

Microsoft OSQL

OSQL ou l'outil de base pour gérer SQL Server



Suite à mon précédent article concernant MSDE, je me suis rendu compte à partir des commentaires que de nombreux utilisateurs avaient des problèmes dans l'installation des outils graphiques. Or, par défaut, MSDE est pourvu d'un outil de gestion et d'administration très puissant : OSQL.

Introduction

Ainsi Microsoft fournit avec MSDE, l'outil de base permettant d'administrer et de gérer complètement le moteur SQL et les bases de données présentes.

Nous allons donc voir comment appréhender cet outil et mieux s'en servir. Pour cela, nous verrons dans un premier temps les options de cet outil puis différents exemples d'utilisation de celui-ci.

Présentation

OSQL est un outil utilisable en ligne de commande (WIN32), ce qui permet de créer facilement des batches l'utilisant. Il résulte de l'évolution de ISQL avec le support ODBC et le support des champs ntext, de plus OSQL ne tronque pas les champs de type texte à 255 caractères comme ISQL le fait.

Cet outil permet de se connecter à un serveur SQL Server 2000 et d'exécuter du code au format du langage TSQL. Ainsi on peut lancer des commandes de sélection, d'exécution, de mise à jour ou de suppression à partir de cet outil.

Nous pouvons donc aisément gérer un serveur SQL Server 2000 et donc MSDE. Il est fourni avec MSDE en standard afin de pallier dans une certaine mesure au manque d'outil graphique.

Normalement, cet outil se trouve dans le répertoire, sur une installation standard de machine :

- C:\Program Files\Microsoft SQL Server\80\Tools\Binn

Quoi qu'il en soit, le répertoire où se situe l'outil est ajouté dans les PATH système de la machine ce qui permet de taper directement "osql" depuis une ligne de commande quelque soit le répertoire où l'on se trouve.

Dans nos exemples qui suivent nous allons nous créer un répertoire où tous les scripts SQL seront stockés.

- C:\OSQLExemples\

Les paramètres

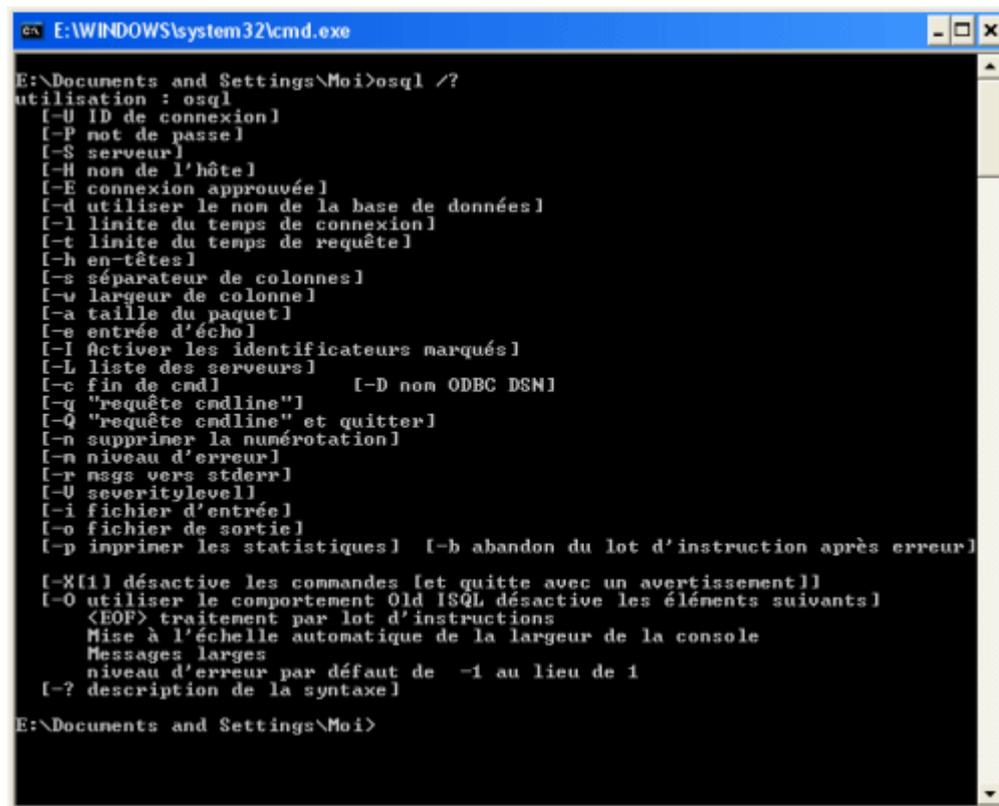
Afin de pouvoir utiliser cet outil, il faut déjà connaître tous les paramètres. Les principaux paramètres sont les suivants :

- **-U ou /U** : Spécification du nom de l'utilisateur ou login
- **-P ou /P** : Mot de passe associé à ce login
- **-S ou /S** : Serveur SQL sous le format NomDuServeur\NomInstance
- **-E ou /E** : Utilisation du mode interactif, OSQL utilisera les paramètres de connexion de l'utilisateur courant (Compte NT et mot de passe)
- **-d ou /d** : Spécification de la base de données courante
- **-i ou /i** : Fichier d'entrée (Requête SQL qui doit se finir par GO)
- **-o ou /o** : Fichier de sortie des données
- **-? ou /?** : Affichage de l'aide d'OSQL

De nombreuses autres options sont disponibles comme celles-ci :

- **-D ou /D** : Utilisation d'un DSN existant déjà sur la machine pour une connexion ODBC
- **-L ou /L** : Découverte de la liste des serveurs SQL Server disponible sur le réseau local
- **-p ou /p** : Affichage des statistiques d'exécution après chaque requête effectuée
- **-q ou /q** : Transmission directe de la requête SQL sans se déconnecter du serveur SQL
- **-Q ou /Q** : Transmission directe de la requête SQL en se déconnectant du serveur SQL
- ...

Si vous souhaitez toutes les connaître, je vous invite à utiliser la commande "osql /?", elle vous ressortira la liste de toutes les options disponibles.



```
E:\WINDOWS\system32\cmd.exe
E:\Documents and Settings\Moi>osql /?
utilisation : osql
[-U ID de connexion]
[-P mot de passe]
[-S serveur]
[-H non de l'hôte]
[-E connexion approuvée]
[-d utiliser le nom de la base de données]
[-l limite du temps de connexion]
[-t limite du temps de requête]
[-h en-têtes]
[-s séparateur de colonnes]
[-v largeur de colonne]
[-a taille du paquet]
[-e entrée d'écho]
[-I Activer les identificateurs marqués]
[-L liste des serveurs]
[-c fin de cmd]          [-D nom ODBC DSN]
[-q "requête cmdline"]
[-Q "requête cmdline" et quitter]
[-n supprimer la numérotation]
[-n niveau d'erreur]
[-r msgs vers stderr]
[-U severitylevel]
[-i fichier d'entrée]
[-o fichier de sortie]
[-p imprimer les statistiques] [-b abandon du lot d'instruction après erreur]

[-X[1] désactive les commandes [et quitte avec un avertissement]]
[-O utiliser le comportement Old ISQL désactive les éléments suivants]
  <EOF> traitement par lot d'instructions
  Mise à l'échelle automatique de la largeur de la console
  Messages larges
  niveau d'erreur par défaut de -1 au lieu de 1
[-? description de la syntaxe]

E:\Documents and Settings\Moi>
```

Les Commandes

Une fois connecté sur le serveur SQL, OSQL possède quelques commandes qui lui sont propres et qui permettent de travailler avec cet outil.

- **GO** : Lance l'exécution du script TSQL tapé
- **EXIT ou QUIT** : Quitte OSQL
- **RESET** : Supprime toutes les commandes tapées dans OSQL (le numéro de la ligne repasse alors à 1)
- **ED** : Lance l'éditeur de texte
- **CTRL+C** : Stoppe l'exécution en cours sans quitter OSQL
- **!! command** : Lance la commande MSDOS spécifiée

La commande ED lance l'éditeur de texte par défaut de la machine. Celui-ci est de base EDIT.COM, éditeur de texte en version MS-DOS. Pour changer cela il faut exécuter la commande "SET EDITOR=notepad" par exemple (pour avoir Notepad de base) dans une console MS-DOS.

Maintenant que nous avons listé les paramètres et les commandes utilisables, voyons comment nous servir de cet outil en utilisation basique de listing, puis en avancé avec l'exécution de scripts complexes pour une création de base.

Utilisation Basique

Maintenant voyons comment faire pour faire un listing simple sur la base que nous avons tous après l'installation de MSDE : la base MASTER.

ATTENTION :

Cet exemple est juste à titre d'information, il est fortement déconseillé de modifier des informations dans cette base sous peine de ne plus pouvoir faire fonctionner votre moteur SQL.

Tout d'abord, il faut ouvrir une fenêtre de commande MS-DOS :

- Démarrer > Exécuter, puis vous tapez : "command"

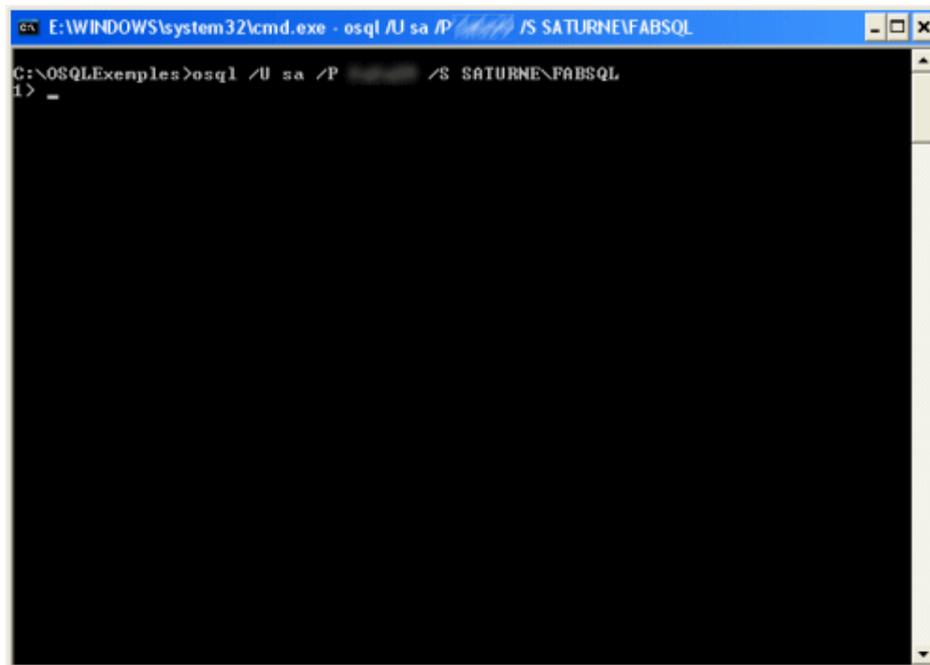
Vous obtenez alors l'invite MS-DOS placé par défaut dans votre répertoire courant. Nous nous placerons dans le répertoire que nous avons créé pour cet article, ceci grâce à la commande suivante :

- CD C:\OSQLExemples\

Une fois dans ce répertoire nous allons lancer OSQL en nous connectant sur notre MSDE que nous avons installé dans le précédent article (Instance FABSQLE et notre machine se nomme SATURNE). Ainsi la commande complète sera la suivante :

- >osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQLE

Cela nous permet de nous connecter sur notre instance MSDE avec le login sa dont nous avons donné le mot de passe "MotDePasseRenforcé". Nous obtenons alors l'écran suivant :



```
cmd E:\WINDOWS\system32\cmd.exe - osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQLE
C:\OSQLExemples>osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQLE
1> _
```

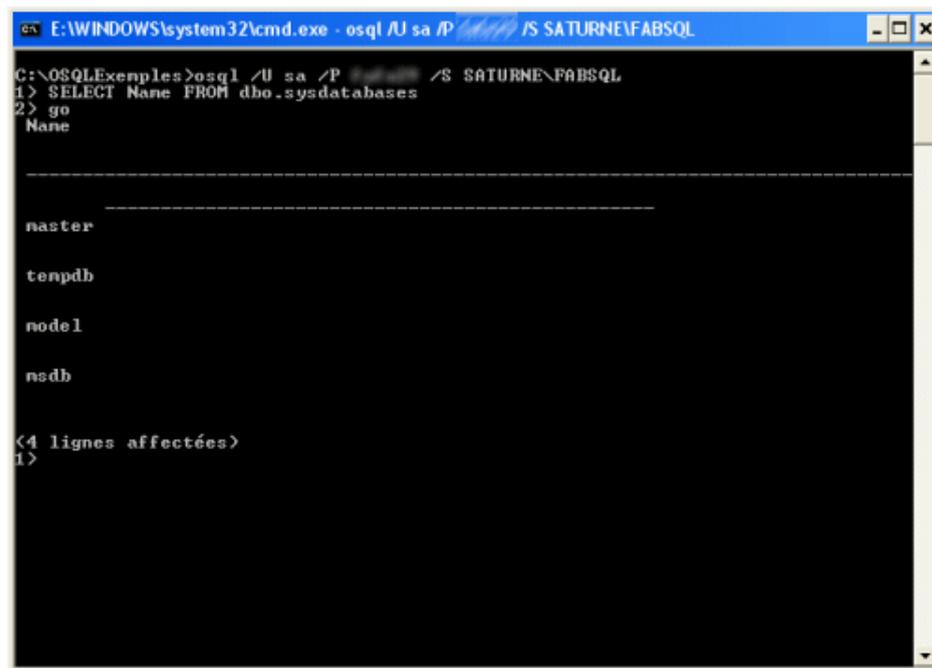
Nous avons alors une ligne numérotée à 1 avec un curseur qui apparaît. Cela nous indique que nous sommes bien connectés sur le serveur SQL et que celui-ci attend une requête de notre part. Par défaut, quand la base n'est pas spécifiée (avec le paramètre -d), nous sommes sur la base MASTER.

Notre requête d'exemple va simplement lister les bases (par leur nom) présentes dans notre instance actuelle.

Pour exécuter une requête, il faut taper le code TSQL et ENTREE puis taper "GO". Pour notre exemple, cela donnera donc :

- SELECT Name FROM dbo.sysdatabases
- GO

Une fois le GO tapé, puis validé par ENTREE, nous voyons apparaître le résultat dans notre invite MS -DOS.



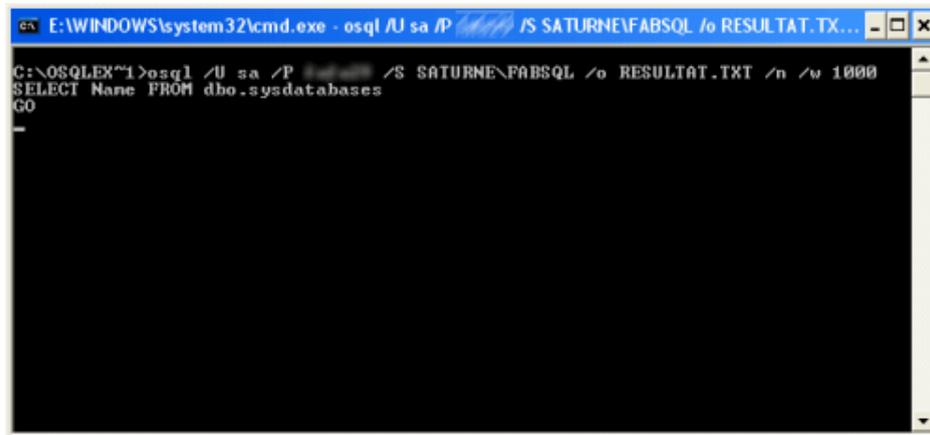
```
E:\WINDOWS\system32\cmd.exe - osql /U sa /P /S SATURNE\FABSQL
C:\OSQLExemples>osql /U sa /P /S SATURNE\FABSQL
1> SELECT Name FROM dbo.sysdatabases
2> go
Name
-----
master
tempdb
model
msdb

<4 lignes affectées>
1>
```

Ainsi, on a le résultat directement à l'écran. Mais nous voyons vite que ceci n'est pas pratique, du fait de la limitation de la largeur de la fenêtre MS -DOS. Donc, nous allons chercher à envoyer ce résultat dans un fichier de sortie (Paramètre -o).

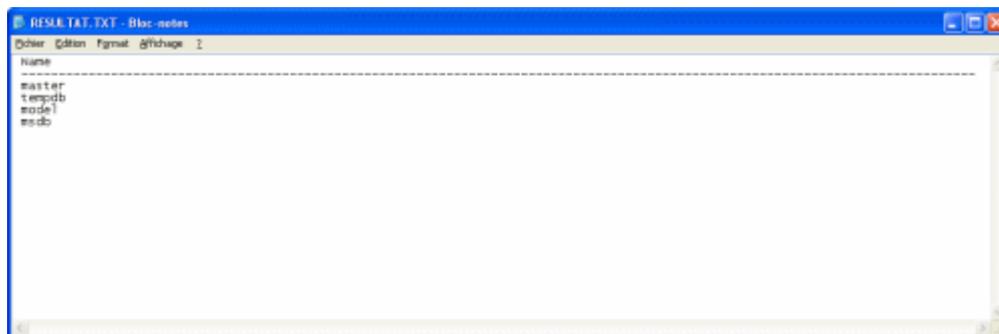
Il nous faut donc nous déconnecter, en utilisant la commande OSQL "EXIT" et nous reconnecter en ajoutant le paramètre du fichier de sortie. Il faut aussi ajoute le -n qui évite d'avoir les numéros de ligne dans le fichier de sortie. L'option -w suivie d'un entier permet de fixer le nombre de caractères de la colonne.

- `osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQ /o RESULTAT.TXT /n /w 1000`



```
E:\WINDOWS\system32\cmd.exe - osql /U sa /P [redacted] /S SATURNE\FABSQ /o RESULTAT.TXT /n /w 1000
C:\OSQLE\EX~1>osql /U sa /P [redacted] /S SATURNE\FABSQ /o RESULTAT.TXT /n /w 1000
SELECT Name FROM dbo.sysdatabases
GO
```

Le fichier de sortie que nous avons spécifié (RESULTAT.TXT) se trouve dans le répertoire courant et contient le résultat de la requête.



```
RESULTAT.TXT - Bloc-notes
-----
Name
-----
master
tempdb
model
msdb
```

Tout cela est bien intéressant mais quelque peu limité. Voyons maintenant comment utiliser OSQL pour exécuter des scripts plus complexes.

Exécution d'un fichier SQL

Maintenant que nous avons vu l'utilisation un peu basique de cet outil, nous allons pousser un peu plus les possibilités de celui-ci.

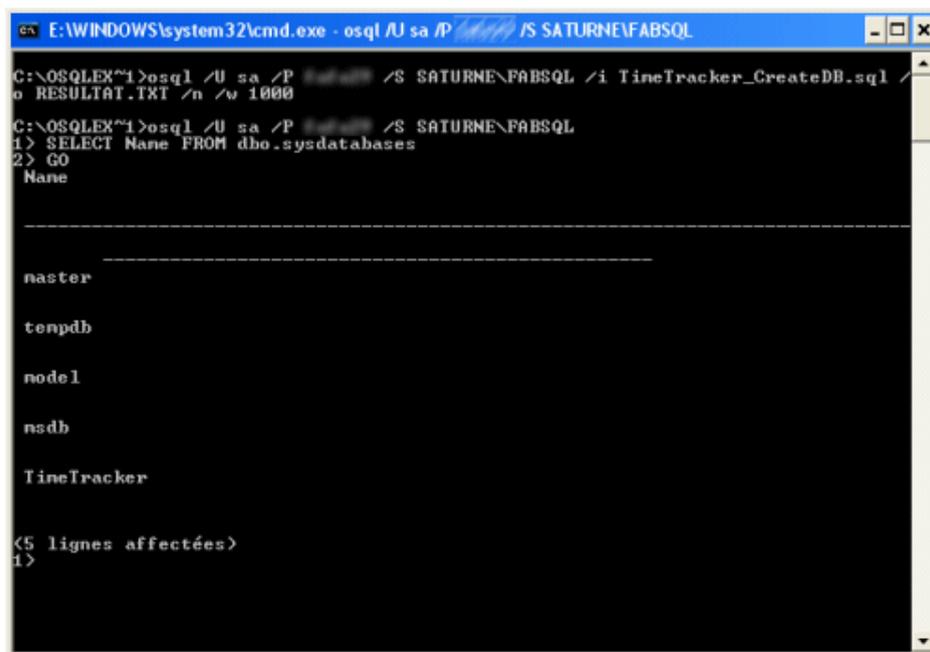
Exécution d'un script SQL enregistré dans un fichier Texte

Dans cette partie nous allons prendre un script TSQL de création d'une base qui nous est fourni lors de l'installation d'un des Starter Kits : **TimeTracker (US)**. Il s'agit d'un script SQL de création de Base (TimeTracker_CreateDB.sql) qui est fourni lors de l'installation du Kit.

Ainsi maintenant pour exécuter ce script SQL, nous allons modifier notre commande précédente en incluant l'option -i avec le nom du fichier. Ce qui nous donne :

- `osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQ /i TimeTracker_CreateDB.sql /o RESULTAT.TXT /n /w 1000`

OSQL va alors fonctionner durant un moment jusqu'à rendre la main. Nous allons pouvoir vérifier le résultat dans le fichier de sortie et le confirmer par la première requête que nous avons exécutée.



```
ca E:\WINDOWS\system32\cmd.exe - osql /U sa /P [redacted] /S SATURNE\FABSQ
C:\OSQLEX~1>osql /U sa /P [redacted] /S SATURNE\FABSQ /i TimeTracker_CreateDB.sql /o
o RESULTAT.TXT /n /w 1000
C:\OSQLEX~1>osql /U sa /P [redacted] /S SATURNE\FABSQ
1> SELECT Name FROM dbo.sysdatabases
2> GO
Name
-----
master
tempdb
model
msdb
TimeTracker
<5 lignes affectées>
1>
```

Ainsi nous voyons que nous pouvons aisément sans avoir besoin d'un client graphique ou WEB effectuer la gestion des bases de données (on s'en doutait un peu, mais il est bon de le prouver).

Les suppressions et mises à jour fonctionnant sur le même principe, nous n'insisterons pas sur le sujet.

Utilisation Avancée

Exécution de taches de gestion du serveur

Nous savons donc que les scripts de gestion des bases et des données sont très facilement exécutables depuis OSQL. Mai nous pouvons aussi effectuer la gestion du moteur SQL Server lui même à partir de script système.

Deux exemples sont particulièrement adaptés pour cela.

- **Sauvegarde des Bases**

N'ayant pas la console graphique de Gestion (SQL Enterprise Manager), il n'est pas facile de faire la sauvegarde des bases telle qu'on peut l'effectuer grâce à un plus de maintenance.

Nous avons le script disponible ici :

- **SQL SERVER - PROCÉDURE STOCKÉE FAISANT UN BACKUP DE TOUTES LES BASES ET JOURNAUX DE TRANSACTION AUTOMATIQUEMENT (FR)**

- **Mise à jour des Statistiques des bases de données**

La mise à jour des statistiques des bases, ceci pour que le moteur SQL Server choisisse toujours le meilleur plan d'exécution et permette que la requête lancée soit rapide, se fait à partir d'une procédure stockée système.

Nous avons le script disponible ici :

- **SQL SERVER - PROCÉDURE STOCKÉE METTANT À JOUR TOUTES LES STATISTIQUES DE TOUTES LES BASES DU SERVEUR (FR)**

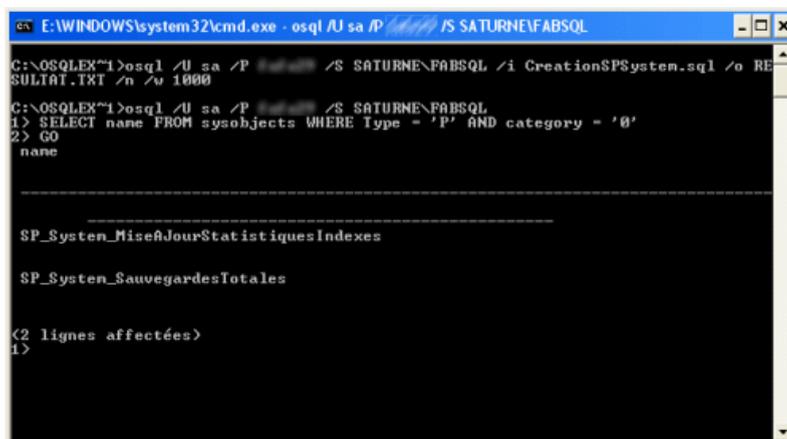
A partir de ces deux scripts nous allons les ajouter en procédures stockées à la base de données du système : MASTER.

Pour cela nous ajoutons ces deux scripts de création de procédure stockée dans un fichier texte : CreationSPSystem.sql.

Une fois que ce fichier, il ne nous reste plus qu'à exécuter OSQL afin de lancer ce script SQL avec la commande suivante :

- `osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQ /i CreationSPSystem.sql /o RESULTAT.TXT /n /w 1000`

On peut ensuite contrôler que ces 2 procédures stockées ont bien été créées.



```
ex E:\WINDOWS\system32\cmd.exe - osql /U sa /P // // /S SATURNE\FABSQ
C:\OSQLE\1>osql /U sa /P // // /S SATURNE\FABSQ /i CreationSPSystem.sql /o RE
SULTAT.TXT /n /w 1000
C:\OSQLE\1>osql /U sa /P // // /S SATURNE\FABSQ
1> SELECT name FROM sysobjects WHERE Type = 'P' AND category = '0'
2> GO
name
-----
SP_System_MiseAJourStatistiquesIndexes
SP_System_SauvegardesTotales
(2 lignes affectées)
1>
```

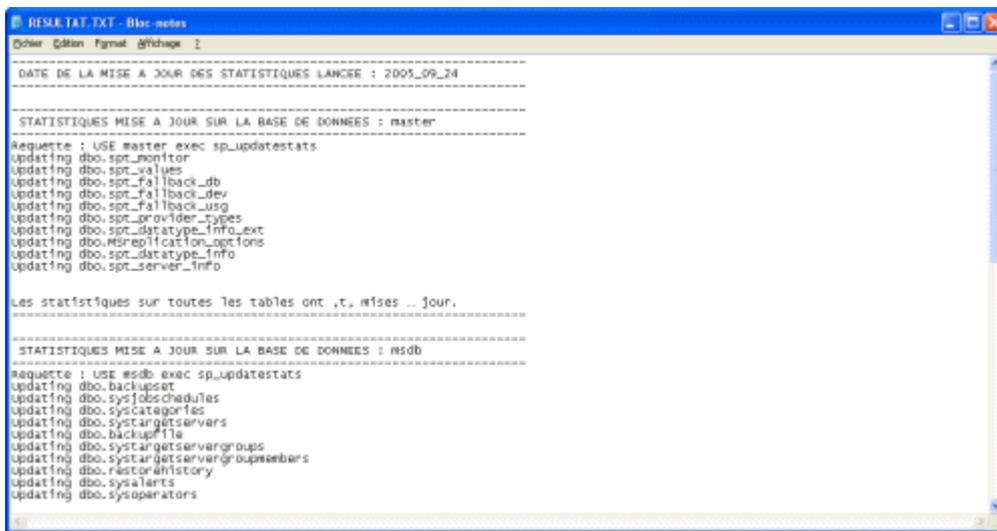
Notre objectif maintenant est de faire un script permettant de remplacer le principe du plan de maintenance géré par SQL Server. Pour cela, nous utiliserons le planificateur de tâches de Windows, en préparant deux scripts BATCH, un par procédure stockée.

Mise à jour des statistiques du serveur

Voyons tout d'abord celui pour les statistiques. Pour cette procédure stockée, nous n'avons pas besoin de transmettre de paramètre et donc nous pouvons lancer OSQL avec l'option -Q (en fournissant le script d'exécution TSQL de la procédure stockée) comme suit :

- `osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQ /Q "exec dbo.SP_System_MiseAJourStatistiquesIndexes" /o RESULTAT.TXT /n /w 1000`

Nous avons alors dans notre répertoire courant le fichier de résultat.



```
RESULTAT.TXT - Bloc-notes
-----
DATE DE LA MISE A JOUR DES STATISTIQUES LANCEE : 2005_09_24
-----
STATISTIQUES MISE A JOUR SUR LA BASE DE DONNEES : master
-----
Requete : USE master exec sp_updatestats
Updating dbo.spt_monitor
Updating dbo.spt_values
Updating dbo.spt_fallback_db
Updating dbo.spt_fallback_dev
Updating dbo.spt_fallback_usg
Updating dbo.spt_provider_types
Updating dbo.spt_datatype_info_ext
Updating dbo.msreplication_options
Updating dbo.spt_datatype_info
Updating dbo.spt_server_info

Les statistiques sur toutes les tables ont été mises à jour.
-----
STATISTIQUES MISE A JOUR SUR LA BASE DE DONNEES : msdb
-----
Requete : USE msdb exec sp_updatestats
Updating dbo.backupset
Updating dbo.sysjobschedules
Updating dbo.syscategories
Updating dbo.systemtargetservers
Updating dbo.backupfile
Updating dbo.systemtargetservergroups
Updating dbo.systemtargetservergroupmembers
Updating dbo.restorehistory
Updating dbo.sysalerts
Updating dbo.sysoperators
```

Il nous suffit donc de créer un fichier MAJStats.BAT dans lequel nous placerons cette commande, et planifier l'exécution de ce fichier toutes les semaines.

Sauvegarde des bases de données

Maintenant, voyons la procédure de sauvegarde. Cette procédure stockée demande des paramètres en entrée. Cela nécessite donc pour mieux comprendre (on pourrait aussi le faire en direct comme pour les statistiques) de passer par un fichier SQL.

Notre fichier contiendra simplement la commande adaptée afin de sauvegarder l'ensemble des bases et des logs de transaction dans le répertoire où nous sommes. Ce qui donne la commande suivante :

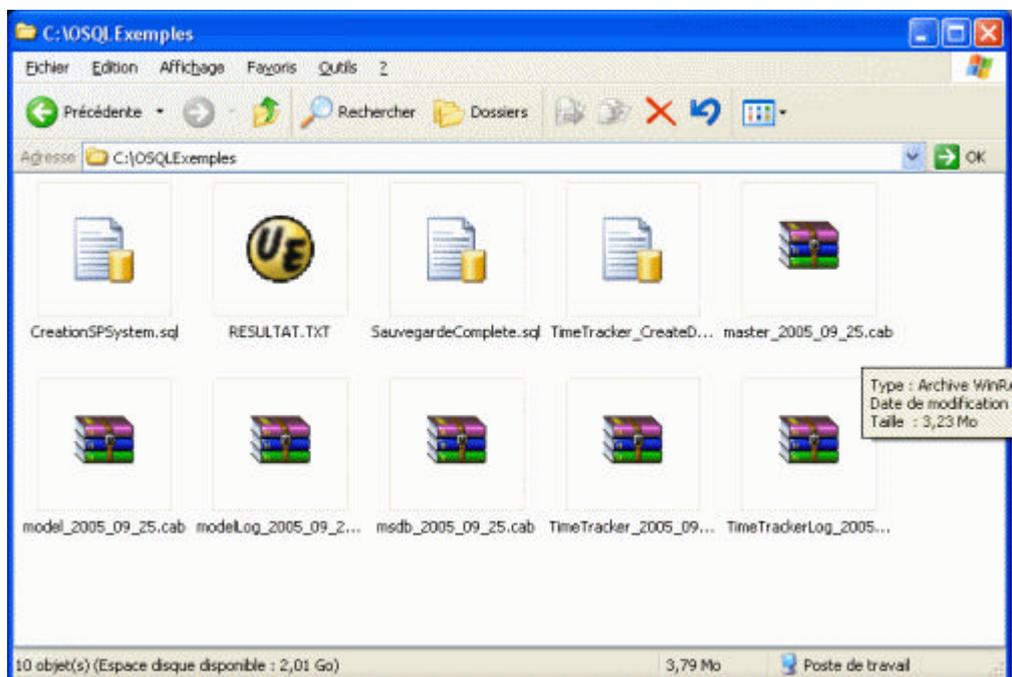
- EXEC dbo.SP_System_SauvegardesTotales 1, 1, 'C:\OSQLExemples\'

Nous nommerons ce fichier 'SauvegardeComplete.sql', ainsi dans le cas où nous voudrions changer les paramètres pour ne finalement sauvegarder que les bases, il nous suffit de modifier le paramètre dans le fichier SQL.

La commande OSQL à lancer sera la suivante :

- osql /U sa /P MotDePasseRenforcé /S SATURNE\FABSQ /i SauvegardeComplete.sql /o RESULTAT.TXT /n /w 1000

Il nous suffit alors de vérifier dans le répertoire spécifié si les fichiers de sauvegarde sont bien présents, puis de planifier cette sauvegarde de façon journalière à une heure où le serveur est le moins sollicité .



Conclusion

Après cette présentation, on comprend mieux pourquoi Microsoft a fourni ce simple outil avec le moteur gratuit MSDE. En effet, dès que l'on connaît suffisamment le langage TSQL, cet outil suffit pour de nombreuses tâches d'administration et de gestion. Il est beaucoup moins pratique qu'un outil graphique, mais dans des situations d'installation ou de mises à jour rapides, cela suffit largement.

J'espère que cet article vous aura servi. Voici quelques articles ou exemples sur ce sujet :

- [Quick tip to discover SQL Servers \(US\)](#)
- [osql Utility - MSDN \(US\)](#)
- [Using the osql Utility - MSDN \(US\)](#)
- [Microsoft DataBase Engine \(MSDE\) \(FR\)](#)
- [Installer et Utiliser MSDE 2000 \(FR\)](#)

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F____)