocole SMTP \(FR\)](#)

Avec la version SQL Server 2005, nous avons deux méthodes pour faire ces envois de mail :

- La couche MAPI (pour des raisons de compatibilité)
- Le protocole SMTP

Nous ferons un petit aparté pour le cas précis de SQL Server 2005 Express Edition afin de voir comment effectuer un envoi de mail via celui-ci.

Présentation

Nous verrons dans cet article le paramétrage de ces deux méthodes grâce au nouvel outil mis à disposition avec le nouveau moteur :

- SQL Server Management Studio

Tout cet article est réalisé depuis la version SQL Server 2005 Developer Edition française (sortie au début du mois de Février 2006). Celle-ci correspond à la version Enterprise Edition sans la couche haute disponibilité et Clustering. De ce fait, tout ce qui est réalisable sur celle-ci est applicable sur un serveur de production.

Avec la version 2005 de SQL Server et l'augmentation générale du niveau de sécurité, les envois d'emails sont désactivés dans SQL Server 2005. De plus, SQL Mail (passage par le client MAPI) a été maintenu uniquement pour des raisons de compatibilités descendantes. Microsoft déconseille d'utiliser cet outil et invite à passer sur "la messagerie de la base de données"(utilisation du protocole SMTP).

Utilisation de SQL Mail

La première possibilité respecte le même principe que pour la version 2000, vous pouvez donc reprendre le paramétrage qui est disponible :

- **Le paramétrage de l'agent SQL Server**

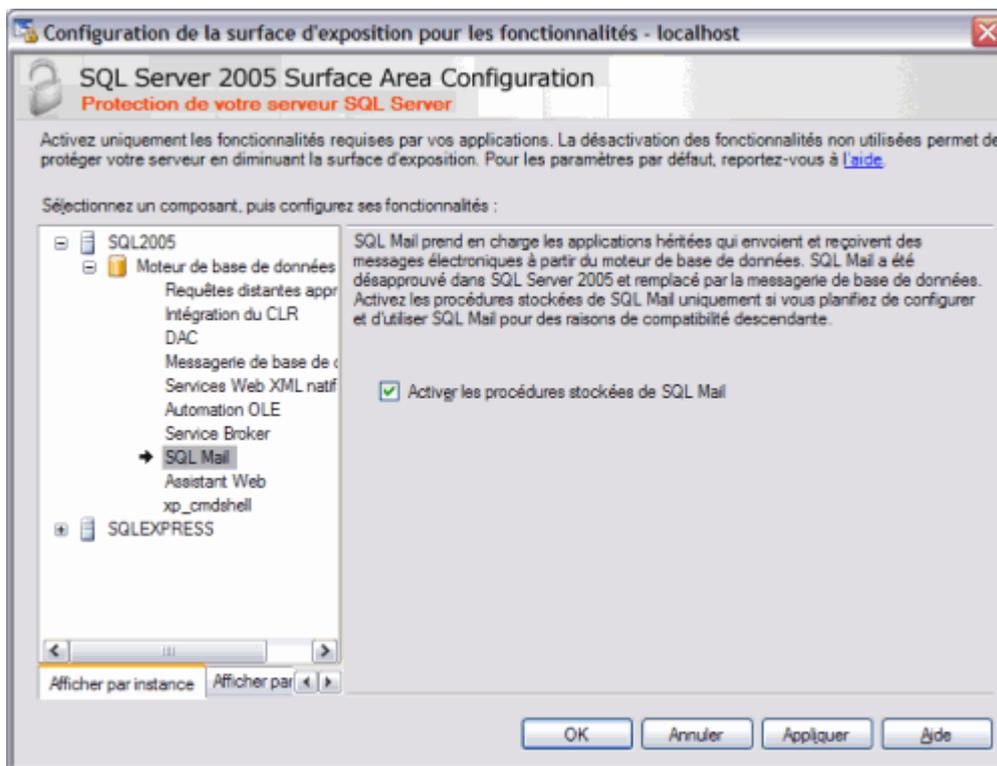
Une fois que le profil Outlook est fait et que le service "SQL Server Agent (SQL2005)" est paramétré pour fonctionner avec le même compte que celui du Profil Outlook, vous pouvez passer au paramétrage proprement dit.

Activation de l'option SQL Mail

Par défaut SQL Mail est désactivé, il faut donc aller dans les options de configuration pour modifier cette option :

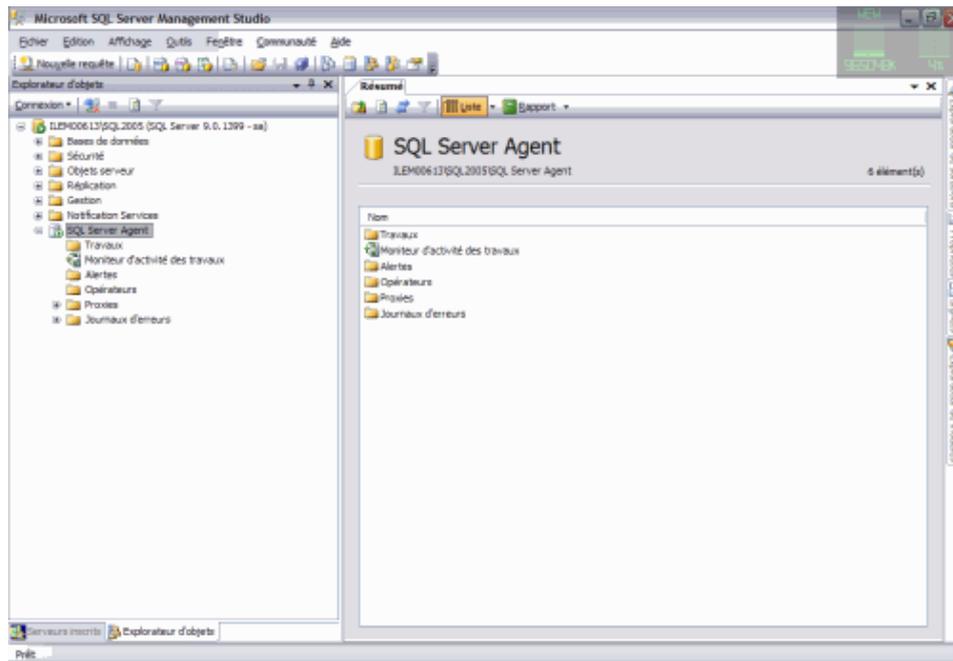
- Démarrer > Programmes > Microsoft SQL Server 2005 > Outils de configuration > Configuration de la surface d'exposition SQL Server

Au lancement de l'outil, celui-ci fait un scan des installations SQL Server 2005 disponibles en local et charge les paramètres de chacune. On trouve donc dans l'instance SQL Server 2005 l'option "SQL Mail".

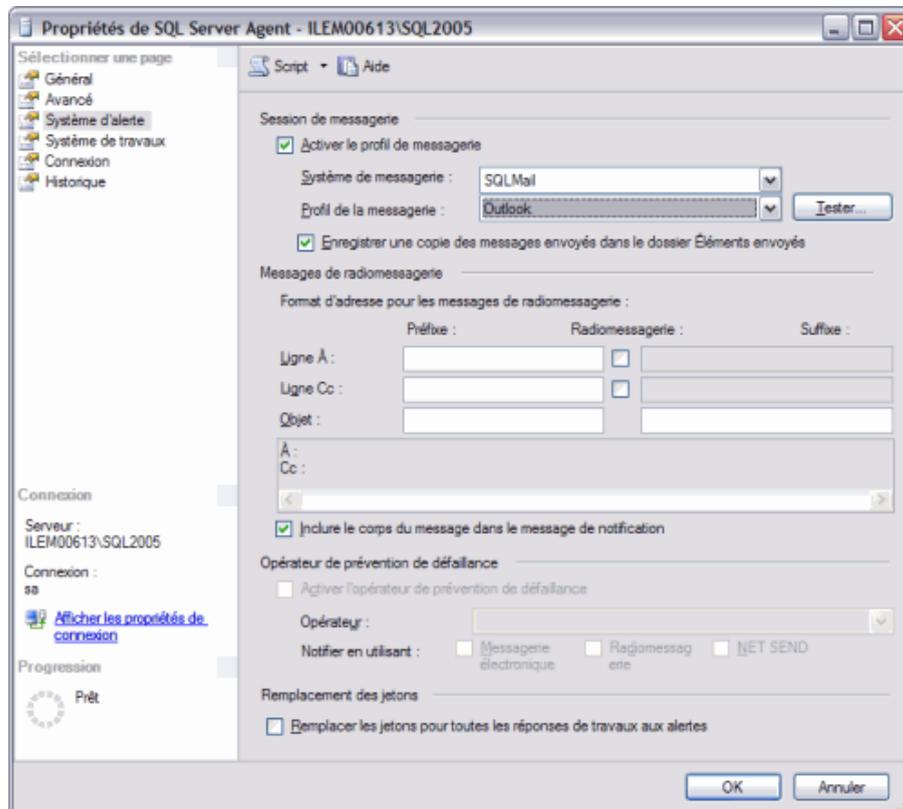


Une fois que l'option est sélectionnée, on peut valider la modification en cliquant sur OK et passer à l'étape suivante. Il est conseillé de relancer le service "Agent SQL" afin de bien prendre en compte ces modifications.

Dans SQL Server Management Studio, on se connecte sur le "Moteur de base de données" SQL Server 2005, puis développe dans l'explorateur d'objets jusqu'à voir "SQL Server Agent".



On clique alors avec le bouton droit sur cet objet et sélectionne "propriétés". Il faut alors choisir la rubrique (sur la gauche) "Système d'alerte".



Il faut alors cocher "Activer le profil de messagerie", puis sélectionner dans le menu déroulant "SQL Mail" et le profil Outlook correspondant. On peut alors le tester et définir si on conserve les mails dans les éléments envoyés. On valide les modifications en cliquant sur OK.

L'intérêt de cette fonctionnalité est clairement lié aux besoins lors des migrations. En effet, comme nous l'avons vu dans le précédent article, il est obligatoire d'effectuer cette configuration sur SQL Server 2000 pour utiliser les procédures stockées d'envoi de Mail.

Ainsi, ceux qui ont fait cet usage se seraient retrouvés pénalisés lors de la migration de leur application sur la nouvelle version.

Microsoft a choisi de faciliter cette migration et donc a conservé cet usage pour les projets l'ayant utilisé cette fonctionnalité.

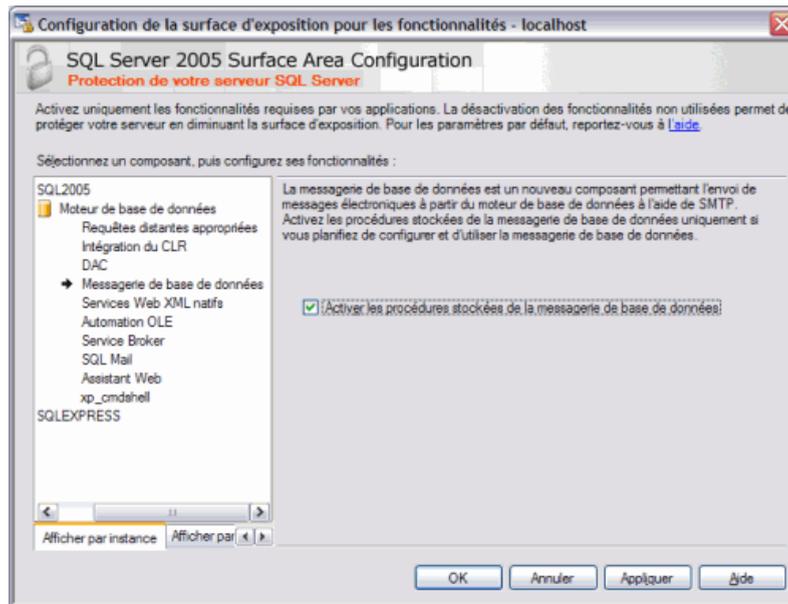
Il est en revanche bien spécifié que ce n'est désormais pas une configuration conseillée par Microsoft, la nouvelle plateforme intègre sa propre solution d'envoi de mail que nous allons voir maintenant.

Messagerie de la base de données

Une des nouveautés de SQL Server 2005 est liée à cette nouvelle possibilité. En effet, comme nous venons de le voir, la solution SQL Mail nécessite un client Outlook avec une configuration sur Exchange. Ce sont donc des exigences que de nombreuses entreprises ne peuvent respecter.

De plus, Microsoft a la volonté de percer dans les petites structures avec la version Workgroup, ainsi une autre solution était obligatoire.

Pour effectuer cette configuration, il faut commencer comme pour SQL Mail par l'outil de "Configuration de la surface d'exposition SQL Server".

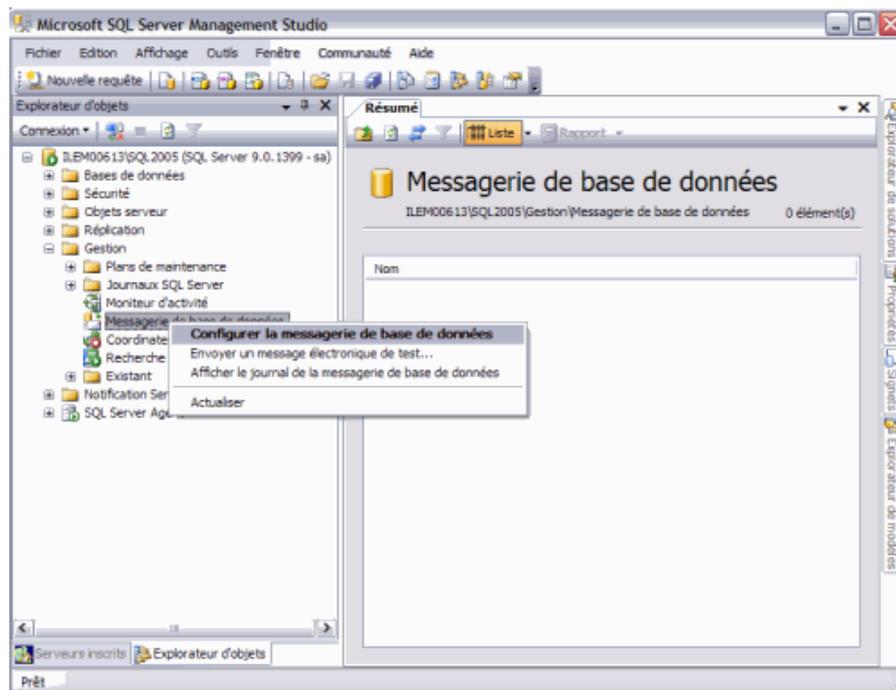


Une fois l'option validée, il faut retourner dans l'outil de gestion de SQL Server 2005 :

- SQL Server Management Studio

Une fois Management Studio ouvert, on se connecte sur notre SQL Server 2005. Il faut alors développer dans l'explorateur d'objets "Gestion". On voit alors apparaître "Messagerie de base de données".

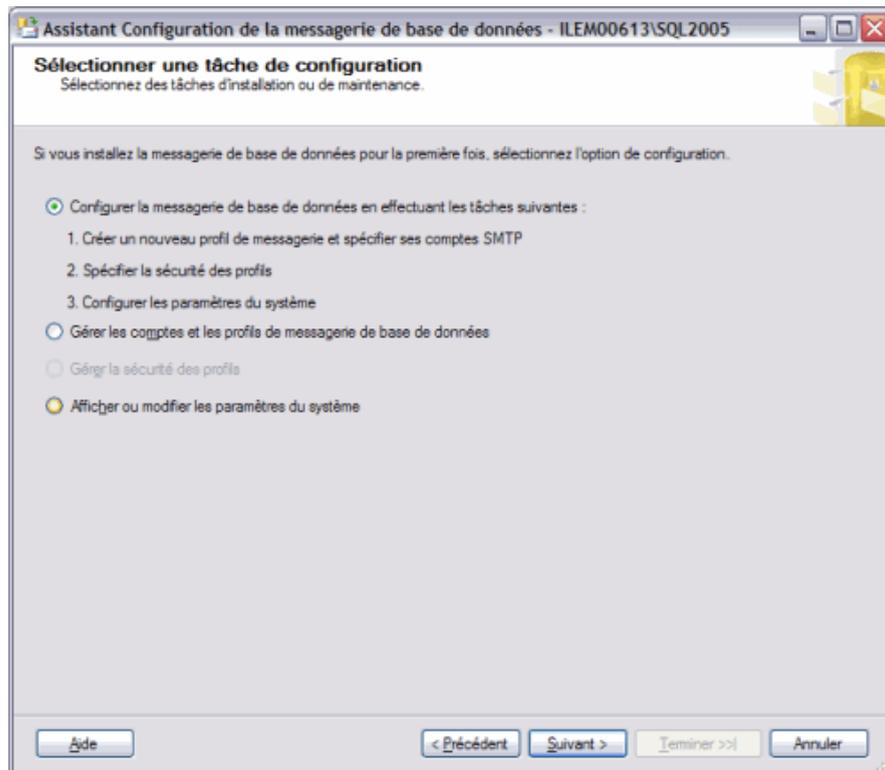
On clique avec le bouton droit de la souris et sélectionne "Configurer la messagerie de base de données".



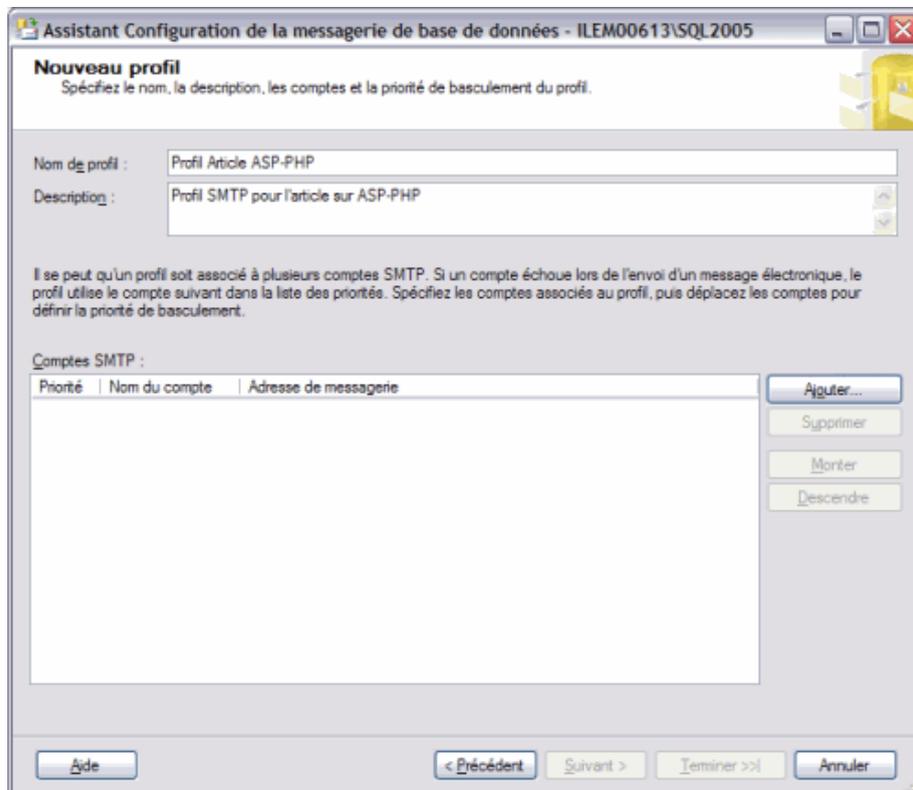
On voit apparaître un assistant de configuration, dont le premier écran nous explique ce que nous pourrons y faire.



L'écran suivant nous demande ce que nous souhaitons faire maintenant, nous choisirons la première possibilité "Configurer la messagerie de base de données"



On déclare un nom et une description de notre profil.



Puis on clique sur "ajouter", afin de définir les paramètres nécessaires à l'envoi des emails. Dans cet écran, il nous faut définir les paramètres de l'adresse email que le destinataire verra comme Emetteur, mais surtout l'adresse et le port du serveur SMTP relais. Dans le cas où votre relais exige une authentification, vous devez le spécifier ici.

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

Nom du compte :

Description :

Serveur de courrier sortant (SMTP)

Adresse de messagerie :

Nom complet :

Répondre au courrier :

Nom du serveur : Numéro du port :

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

Dans le cas où on a plusieurs serveurs SMTP, on peut les définir de la même façon ici en ajoutant autant de relais que l'on souhaite. Cela permettra à SQL Server de tester au préalable s'il peut émettre sur le premier, puis en cas d'échec le second, etc ...

Assistant Configuration de la messagerie de base de données - ILEM00613\SQL2005

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.

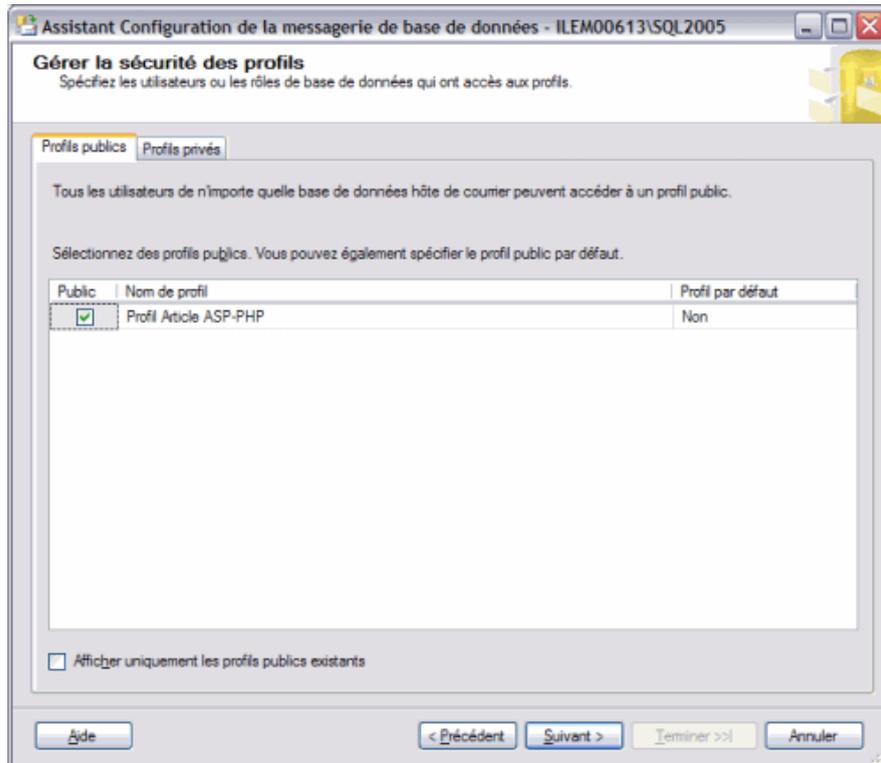
Comptes SMTP :

| Priorité | Nom du compte | Adresse de messagerie |
|----------|-------------------------|-----------------------|
| 1 | Compte SMTP sur Article | fromelard@hotmail.com |
| 2 | Compte SMTP Article 2 | fromelard@hotmail.com |

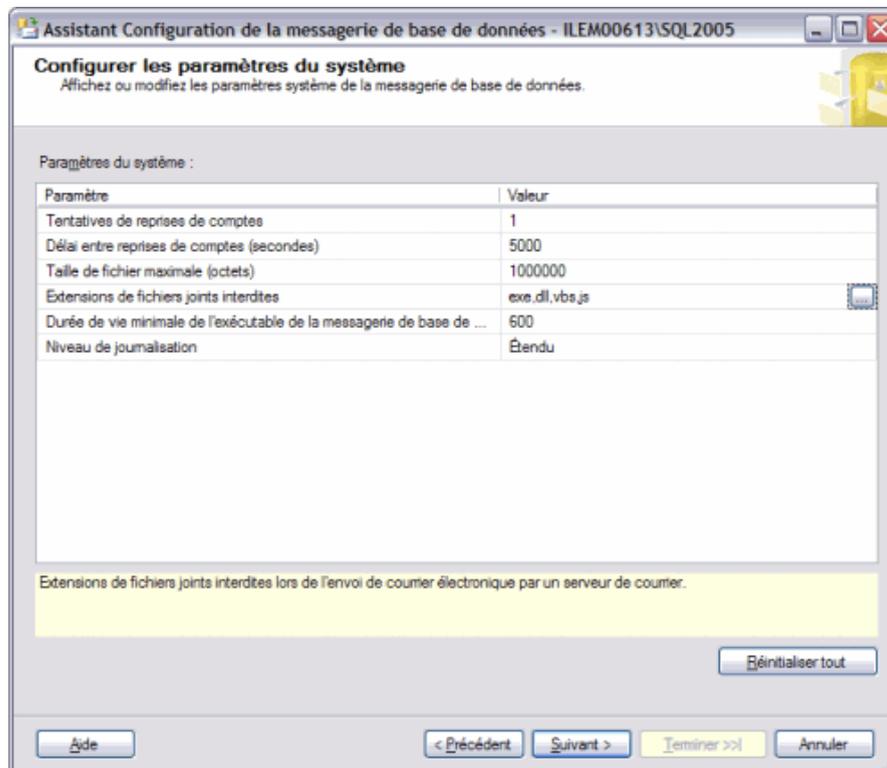
Ajouter...
Supprimer
Monter
Descendre

Aide < Précédent Suivant > Terminer >> Annuler

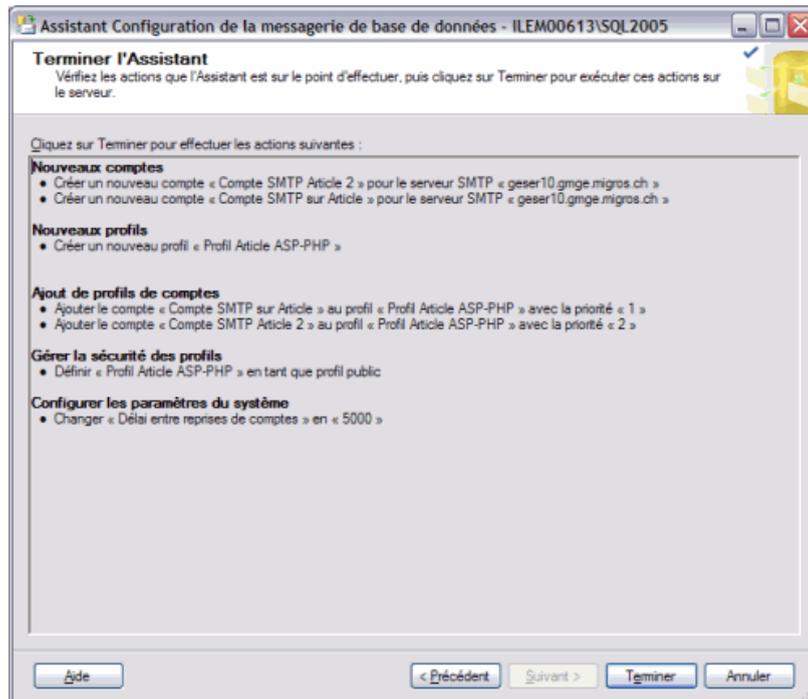
Maintenant, on doit spécifier si le profil SMTP que l'on vient de créer est public ou privé. Le fait d'être public permet que ce profil soit utilisable depuis toutes les bases de données hôtes du courrier. Dans le cas d'un profil privé, le profil ne sera utilisable que par l'utilisateur que l'on aura choisi. Dans notre cas, nous souhaitons rendre ce profil public.



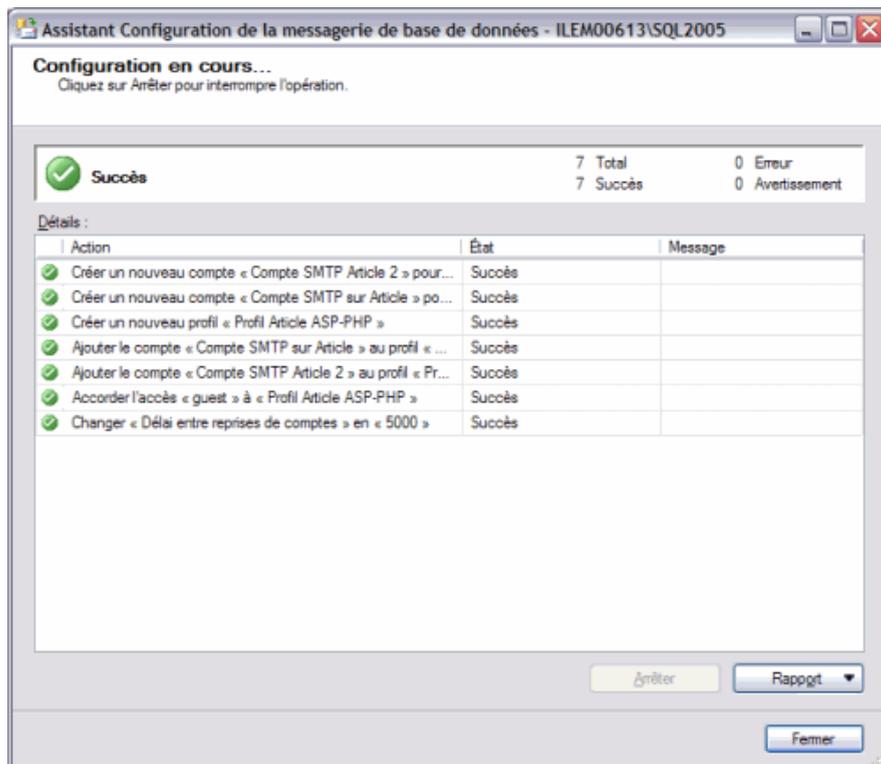
L'écran suivant nous donne un résumé des paramètres à suivre par le serveur lors d'une tentative d'envoi de mail (les fichiers joints à interdire, le nombre de tentatives, ...).



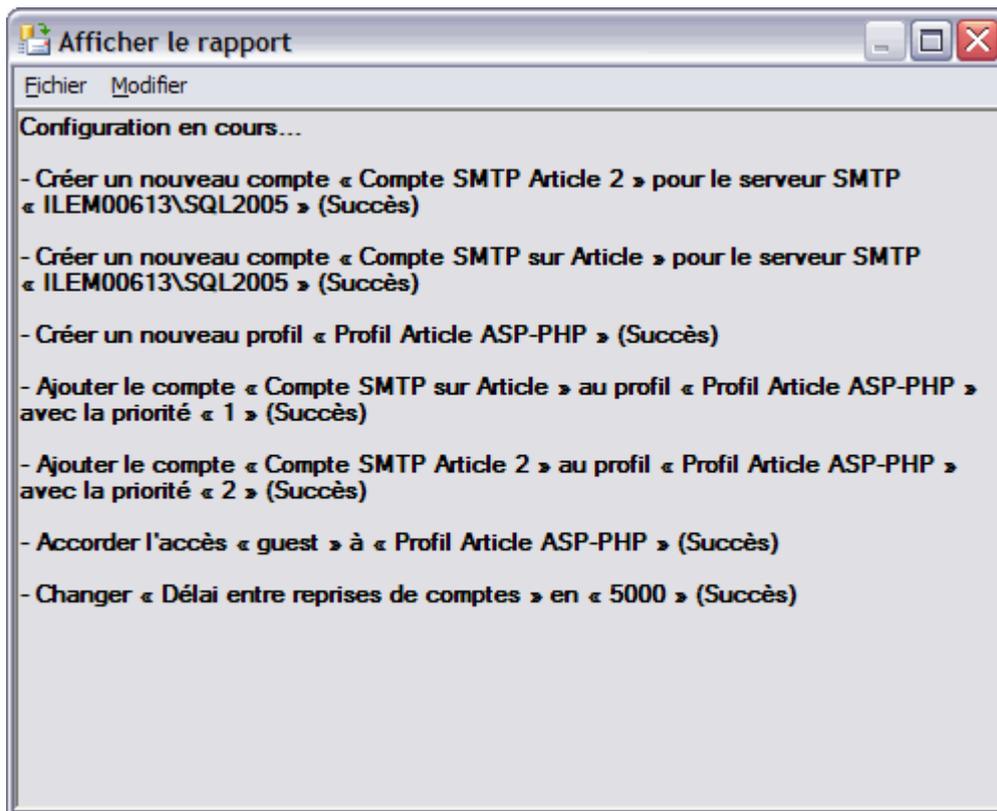
On a alors un récapitulatif de la configuration sélectionnée.



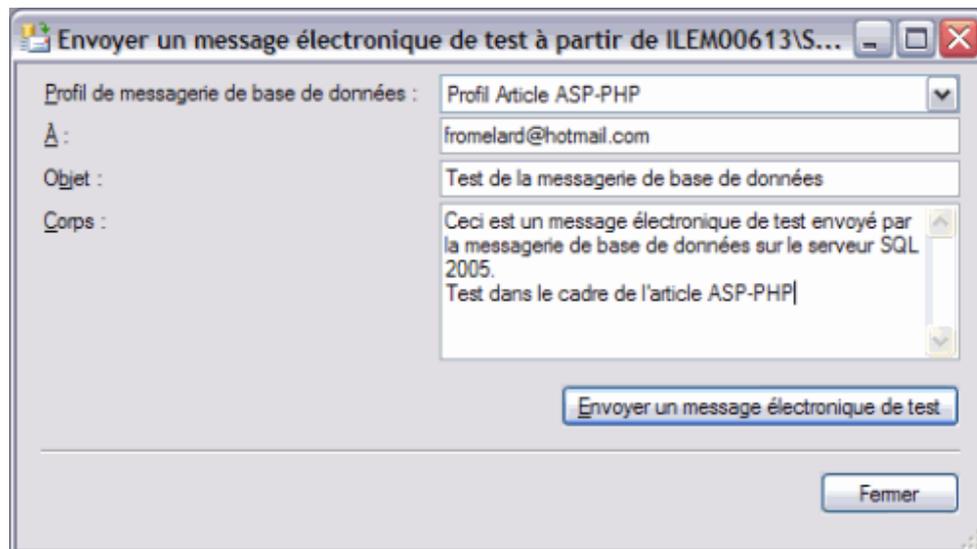
Puis le rapport d'exécution de cette configuration.



Ce rapport peut être extrait au format texte pour une visualisation plus détaillée.



Maintenant que cette configuration est appliquée, nous pouvons effectuer un test d'envoi en cliquant avec le bouton droit sur "Messagerie de base de données" et sélectionnant "Envoyer un message électronique de test ...". On obtient alors un écran nous demandant l'adresse Email de test et le texte à envoyer.



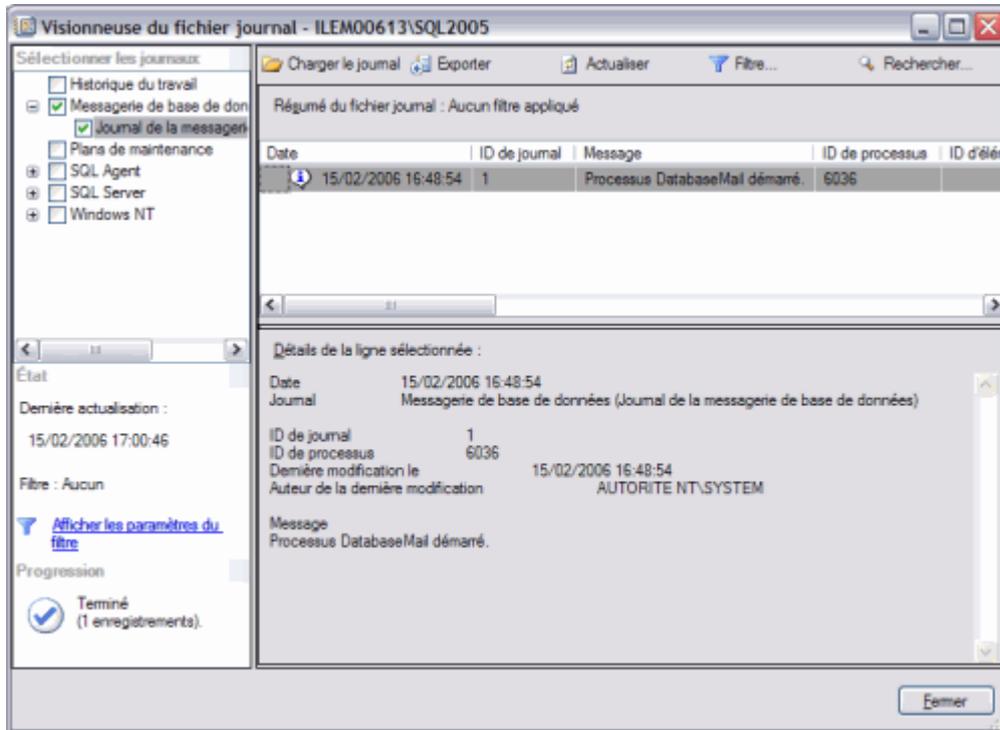
En cliquant sur le bouton "Envoyer un message électronique de test", nous avons une console nous demandant de valider la réception de ce mail par OK.



Vous devez contrôler que le mail a bien été reçu. Attention, dans certains cas, votre mail peut être interprété comme étant indésirable (cas de Hotmail notamment).



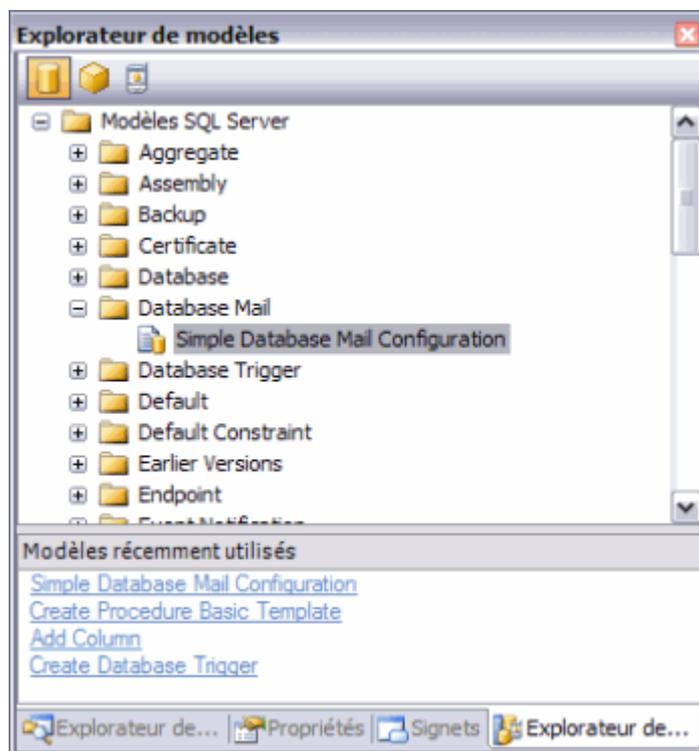
Une fois cette configuration effectuée et testée, vous pouvez visionner le journal de cette messagerie, en cliquant avec le bouton droit de la souris sur "Messagerie de base de données" et sélectionnant "Afficher le journal de la messagerie de la base de données". Vous aurez alors l'écran suivant qui reprend les journaux (comme ceux de Windows) du moteur SQL Server 2005.



Il est aussi possible de configurer cet outil en console via une requête système.

Configuration via la commande TSQL

Dans le cadre de mise en place de configuration sur des serveurs où l'outil client n'est pas installé, on peut configurer via une requête TSQL qui est fournie dans le Management Studio.



Script TSQL

```
-----  
-- Database Mail Simple Configuration Template.  
--  
-- This template creates a DatabaseMail profile, an SMTP account and  
-- associates the account to the profile.  
-- The template does not grant access to the new profile for  
-- any database principals. Use msdb.dbo.sysmail_add_principalprofile  
-- to grant access to the new profile for users who are not  
-- members of sysadmin.  
-----  
  
DECLARE @profile_name sysname,  
        @account_name sysname,  
        @SMTP_servername sysname,  
        @email_address NVARCHAR(128),  
        @display_name NVARCHAR(128);  
  
-- Profile name. Replace with the name for your profile  
SET @profile_name = '<profile_name,sysname,SampleProfile>';  
  
-- Account information. Replace with the information for your account.  
  
SET @account_name = '<account_name,sysname,SampleAccount>';  
SET @SMTP_servername = '<SMTP_servername,sysname,your SMTP server name>';  
SET @email_address = '<email_address,nvarchar(128),sender email address>';  
SET @display_name = '<display_name,nvarchar(128),name of the sender>';  
  
-- Verify the specified account and profile do not already exist.  
IF EXISTS (SELECT * FROM msdb.dbo.sysmail_profile WHERE name = @profile_name)  
BEGIN  
    RAISERROR('The specified DatabaseMail profile (<profile_name,sysname,SampleProfile>) already exists.', 16, 1);  
    GOTO done;  
END;  
END;
```

```

IF EXISTS (SELECT * FROM msdb.dbo.sysmail_account WHERE name = @account_name )
BEGIN
RAISERROR('The specified DatabaseMail account (<account_name,sysname,SampleAccount>) already exists.', 16, 1) ;
GOTO done;
END;

-- Start a transaction before adding the account and the profile
BEGIN TRANSACTION ;

DECLARE @rv INT;

-- Add the account
EXECUTE @rv=msdb.dbo.sysmail_add_account_sp
@account_name = @account_name,
@email_address = @email_address,
@display_name = @display_name,
@mailserver_name = @SMTP_servername;

IF @rv<>0
BEGIN
RAISERROR('Failed to create the specified Database Mail account
(<account_name,sysname,SampleAccount>).', 16, 1) ;
GOTO done;
END

-- Add the profile
EXECUTE @rv=msdb.dbo.sysmail_add_profile_sp
@profile_name = @profile_name ;

IF @rv<>0
BEGIN
RAISERROR('Failed to create the specified Database Mail profile
(<profile_name,sysname,SampleProfile>).', 16, 1);
ROLLBACK TRANSACTION;
GOTO done;
END;

-- Associate the account with the profile.
EXECUTE @rv=msdb.dbo.sysmail_add_profileaccount_sp
@profile_name = @profile_name,
@account_name = @account_name,
@sequence_number = 1 ;

IF @rv<>0
BEGIN
RAISERROR('Failed to associate the specified profile with the specified account
(<account_name,sysname,SampleAccount>).', 16, 1) ;
ROLLBACK TRANSACTION;
GOTO done;
END;

COMMIT TRANSACTION;

done:
GO

```

Quelque soit la possibilité choisie (script ou visuel), votre serveur SQL est maintenant capable d'envoyer des emails. Nous allons maintenant voir comment utiliser cette possibilité.

Usage de Database Mail

Les opérateurs et les alertes

Une fois la configuration effectuée avec succès, vous pouvez ajouter des opérateurs par Management Studio :

- SQL Server Agent > Opérateurs

Vous pouvez aussi créer un opérateur via le script suivant :

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_operator @name=N'Romelard Fabrice [MVP]',
    @enabled=1,
    @pager_days=0,
    @email_address=N'fromelard@hotmail.com'
GO
```

Une fois ce nouvel opérateur créé, nous pouvons définir une alerte qui préviendra par Email celui-ci lors d'un évènement choisi.

Ceci se fait graphiquement dans Management Studio dans la rubrique :

- SQL Server Agent > Alertes

On peut aussi utiliser le script suivant. Notre exemple enverra un email contenant le texte "Test d'alerte", si la base "AdventureWorks" reçoit une erreur fatale matérielle (code interne 24).

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_alert @name=N'Test d''alerte',
    @message_id=0,
    @severity=24,
    @enabled=1,
    @delay_between_responses=0,
    @include_event_description_in=1,
    @database_name=N'AdventureWorks',
    @notification_message=N'Test d''alerte',
    @job_id=N'00000000-0000-0000-0000-000000000000'
GO
EXEC msdb.dbo.sp_add_notification @alert_name=N'Test d''alerte',
    @operator_name=N'Romelard Fabrice',
    @notification_method = 1
GO
```

Utilisation de la procédure stockée

Deux cas se présentent suivant la configuration sélectionnée, si vous avez installé, comme pour la version 2000, SQL Mail, vous pouvez toujours utiliser la procédure stockée "xp_sendmail". Cela vous donnera un script comme suit :

```
EXEC master.dbo.xp_sendmail
@recipients=N'fromelard@hotmail.com,
@message=N'Message à envoyer.',
@subject=N'Sujet du message' ;
GO
```

Dans le cas où SQL Mail n'a pas été installé au profit de la "Messagerie de la base de données", vous aurez le message d'erreur suivant :

Msg 15281, Niveau 16, État 1, Procédure xp_sendmail, Ligne 1

SQL Server a bloqué l'accès à procédure 'sys.xp_sendmail' du composant 'SQL Mail XPs' car ce composant est désactivé dans le cadre de la configuration de la sécurité du serveur. Un administrateur système peut activer l'utilisation de 'SQL Mail XPs' via sp_configure. Pour plus d'informations sur l'activation de 'SQL Mail XPs', voir « Configuration de la surface d'exposition » dans la documentation en ligne de SQL Server.

Ainsi, vous êtes dans la configuration avec l'envoi via SMTP. La procédure stockée est alors différente, mais surtout n'est plus intégrée dans la base "master". On la trouve donc dans la base "msdb". Il s'agit de :

- sp_send_dbmail

Dont la syntaxe est la suivante :

```
sp_send_dbmail [ [ @profile_name = ] 'profile_name' ]
[ , [ @recipients = ] 'recipients [ ; ...n ]' ]
[ , [ @copy_recipients = ] 'copy_recipient [ ; ...n ]' ]
[ , [ @blind_copy_recipients = ] 'blind_copy_recipient [ ; ...n ]' ]
[ , [ @subject = ] 'subject' ]
[ , [ @body = ] 'body' ]
[ , [ @body_format = ] 'body_format' ]
[ , [ @importance = ] 'importance' ]
[ , [ @sensitivity = ] 'sensitivity' ]
[ , [ @file_attachments = ] 'attachment [ ; ...n ]' ]
[ , [ @query = ] 'query' ]
[ , [ @execute_query_database = ] 'execute_query_database' ]
[ , [ @attach_query_result_as_file = ] attach_query_result_as_file ]
[ , [ @query_attachment_filename = ] query_attachment_filename ]
[ , [ @query_result_header = ] query_result_header ]
[ , [ @query_result_width = ] query_result_width ]
[ , [ @query_result_separator = ] 'query_result_separator' ]
[ , [ @exclude_query_output = ] exclude_query_output ]
[ , [ @append_query_error = ] append_query_error ]
[ , [ @query_no_truncate = ] query_no_truncate ]
[ , [ @mailitem_id = ] mailitem_id ] [ OUTPUT ]
```

Voici d'ailleurs un exemple de son utilisation :

```
DECLARE @tableHTML NVARCHAR(MAX) ;

SET @tableHTML =
N'<H1>Work Order Report</H1>' +
N'<table border="1">' +
N'<tr><th>Work Order ID</th><th>Product ID</th>' +
N'<th>Name</th><th>Order Qty</th><th>Due Date</th>' +
N'<th>Expected Revenue</th></tr>' +
CAST ( ( SELECT td = wo.WorkOrderID, "",
td = p.ProductID, "",
td = p.Name, "",
td = wo.OrderQty, "",
td = wo.DueDate, "",
td = (p.ListPrice - p.StandardCost) * wo.OrderQty
FROM AdventureWorks.Production.WorkOrder as wo
JOIN AdventureWorks.Production.Product AS p
ON wo.ProductID = p.ProductID
```

```
WHERE DueDate > '2004-04-30'  
AND DATEDIFF(dd, '2004-04-30', DueDate) < 2  
ORDER BY DueDate ASC,  
      (p.ListPrice- p.StandardCost) * wo.OrderQty DESC  
FOR XML PATH('tr'), TYPE  
) AS NVARCHAR(MAX) ) +  
N'</table>' ;  
  
EXEC msdb.dbo.sp_send_dbmail @profile_name='Profil Article ASP-PHP',  
  @recipients='danw@Adventure-Works.com',  
  @subject = 'Work Order List',  
  @body = @tableHTML,  
  @body_format = 'HTML' ;
```

Nous avons vu pour le moment toute la configuration pour SQL Server. Toutes les versions, excepté Express Edition respectent ce qui a été dit jusqu'à maintenant. Nous allons donc nous attarder au cas de SQL Server 2005 Express Edition.

Cas de SQL Server 2005 Express Edition

Dans la configuration de base, SQL Server 2005 Express Edition ne supporte pas Database Mail. N'étant pas fourni avec le service SQL Server Agent, il n'était déjà pas possible d'utiliser la configuration avec le client Outlook.

En regardant plus en détail, on sait que Database Mail s'appuie sur le Framework 2.0 et qu'Express est aussi basé sur le Framework 2.0, ainsi quelle est la raison pour laquelle cela ne fonctionne pas. J'ai donc cherché à configurer cette option sur la version Express.

Toutes les recherches ont été effectuées depuis le Book Online de SQL Server 2005, que vous pouvez trouver ici :

- [Documentation en ligne de SQL Server 2005 \(FR\)](#)

Ainsi, on trouve :

- Procédure : créer des profils publics de messagerie de base de données (Transact-SQL)

Qui nous fourni donc tout le script TSQL permettant de créer un profil pour DataBase Mail.

```
-- Create a DatabaseMail account
EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name = 'Profil_SQL_Express_Account_Demo',
    @description = 'Mail account for use by all database users.',
    @email_address = 'fromelard@hotmail.com',
    @replyto_address = 'fromelard@hotmail.com',
    @display_name = 'Profil_SQL_Express Mailer',
    @mailserver_name = 'smtp.votreserver.com' ;

-- Create a DatabaseMail profile
EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name = 'Profil_SQL_Express_Demo',
    @description = 'Profil pour mail administratifs.' ;

-- Add the account to the profile
EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name = 'Profil_SQL_Express_Demo',
    @account_name = 'Profil_SQL_Express_Account_Demo',
    @sequence_number = 1 ;

-- Grant access to the profile to all users in the msdb database
EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name = 'Profil_SQL_Express_Demo',
    -- @database_name = 'msdb',
    @principal_name = 'public',
    @is_default = 1 ;
```

Attention :

Il faut jouer ce script par bloc et non en une seule fois.

Cette configuration nous permet de paramétrer Database Mail, mais celui-ci n'est pas activé, en effet, pour les versions Express l'outil "Configuration de la surface d'exposition SQL Server" ne nous affiche pas les paramètres de mail. Il nous faut donc voir comment faire autrement.

La documentation nous fournit la méthode avec l'utilitaire de configuration en ligne de commande :

- Utilitaire sac

L'objectif de cet outil est de permettre aux DBA de faire des configurations en "batterie". Ainsi, cet outil permet d'extraire une configuration de SQL Server 2005 dans un fichier (au format XML), mais aussi d'appliquer une configuration depuis un fichier XML.

Ainsi dans notre cas, nous allons extraire la configuration de notre moteur relationnel SQL Server 2005 Express Edition, puis modifier la clé spécifique à DataBase Mail et appliquer cette modification. Il ne restera plus qu'à relancer le service NT.

Cet outil se trouve dans le répertoire :

- C:\Program Files\Microsoft SQL Server\90\Shared\

La commande d'exportation vers un fichier "serverExpress.out" sera la suivante :

- sac out serverExpress.out -S localhost -U sa -P Motdepasse -I SQLEXPRESS

Ensuite on ouvre le fichier "serverExpress.out" avec notepad et modifie la ligne :

```
....  
<anyType xsi:type="Pair">  
<Key>DatabaseMail</Key>  
<Value xsi:type="xsd:int">0</Value>  
</anyType>  
....
```

Cette clé doit donc être fixée à 1. Une fois cette modification faite, on enregistre le fichier et exécute la commande sac inverse :

- sac in serverExpress.out -S localhost -U sa -P Motdepasse -I SQLEXPRESS

On relance le service NT "SQL Server (SQLEXPRESS)". La configuration est alors terminée. Le soucis est que le moteur est correctement paramétré, mais on observe encore des erreurs dans la table :

```
SELECT * FROM dbo.sysmail_log  
-----  
3 3 2006-02-16 00:49:23.717 Activation failure. NULL NULL NULL 2006-02-16 00:49:23.717 sa  
-----
```

Ceci provient du fait qu'au lancement du service NT, et grâce au paramétrage effectué, notre moteur essaye de charger la gestion des mails via SMTP (Database Mail). Le souci est que ces fichiers ne sont pas fournis avec la version Express Edition. Ils sont 3 (un exécutable et 2 DLL) :

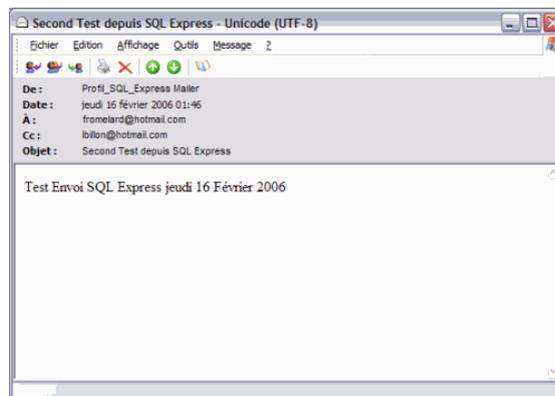
- DatabaseMail90.exe
- DatabaseMailengine.dll
- DatabaseMailprotocols.dll

Ces fichiers doivent être placés dans le même répertoire que "sqlservr.exe".

Une fois ceci fait, il faut relancer le service NT et on peut alors tester l'envoi de mail avec le script suivant :

```
EXEC msdb.dbo.sp_send_dbmail  
@profile_name='Profil_SQL_Express_Demo',  
@recipients='fromelard@hotmail.com',  
@subject = 'Second Test depuis SQL Express',  
@body = 'Test Envoi SQL Express jeudi 16 Février 2006',  
@body_format = 'HTML' ;
```

Ce qui nous donne alors le résultat suivant.



Conclusion

La configuration de cette option sur un moteur SQL Server 2005 permet de passer outre la précédente exigence qui était d'installer le client Outlook. L'usage des envois via SMTP augmente la simplicité d'installation et de gestion pour les administrateurs système ou les DBA.

Vous pouvez trouver de plus amples informations sur les liens ci-dessous :

- [Documentation en ligne de SQL Server 2005 \(FR\)](#)
 - [Intro to Database Mail in SQL 2005\(US\)](#)
 - [How to configure SQL Server Agent to send job status notifications and alert notifications in SQL Server 2005 64-bit editions \(US\)](#)
 - [Daily Dose of Yukon - SQL Server 2005 Tips \(US\)](#)
-

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F____)
Consultant Technique **I L E M**