

Déplacement, sauvegardes et restauration de bases sous MS-SQL Server

par [Fabien Celaia](#)

Date de publication : 20.05.2006

Dernière mise à jour : 15.10.2006

Diverses méthodes pour déplacer, copier, sauvegarder, restaurer une base de données MS-SQL Server

- I - Base de données et fichiers
- II - Sauvegardes basiques via les assistants
- III - La sauvegarde physique
 - III-A - Avant SQL Server 2000
 - III-B - Dès SQL Server 2000
- IV - La sauvegarde logique
 - IV-A - Les bases et les journaux
 - IV-B - Sauvegardes et restaurations complètes
 - IV-C - Sauvegardes et restaurations incrémentielles
 - IV-D - Restauration de la base master
 - IV-E - Sauvegarde, puis restauration sur une instance différente
- V - Déplacement de bases
 - V-A - Restaurer physiquement une base sous un autre nom.
 - V-B - Restaurer logiquement une base sous un autre nom
 - V-C - Déplacer une base système
 - V-C-1 - Déplacer tempdb ou model
 - V-C-1-a - Jusqu'en version 2000
 - V-C-1-b - Depuis la version 2005
 - V-C-2 - Déplacer msdb
 - V-C-3 - Déplacer master

I - Base de données et fichiers

Une base de données MS-SQL comprend entre autre un espace logique où l'on peut créer des objets tables, vues, procédures stockées, etc.

Physiquement, tout se retrouve stocké sur disque(s) dans des fichiers.

SQL Server gère donc, comme la plupart des autres SGBDR du marché, différents types de fichiers

Fichiers	Emplacement par défaut	Création
binaires et de configuration	C:\Program Files\Microsoft SQL Server\90	Lors de l'installation
fichiers de données pour les bases système	C:\Program Files\Microsoft SQL Server\Data	Lors de l'installation
fichiers de données pour les bases utilisateurs	C:\Program Files\Microsoft SQL Server\Data	Lors de la création de la base ou de l'ajout d'espace pour cette base

Concentrons-nous sur les bases de données utilisateurs.

Dans sa mission initiale, la sauvegarde/restauration sert à mettre en sécurité des données sur un support amovible, et - au besoin - à remonter ces données sur leur emplacement d'origine.

Très vite, les sauvegardes ont été utilisées pour remonter les bases sur d'autres instances, avant tout à des fins de test, car on n'est sûr qu'une sauvegarde est bonne que lorsqu'on l'a remontée... ce qui n'est pas toujours le cas en environnement de production.

Suivant cette même logique, il est possible de les utiliser afin de cloner/renommer une base (quoique le `sp_renamedb` est quand même plus efficace si l'on reste sur une même instance et que l'on veut simplement renommer la base).

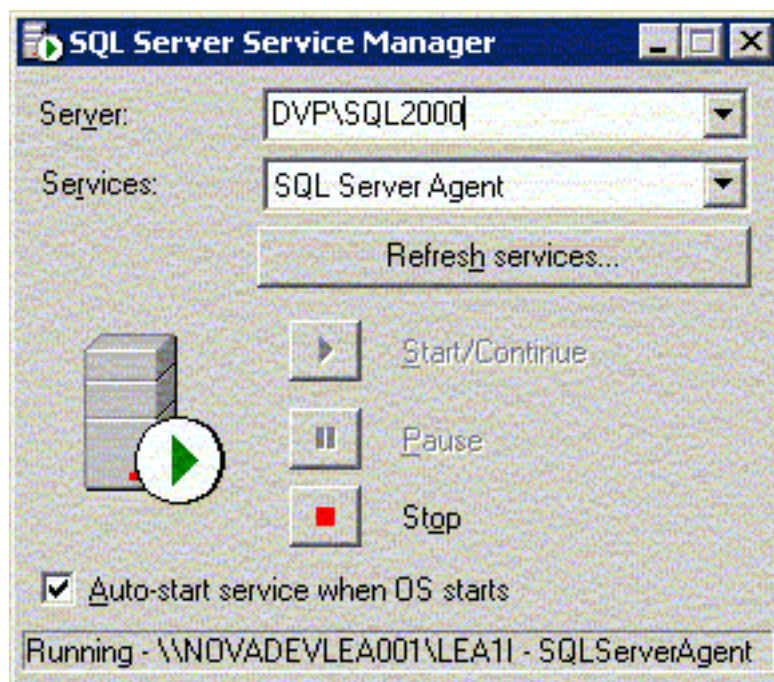
Depuis la version 2000 de MS-SQL Server, chaque base de donnée utilisateur gère les fichiers qui lui sont associés, et qui sont au minimum au nombre de 2

- un pour le segment de données et le segment système (soit pour stocker les tables)
- un pour stocker le journal de transactions

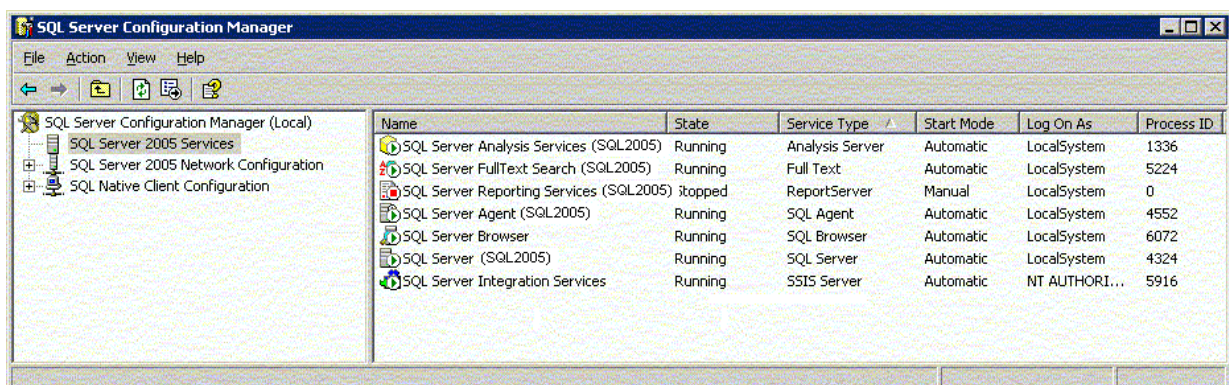
II - Sauvegardes basiques via les assistants

MS-SQL Server inclut un agent et un ordonnanceur permettant à un débutant de déterminer aisément une stratégie de sauvegarde.

Dans un premier temps, il faut s'assurer que l'agent est bien démarré. C'est aisément faisable via SQL Server Service Manager (jusqu'à la version 2000), SQL Server Configuration Manager (dès la version 2005) ou en allant directement dans les Services Windows.



SQL Server Service Manager jusqu'à la version 2000



SQL Server Configuration Manager dès la version 2005

Ces actions sont toujours effectuables en ligne de commande via NET

```
net start SQLAgent$SQL2005
```

Ensuite, les assistants se chargent de vous guider dans la création des travaux de sauvegarde et leur planification.

- Se connecter à l'instance
- Cliquer sur Maintenance
- Clic droit sur Database Maintenance Plans
- Choisir un nouveau plan de maintenance
- Suivre l'assistant

Les version 2000 et 2005 diffèrent et cela que l'assistant de la 2005 vous plonge réellement dans un environnement de type "workflow" où les diverses tâches sont connectables à souhait. Grandement complexifié pour une tâche basique, il apporte sans conteste un plus dans les traitements plus complexes.

Il va de soi que toutes ces commandes peuvent être lancées à la main via script T-SQL. Utilisez l'assistant en premier et allez ensuite visualiser les scripts générés afin de les remettre "à votre sauce".

III - La sauvegarde physique

Il s'agit donc de sauvegarder indirectement la base de données en sauvant en fait les fichiers physiques qui la composent.

C'est une sauvegarde "système" faite par copie de fichiers depuis le système d'exploitation.

II-A - Avant SQL Server 2000

- Les bases de données n'ont pas de fichiers associés spécifiquement.
- Des fichiers (devices) sont mis à disposition du serveur.
- Les bases se partagent des espaces sur ces devices.
- Une base peut donc utiliser plusieurs parts de devices.
- Un device peut être utilisé par plusieurs bases.

Table	Description	Requête associée
master..sysdatabases	Liste des bases de données	select id, name from master..sysdatabases
master..sysdevices	Liste des fichiers physiques disponibles	select name from master..sysdevices
master..sysusages	Table de lien pour l'allocation des bases sur les fichiers	USE master GO SELECT d.name, v.physname FROM sysdevices v, sysusages u, sysdatabases b WHERE u.vstart BETWEEN v.low AND v.high AND u.dbid=b.dbid
Base..syssegments	Table des segments de la base devant mixer avec sysusages.segmaps (masque de bits)	

Une base ne pouvant être associée à un fichier spécifique, seule la sauvegarde de l'instance entière est possible, et ceci uniquement lorsque l'instance est arrêtée (shutdown). Tout autre essai vous expose à corrompre votre base, une désynchronisation mémoire/disque étant quasi sûre. La liste des fichiers à sauver peut s'afficher grâce au T-SQL suivant (la condition sur *status* supprime les *dump devices*):

```
SELECT physname
FROM master..sysdevices
WHERE status & 16=0
```

II-B - Dès SQL Server 2000

- Les bases de données ont des fichiers associés spécifiquement.
- Des fichiers et donc leur base associée sont plus aisément détachables.
- Les bases ne se partagent pas des espaces sur ces devices.
- Une base peut donc utiliser plusieurs fichiers.
- Un device ne peut donc être utilisé qu'à une seule et unique base.

Il devient donc possible de détacher et d'attacher une base de données en se fixant sur ses fichiers.

Afficher les fichiers d'une base

```
USE maBase
GO
SELECT filename
FROM sysfiles
GO
```

Détacher une base

```
USE maBase
GO
EXEC sp_detach_db 'MaBase'
GO
```

Dès ce moment, la base n'est plus attachée à serveur SQL et n'est donc plus utilisable. Ses fichiers peuvent par contre être copiés, au besoin sur un autre serveur ayant une autre instance SQL Server.

Rattacher une base

```
USE maBase
GO
EXEC sp_attach_db @dbname = N'MaBase',
  @filename1 = N'D:\MSSQL\Data\MaBase_Data.mdf',
  @filename2 = N'D:\MSSQL\Log\MaBase_log.ldf' ;
GO
```

IV - La sauvegarde logique

L'idée est donc de sauvegarder la base, indépendamment des fichiers physiques qui lui sont rattachés. L'avantage est de ne sauvegarder que les pages allouées de la base.

III-A - Les bases et les journaux

3 segments principaux sont créés dans une base de données.

- le segment de données : il va contenir tous les objets utilisateurs
- le segment de log : il va contenir la table des transactions (syslogs); dès v.2000, il est sur un fichier dissocié spécifique
- le segment système : il se partage généralement le même fichier physique que le segment de données et contient les tables ou objets systèmes.

Pour des raisons de performances, on peut se créer d'autres segments particuliers (indexes, tables spécifiques, ...) Jusqu'en pré-v.2000, il était fortement recommandé de dissocier le segment du journal de transactions des autres segments, ceci afin de simplifier la récupération lors de corruptions, mais surtout d'autoriser la sauvegarde du journal, que l'on peut associer à une sauvegarde incrémentielle. Depuis la version 2000, ceci n'est plus nécessaire puisque le journal se trouve sur un fichier dédié (Nomdebase.ldb).

III-B - Sauvegardes et restaurations complètes

Une base de données se découpe physiquement en blocs que l'on appelle "pages". Une base peut donc être beaucoup plus grande que ce qu'elle contient. Elle contient alors beaucoup de pages non encore allouées à un objet. La sauvegarde complète va donc être la sauvegarde de toutes les pages allouées d'une base dans un fichier externe. Ceci se fait à l'aide de la commande BACKUP.

```
BACKUP DATABASE MaBase TO DISK=N'D:\BACKUP\MaBase.bak'
BACKUP TRAN MaBase TO DISK=N'D:\BACKUP\MaBase_log.bak'
GO
```

On peut travailler avec des dumpdevices (cibles de fichiers prédéfinies), directement sur bande, etc mais ceci n'est pas le sujet de ce didacticiel.

Relevons qu'il est tout à fait possible d'exécuter ce type de sauvegarde à chaud = durant l'utilisation courante de la base. Seules les performances seront sensiblement dégradées.

La récupération se fait avec la commande RESTORE, en ayant pris soins au préalable de supprimer toute connexion sur la base à remonter. Contrairement à la sauvegarde, la récupération requiert - on peut s'en douter - un accès exclusif à la base.

```
USE master
GO
SELECT 'kill',spid
FROM sysprocesses
WHERE dbid=db_id('MaBase')
GO
/* Exécution de l'output de la commande ci-dessus afin de libérer la base */
GO
RESTORE DATABASE MaBase FROM DISK=N'D:\BACKUP\MaBase.bak'
GO
```

III-C - Sauvegardes et restaurations incrémentielles

Selon l'importance de la base, il est parfois préférable de faire, entre les divers BACKUP DATABASE, des sauvegardes incrémentielles des journaux de transactions, ceci afin de pouvoir retrouver au besoin non pas l'état de la base depuis la dernière sauvegarde complète (souvent la veille au soir), mais celui depuis la dernière sauvegarde du journal de transaction (par exemple chaque heure).

```
BACKUP TRAN MaBase TO DISK='D:\BACKUP\MaBase_MonHeureMaDate.trn'
GO
```

Soit donc le programme de sauvegardes suivant:

- Sauvegarde complète tous les soir à 20h
- Sauvegarde des journaux de transactions toutes les heures ouvrables (7-19h)

Imaginons maintenant qu'un crash apparaisse à 11h20 du matin. Que devez-vous faire ?

- Tenter d'effectuer une dernière sauvegarde du journal de transactions
- Remonter le serveur SQL dans un état stable (ou passer sous un autre SQL Serveur)
- Restaurer la base depuis la sauvegarde complète la plus récente
- Restaurer tous les journaux de transactions à disposition.

Soit, en T-SQL

```
USE master
GO
/* Tentative de sauvegarde du dernier journal */
BACKUP TRAN MaBase To DISK=N'D:\Backup\MaBase_crash11h30_20060518.Bak'
SELECT @@error
GO

/* Si même arborescence disque */
RESTORE DATABASE MaBase FROM DISK=N'D:\Backup\MaBase_20h_20060517.Bak' WITH NORECOVERY
GO

/* Si arborescence disque différente, repositionnement sur la nouvelle arborescence */
RESTORE DATABASE MaBase FROM DISK=N'D:\Backup\MaBase_20h_20060517.Bak'
WITH MOVE 'MaBase_Data' TO DISK='C:\MaNouvelleLocalisation\Data\MaBase.mdb' ,
MOVE 'MaBase_Log' TO DISK='C:\MaNouvelleLocalisation\Log\MaBase.ldf' ,
NORECOVERY
GO

/* ... et récupération des journaux de transactions successifs, jsute avant le crash */
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_7h_20060518.Bak' WITH NO_RECOVERY
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_8h_20060518.Bak' WITH NO_RECOVERY
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_9h_20060518.Bak' WITH NO_RECOVERY
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_10h_20060518.Bak' WITH NO_RECOVERY
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_11h_20060518.Bak' WITH NO_RECOVERY
RESTORE TRAN MaBase FROM DISK=N'D:\Backup\MaBase_crash11h30_20060518.Bak' WITH RECOVERY STOPAT =
'May 18, 2006 11:19 AM'
GO
```

III-D - Restauration de la base master

La récupération de la base master pose un problème particuliers compte tenu qu'il faut obtenir un accès exclusif à une base pour la restaurer, et que la base master contient des processus internes de SQL Server.

Il va donc falloir arrêter le serveur et le remonter dans un mode bien particuliers, avec une base master en mode *accès exclusif*.

```
net stop MSSQL$SQL2005
sqlservr.exe - m -s SQL2005
```

Connectez-vous ensuite via votre outil préféré (isql, sqlcmd ou un équivalent graphique) et exécutez l'opération de restauration

```
RESTORE DATABASE master FROM DISK='D:\BACKUP\master.bak'
GO
```

N'ayez pas d'inquiétude si à la fin de cette commande, vous perdez votre connexion : aussitôt la récupération effectuée, le SQL Server se suicide (shutdown) afin de pouvoir être redémarré correctement avec le nouvel état de sa base master.

```
net start MSSQL$SQL2005
```

III-E - Sauvegarde, puis restauration sur une instance différente

Précondition : quelle que soit la méthode, il n'est pas possible de remonter une version récente sur un serveur moins récent (incompatibilité descendante).

Piège à éviter : une désynchronisation des logins. Les logins dépendant de la base master et les utilisateurs étant un mappage de login propre à la base, le fait de remonter une base et ses utilisateurs sur un serveur qui n'aurait pas les bons logins risque de causer problème.

Il est donc nécessaire, dans ces cas, de sauvegarder les DDL de création des users AVANT d'écraser l'ancienne base, via des ordres T-SQL de type

```
USE MaBase
GO

/* Version pré-2005 */

select 'EXEC SP_ADDUSER ' + name + ', ' + suser_sname(sid)
from sysusers
WHERE uid between 5 and 16000
AND STATUS=0 -- Suppression des application roles

/* Version 2005 */

select 'CREATE USER ' + name + ' FOR LOGIN '+suser_sname(sid)
from sys.database_principals
WHERE principal_id between 5 and 16000
AND TYPE='S'
```

Il faudra ensuite détecter, après le chargement, les utilisateurs désynchroniser et les recréer au besoin à l'aide de la procédure `sp_change_users_login`.

V - Déplacement de bases

IV-A - Restaurer physiquement une base sous un autre nom.

```
BACKUP DATABASE MaBase TO N'D:\Backup\MaBase.BAK'
GO
RESTORE DATABASE MonNouveauNomDeBase
FROM DISK = N'D:\Backup\MaBase.BAK'
WITH MOVE N'MaBase_Data' TO N'D:\MSSQL\Data\MonNouveauNomDeBase_Data.mdf',
MOVE N'MaBase_Log' TO N'D:\MSSQL\Data\MonNouveauNomDeBase_Log.ldf'
GO

ALTER DATABASE MonNouveauNomDeBase MODIFY FILE (NAME=N'MaBase_Data',
NEWNAME=N'MonNouveauNomDeBase_Data')
ALTER DATABASE MonNouveauNomDeBase MODIFY FILE (NAME=N'MaBase_Log',
NEWNAME=N'MonNouveauNomDeBase_Log')
GO
```

IV-B - Restaurer logiquement une base sous un autre nom

On détache préalablement la base à cloner afin que SQL Server relâche son accès en lecture sur les fichiers.

```
EXEC SP_DETACH_DB 'MaBase'
GO
```

Via Windows, duplication des fichiers de la base

```
copy D:\MSSQL\Data\MaBase_Data.mdf D:\MSSQL\Data\MonNouveauNomDeBase_Data.mdf
copy D:\MSSQL\log\MaBase_Log.ldf D:\MSSQL\log\MonNouveauNomDeBase_Log.ldf
```

Puis rattachement de la base que nous venons de détacher et de son clône.

```
USE MASTER
GO
EXEC sp_attach_db @dbname = N'MaBase',
@filename1 = N'D:\MSSQL\Data\MaBase_Data.mdf',
@filename2 = N'D:\MSSQL\Log\MaBase_log.ldf'
GO
EXEC sp_attach_db @dbname = N'MonNouveauNomDeBase',
@filename1 = N'D:\MSSQL\Data\MonNouveauNomDeBase_Data.mdf',
@filename2 = N'D:\MSSQL\Log\MonNouveauNomDeBase_log.ldf'
GO
ALTER DATABASE MonNouveauNomDeBase MODIFY FILE (NAME=N'MaBase_Data',
NEWNAME=N'MonNouveauNomDeBase_Data')
ALTER DATABASE MonNouveauNomDeBase MODIFY FILE (NAME=N'MaBase_Log',
NEWNAME=N'MonNouveauNomDeBase_Log')
GO
```

IV-C - Déplacer une base système

Les bases système de MS-SQL Server sont, via les outils graphiques, quasiment inamovibles. Pour les déplacer, il va falloir user de subterfuges.

IV-C-1 - Déplacer tempdb ou model

IV-C-1-a - Juqu'en version 2000

En version pré-2000, possibilité d'influer sur la table système sysdevices.

```
USE master
GO
sp_configure 'allow updates', 1
GO
UPDATE master..sysdevices SET phyname = 'D:\MaNouvelleLocalisation\tempdb.mdf' where name =
'tempdev'
UPDATE master..sysdevices SET phyname = 'D:\MaNouvelleLocalisation\templog.ldf' where name =
'templog'
GO
SHUTDOWN with nowait
GO
```

Puis copie des fichiers et redémarrage du service MS-SQL Server (ici, non de l'instance = SQL2005)

```
move "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\tempdb.mdf"
D:\MaNouvelleLocalisation\Data\tempdb.mdf
move "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\templog.ldf"
D:\MaNouvelleLocalisation\Log\templog.ldf
net start MSSQL$SQL2005
```

IV-C-1-b - Depuis la version 2005

Dès la version 2005, plus moyen d'influer directement sur les tables système, donc plus moyen de modifier manuellement les points d'entrée.

```
SELECT name, filename
FROM tempdb..sysfiles
GO
ALTER DATABASE tempdb MODIFY FILE (NAME=N'tempdev', FILENAME =
'D:\NouvelleLocalisation\Data\tempdb.mdf' )
ALTER DATABASE tempdb MODIFY FILE (NAME=N'templog', FILENAME =
'D:\NouvelleLocalisation\Log\templog.ldf' )
GO
SHUTDOWN WITH nowait
```

```
move "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\tempdb.mdf"
"D:\NouvelleLocalisation\data\tempdb.mdf"
move "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\templog.ldf"
"D:\NouvelleLocalisation\log\templog.ldf"
net start MSSQL$SQL2005
```

IV-C-2 - Déplacer msdb

Pour msdb, le déplacement se fait comme pour une base utilisateur. Désactivez l'agent SQL afin qu'il n'y accède pas pendant la phase de restauration qui demande, je le rappelle, un accès exclusif à la base.

```
USE MASTER
GO
BACKUP DATABASE msdb TO DISK = N'd:\backup\msdb.bak'
GO
RESTORE DATABASE msdb FROM DISK = N'd:\backup\msdb.bak'
WITH MOVE N'MSDBData' TO N'C:\MaNouvelleLocalisation\DATA\MSDBData.mdf',
MOVE N'MSDBLog' TO N'C:\MaNouvelleLocalisation\log\MSDBLog.ldf'
GO
```

IV-C-3 - Déplacer master

Ici, difficile de se passer des outils graphiques.

