# Visual Studio .NET et Visual SourceSafe - Part 2

## Utilisation de VSS avec VS.NET

Dans la première partie nous avons présenté la configuration de l'outil d'archivage et de 'versionning' des sources proposé par Microsoft :

- Visual SourceSafe (actuellement la version 6.0 d).

Nous allons voir comment utiliser cet outil dans notre outil favori :

- Visual Studio

## Visual Studio .NET et Visual SourceSafe

Cet article s'inscrit dans la suite du précédent concernant l'utilisation de la solution d'archivage et de travail collaboratif :

- Microsoft Visual SourceSafe

La première partie est disponible ici :

- **Visual Studio .NET et Visual SourceSafe - Part 1 (FR)**

Cette seconde partie sera consacrée à l'utilisation de VSS avec les différents outils de développements possibles chez Microsoft.

Avant de commencer cet article, il existe deux versions de Visual SourceSafe fonctionnant avec Visual Studio.NET :

- Visual SourceSafe 6.0c for Visual Studio .NET
- Visual SourceSafe 6.0d

Toutes les autres ne sont pas fonctionnelles avec Visual Studio.NET (seulement avec les versions précédentes de VS). Il faut préciser que l'outil fonctionne mais ils ne pourront pas utiliser le mode intégré prévu dans VS.NET.

---

Etant donné qu'une équipe de Microsoft a créé un article sur le sujet (spécifique pour les projets ASP.NET) :

- **Utilisation de Visual SourceSafe avec ASP.NET : procédure détaillée (FR)**

Je vais faire une présentation plus rapide que prévue de cette utilisation. En revanche je rajouterai une partie pour l'outil encore très utilisé en entreprise :

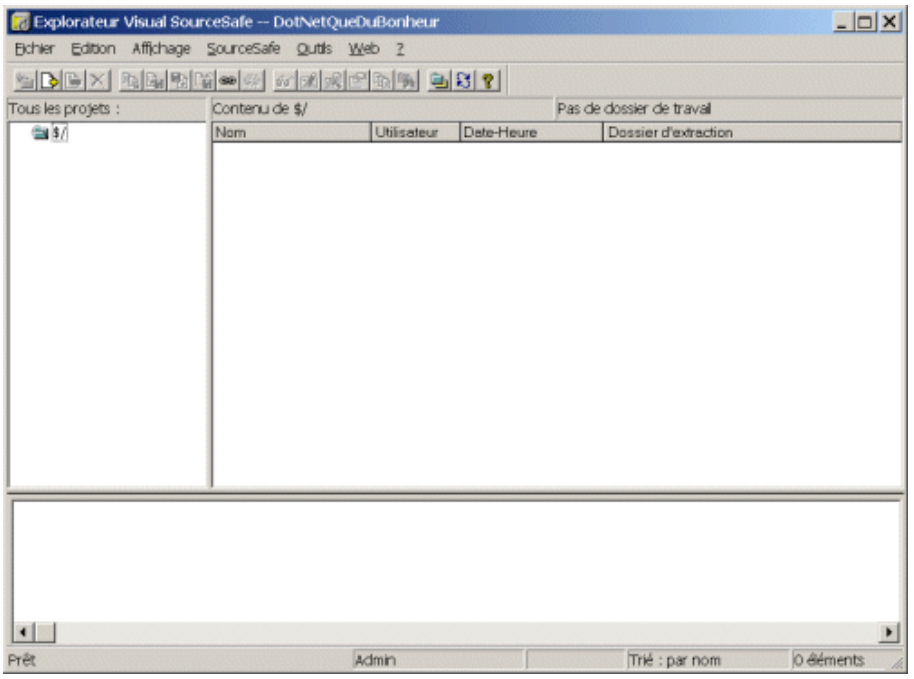
- Visual Studio 6 avec en particulier Visual Basic 6

**Utilisation de Visual SourceSafe avec Visual Studio 6**

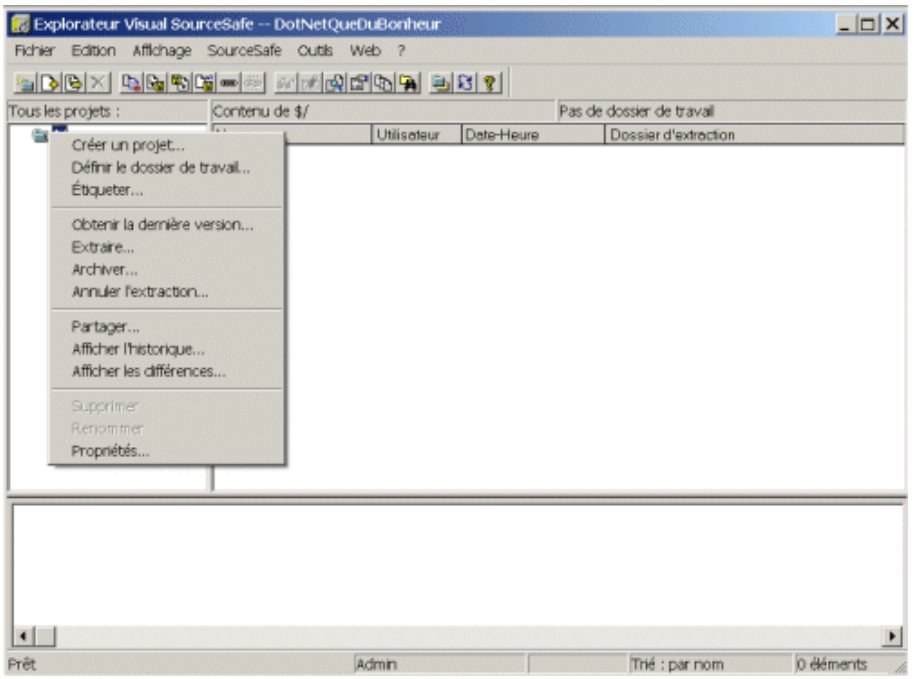
Débutons notre présentation par l'utilisation de ce logiciel avec un IDE bien connu qu'est Visual Studio 6. Je vais montrer l'exemple avec Visual Basic 6, en archivant dans la base créée dans le précédent article, le projet dont l'article est ici même :

- **Fab Network Shutdown (VB 6)**

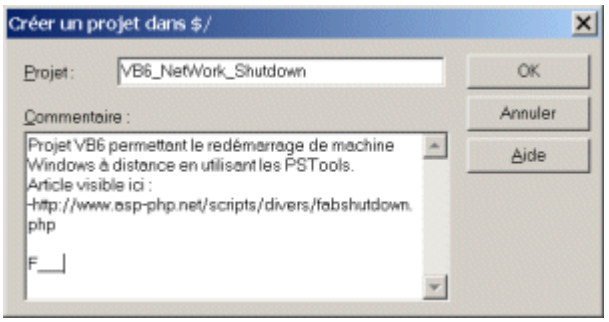
Nous allons pour cela lancer le logiciel Visual SourceSafe 6.0, nous authentifier avec un compte ayant les droits nécessaires sur la base (comme expliqué dans l'article précédent) pour obtenir la fenêtre suivante :



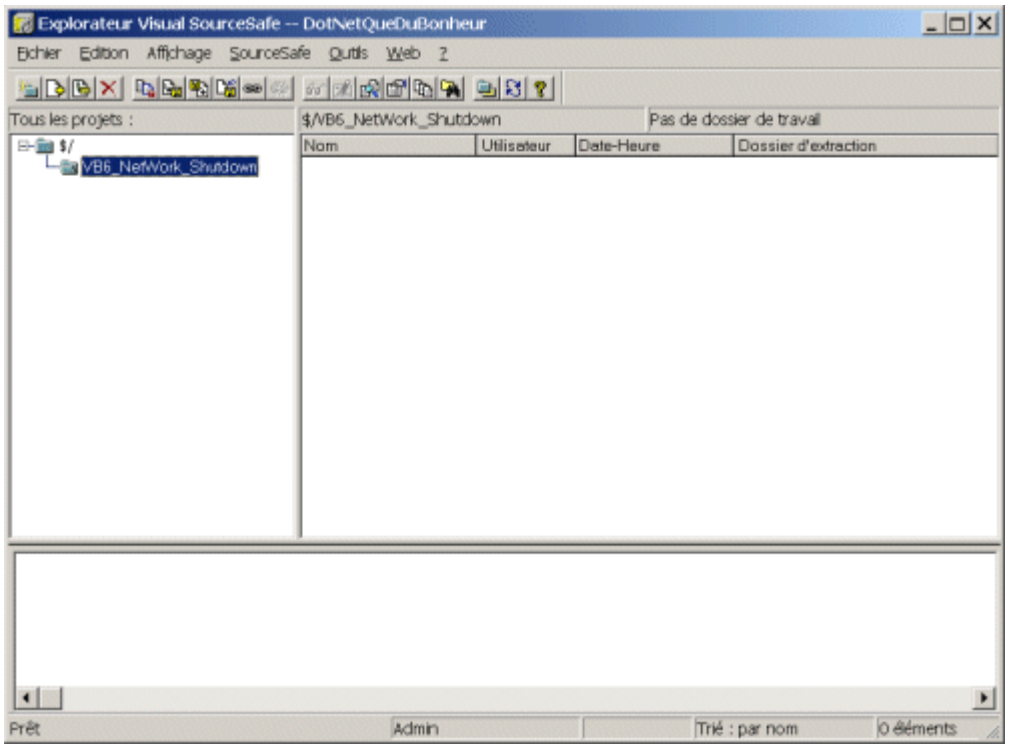

Maintenant dans cette base nous allons créer un nouveau projet, que nous nommerons 'VB6_NetWork_Shutdown'. Ceci se fait en cliquant dans "Fichier" > "Créer un Projet", ou plus simplement en cliquant avec le bouton droit de la souris sur la Racine (symbolisée par un $) de la base SourceSafe.



Alors on obtient une boite de dialogue nous demandant le nom du projet et un commentaire permettant de le décrire.

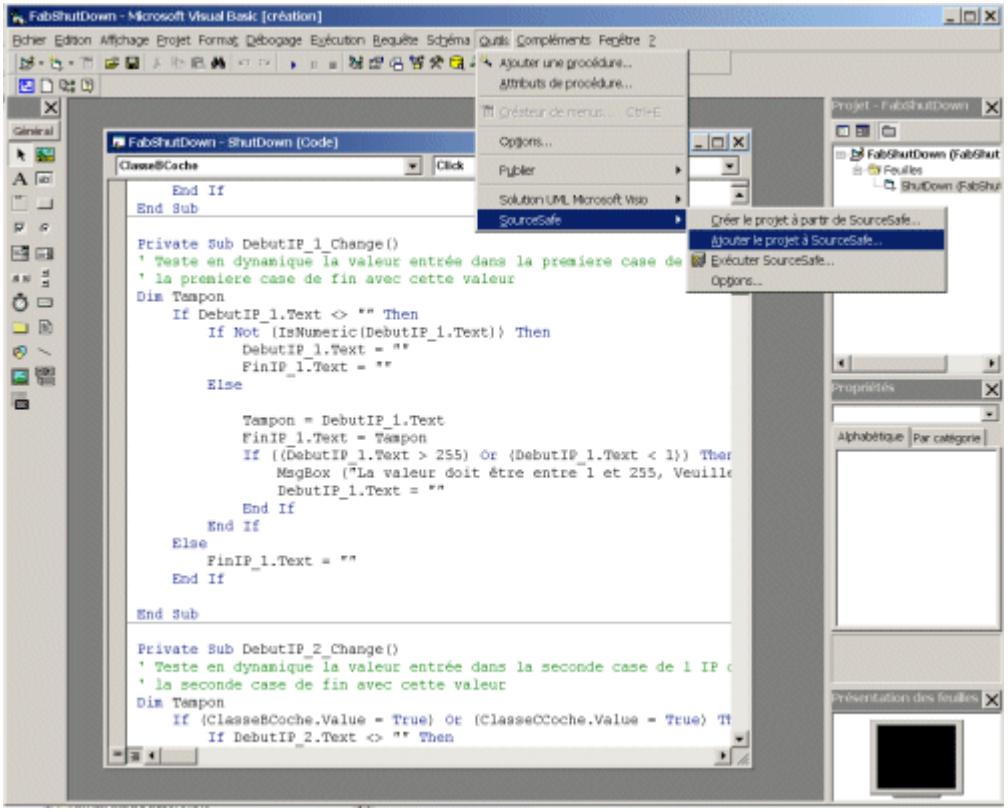



Une fois cette base créée, on voit apparaître dans l'explorateur de Visual SourceSafe une branche dans l'arborescence. En effet, les projets sont représentés dans VSS comme des 'répertoires' dans l'explorateur Windows.



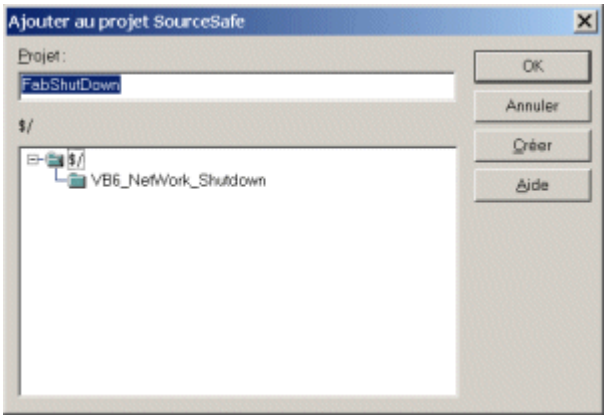
A partir de maintenant, nous pouvons ouvrir Visual Studio 6 afin d'ouvrir notre projet VB 6 afin de l'archiver dans le nouveau Projet de notre base SourceSafe.
Pour cela il faut aller dans :

- "Outils" > "SourceSafe > "Ajouter le Projet à SourceSafe"

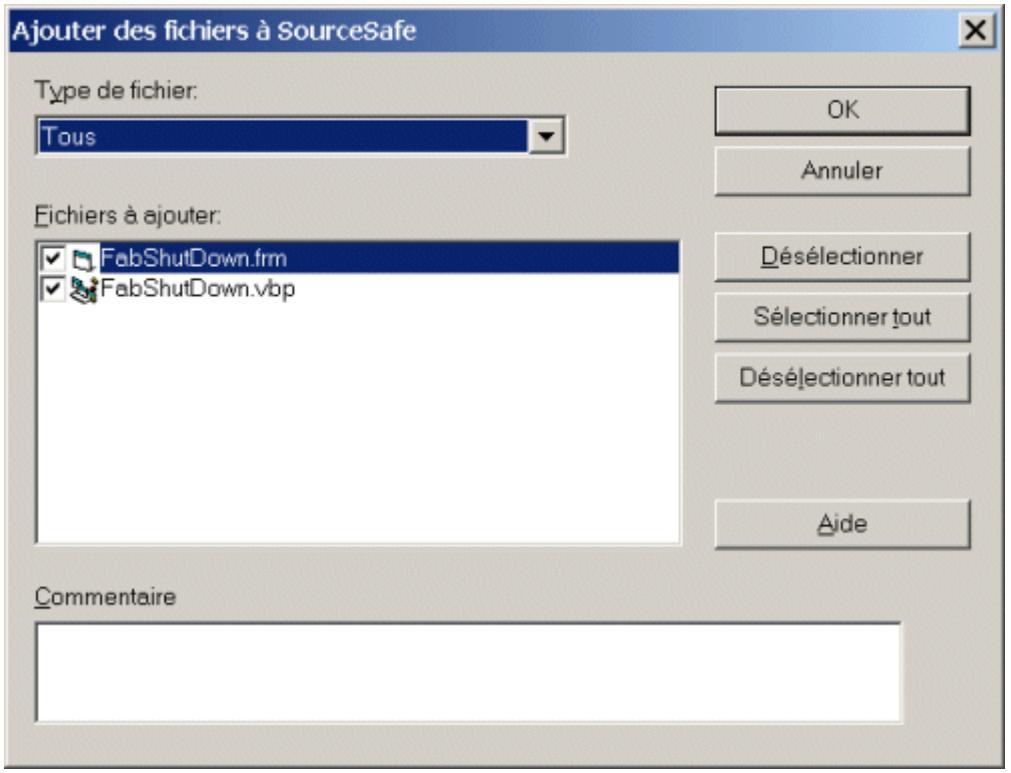



A ce moment là, nous aurons la boite de connexion habituelle à la base SourceSafe, nous demandant le login, le mot de passe et la base à prendre.

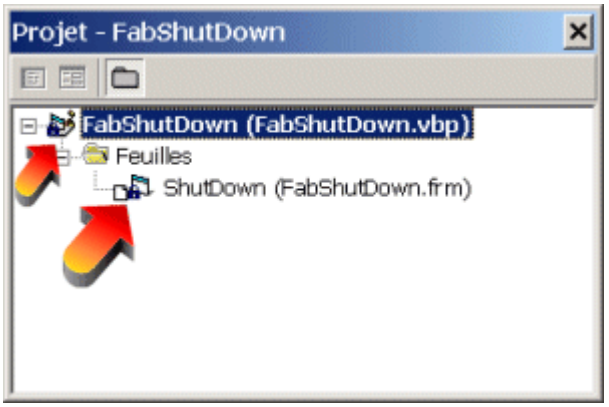
Nous auront alors un 'mini' Explorateur de la base SourceSafe nous invitant à choisir le projet (répertoire) dans lequel on archivera ce projet VB6.



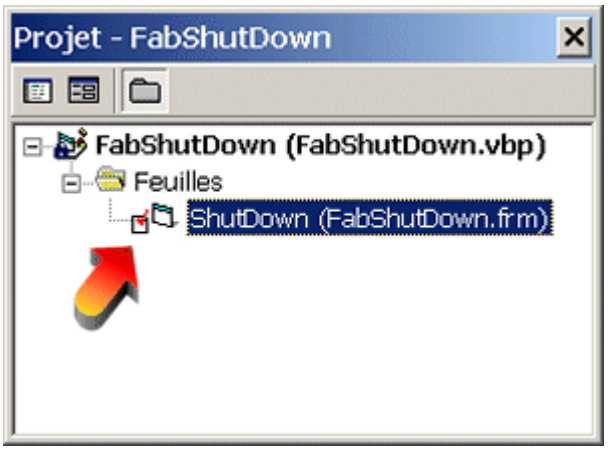
Un écran apparaît, nous demandant les types de fichiers (du projet VB6) que nous souhaitons archiver dans notre base SourceSafe. Dans notre exemple, nous archiverons tous les fichiers du Projet VB6.




Une fois cet archivage fait, nous voyons apparaître des petits cadenas sur les icônes des fichiers du projet VB6.



Ces cadenas symbolisent le fait que les fichiers soient archivés et qu'ils ne sont à ce moment précis pas modifiables. Pour cela, il faut faire une extraction (Bouton droit sur le fichier 'Extraire'). Cette extraction sera symbolisée sur l'icône du fichier par un petit "v" en rouge.




Tant que cette extraction n'est pas faite, il est impossible de modifier le fichier dans l'IDE. Chaque tentative qui sera faite vous demandera d'extraire le fichier que vous essayez de modifier. Une fois la modification faite, il suffit de cliquer avec le bouton droit de la souris sur le projet ou le fichier voulu pour l'archiver.

A partir de maintenant, vous savez archiver et extraire les fichiers d'une base SourceSafe, toutes les autres options (Etiquetage des projets, comparaison des fichiers, affichage des historiques, ...) sont très bien expliquées dans l'article de Microsoft, il est donc inutile d'insister là dessus.

- **Utilisation de Visual SourceSafe avec ASP.NET : procédure détaillée (FR)**

---

Dans cette partie, nous avons vu l'utilisation de Visual SourceSafe avec Visual Basic 6, vous trouverez dans l'article suivant l'utilisation de VSS avec Interdev 6 :

- **Interaction entre Visual Interdev et SourceSafe : contrôle de code source (FR)**

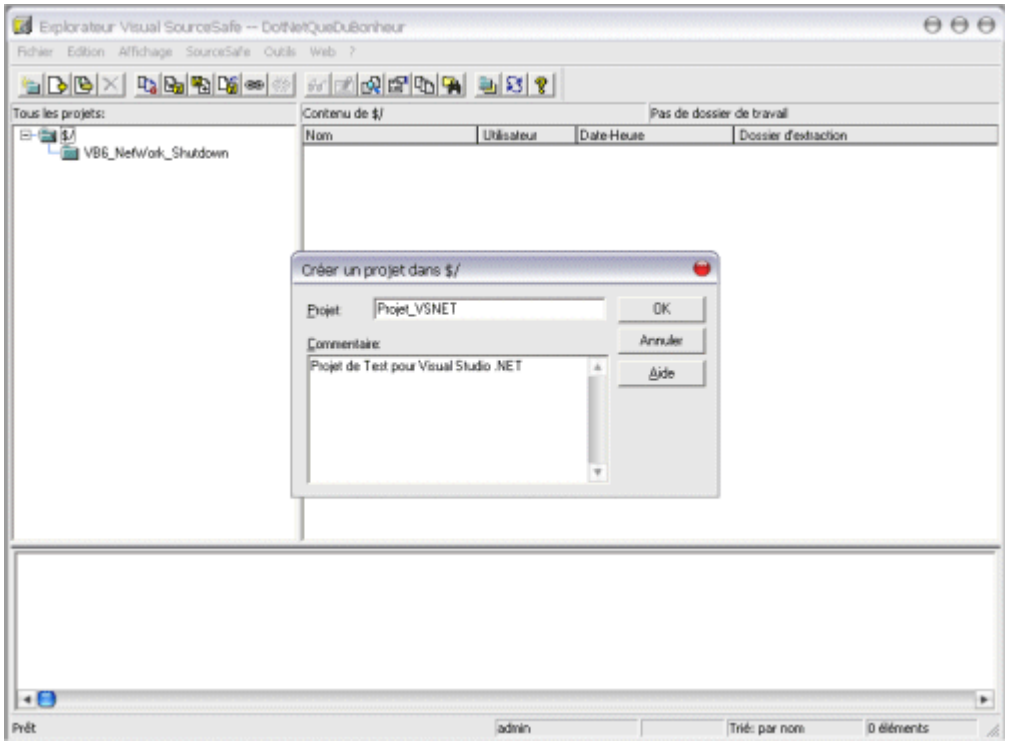
Passons maintenant à l'objectif premier de l'article qui est l'utilisation de VSS avec Visual Studio .NET. La version 2002 possède quasiment les mêmes possibilités que la version 2003, car cette partie a été très peu modifiée.

**Utilisation de Visual SourceSafe avec Visual Studio.NET**

Depuis la première version de Visual Studio .NET, VSS était pris en charge, nous allons donc montrer dans cette partie comment travailler donc avec VSS sous VS.NET.

Nous allons donc toujours travailler avec la même base SourceSafe, dans laquelle nous créerons d'abord un nouveau projet. Notre exemple sera très simple :

- Une solution comprenant un simple projet qui générera une librairie de classes : **ASP.NET Chaînes Utilitaires (FR)**




Maintenant que ce projet est créé dans VSS, voyons comment archiver le projet .NET dans la base.

Cela se fait par le sous-menu 'Contrôle de Code Source' qui se trouve sous 'Fichier'.

A partir de là, deux méthodes sont possibles :

- 1) Soit lier directement le projet en cours avec la base SourceSafe voulue
- 2) Ouvrir la gestion des liaisons des projets .NET vers VSS
- 3) Le "Glisser Déposer" des Répertoires depuis l'explorateur Windows vers VSS

Personnellement, je préfère la seconde solution, car on contrôle mieux ce que l'on fait, mais je vais présenter les deux possibilités.

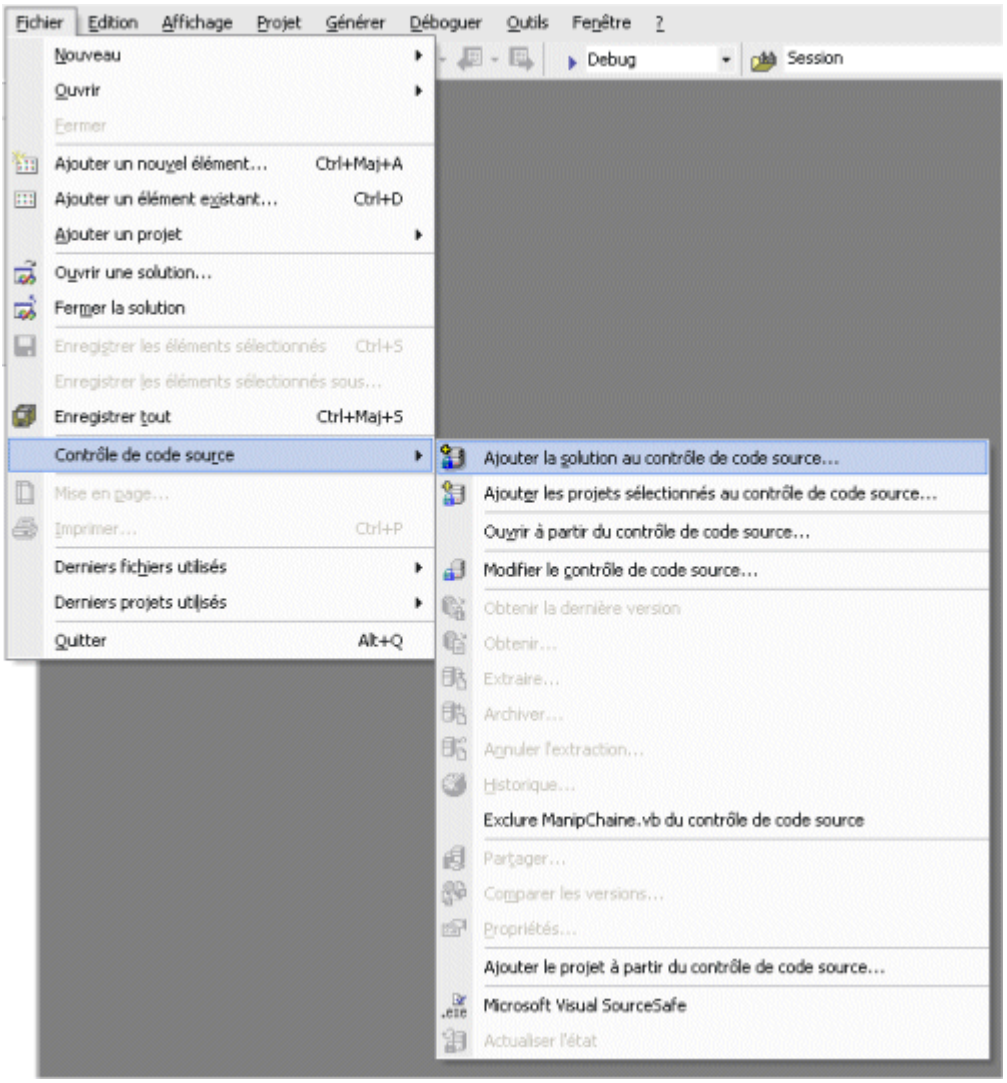
---

**Attention :** Visual Studio ne peut pas fonctionner en mode Multi Base. C'est-à-dire que lorsque vous travaillez sur des solutions complexes (avec de nombreux projets) sous Visual Studio.NET, vous devez avoir tous ces projets dans une seule base VSS. Il est impossible depuis VS.NET de se connecter sur plus d'une seule base VSS pour toute la solution. Cela vous oblige donc d'avoir tous les projets qui sont dans votre solution en cours dans cette seule base VSS.

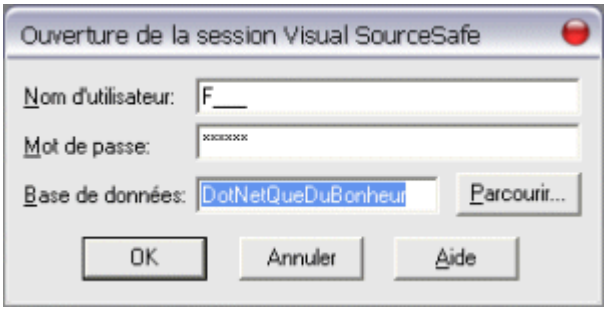
De ce fait pour les petites, voire les moyennes structures, il est fortement conseillé d'avoir une seule base VSS et de créer des projets dans cette base. Ainsi, vous pourrez gérer la sécurité dans cette base au niveau des droits dans Visual SourceSafe.

### 1) Liaison directe vers Visual SourceSafe

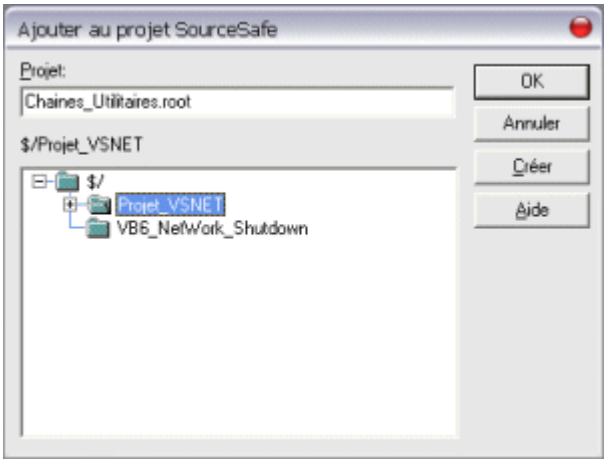
Ainsi dans 'Fichier', puis 'Contrôle de code source', nous trouvons 'Ajouter la solution au contrôle de code sources ...' :



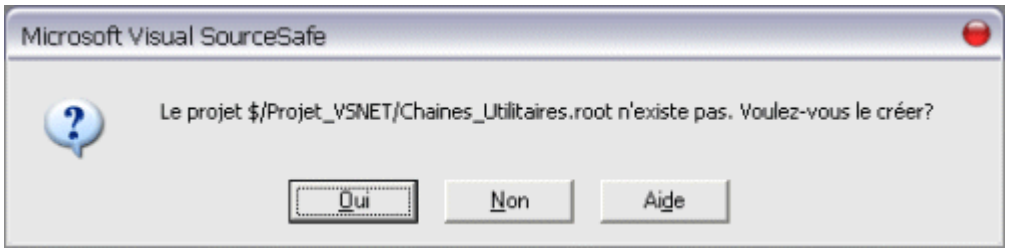

A ce moment nous retrouvons l'ouverture de la base SourceSafe, comme précédemment nous demandant le Login et Mot de Passe :



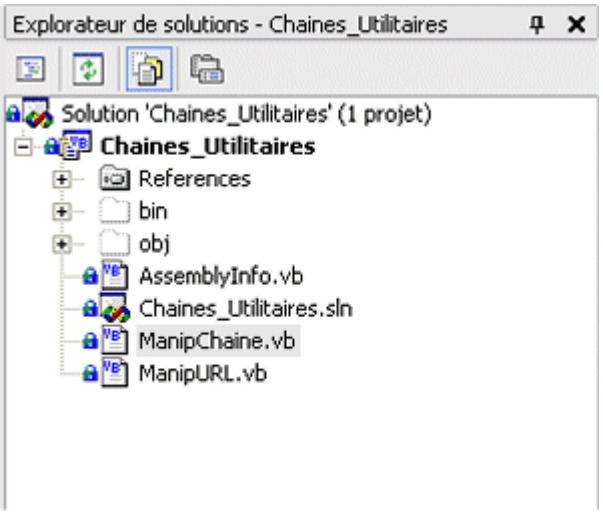
Une fois ces informations entrées, il vous demande dans quel projet de SourceSafe vous souhaitez ajouter ce projet :




A cet instant, vous aurez un message vous indiquant que le répertoire du nom de votre projet n'existe pas dans la base spécifiée et vous demande si vous acceptez de le créer. Par défaut il donnera comme nom de 'répertoire' celui de votre projet ajouté avec .root, vous pouvez changer ce nom dans la boite de dialogue précédente :




Une fois que vous avez cliqué sur 'Oui', Visual Studio.NET enverra tous les fichiers à SourceSafe pour que celui-ci les archive dans la base. Une fois cette archive faite, vous aurez dans votre explorateur de solution de petits cadenas sur tous les fichiers :

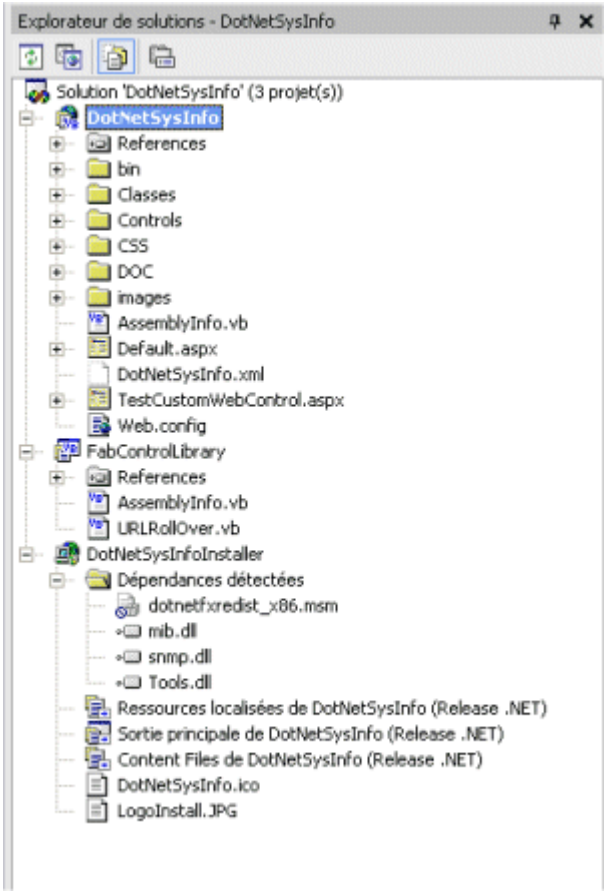



Comme vous le voyez cette méthode est très simple. Voyons maintenant la seconde méthode.

### 2) Utilisation de la Modification du Contrôle de Code Source

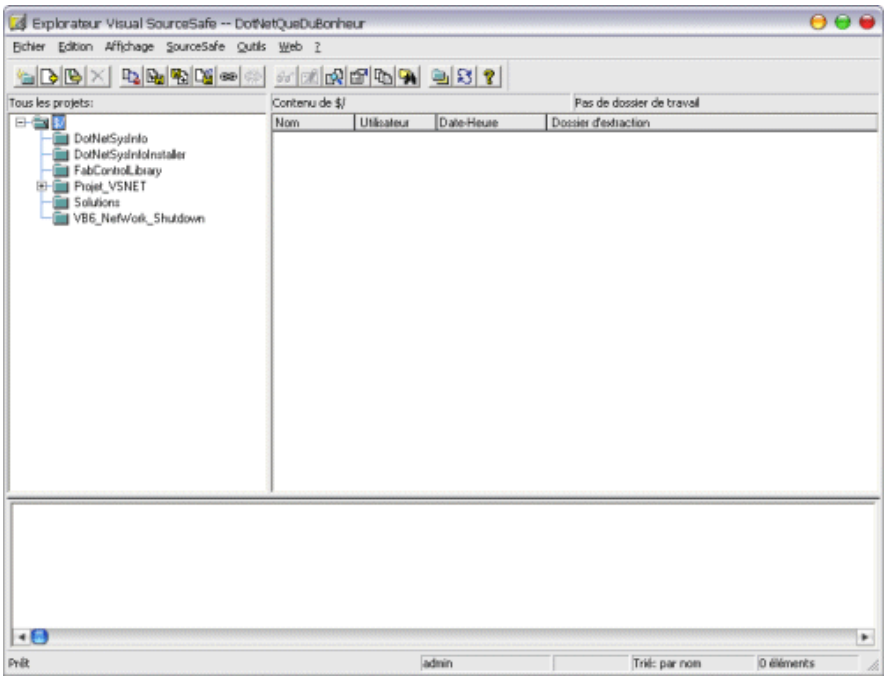
Pour mieux comprendre cette méthode, voyons celle-ci avec une solution contenant plusieurs projets : **DotNet SysInfo (FR)**

On voit déjà lorsque l'on charge la solution que celle-ci contient différents projets qui sont de plus chacun d'un type différent :

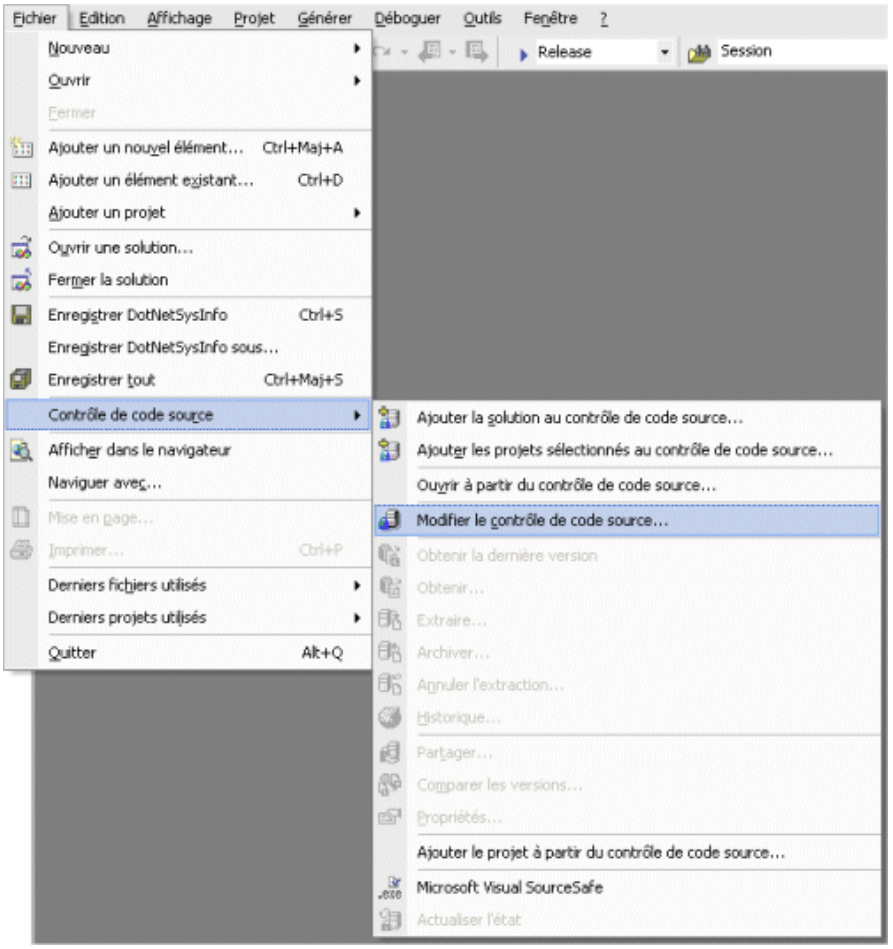



Le but est donc maintenant de définir pour chaque projet de la solution un répertoire différent de SourceSafe. Au passage, cette fois nous fixerons le fichier SLN dans un répertoire donné.

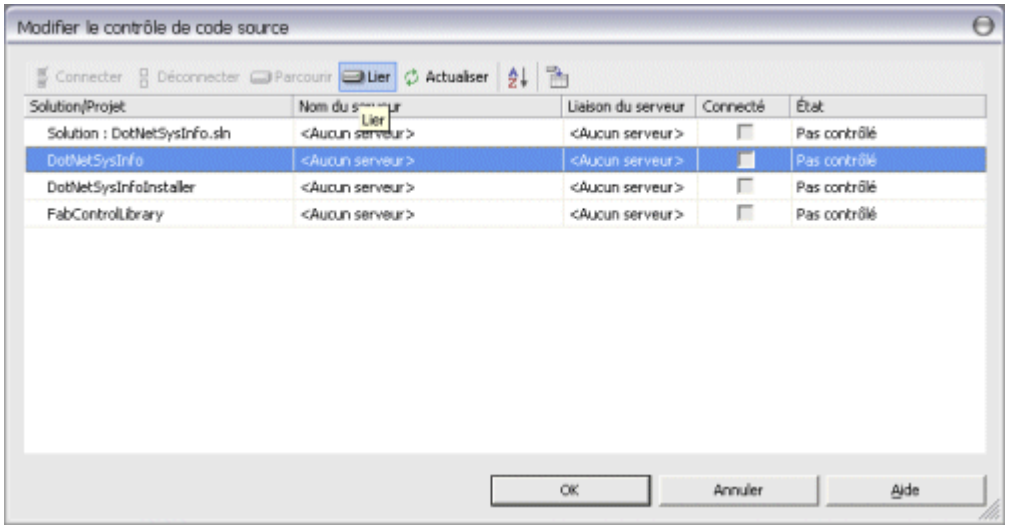
Au préalable, on va créer tous les répertoires nécessaires dans la base SourceSafe :



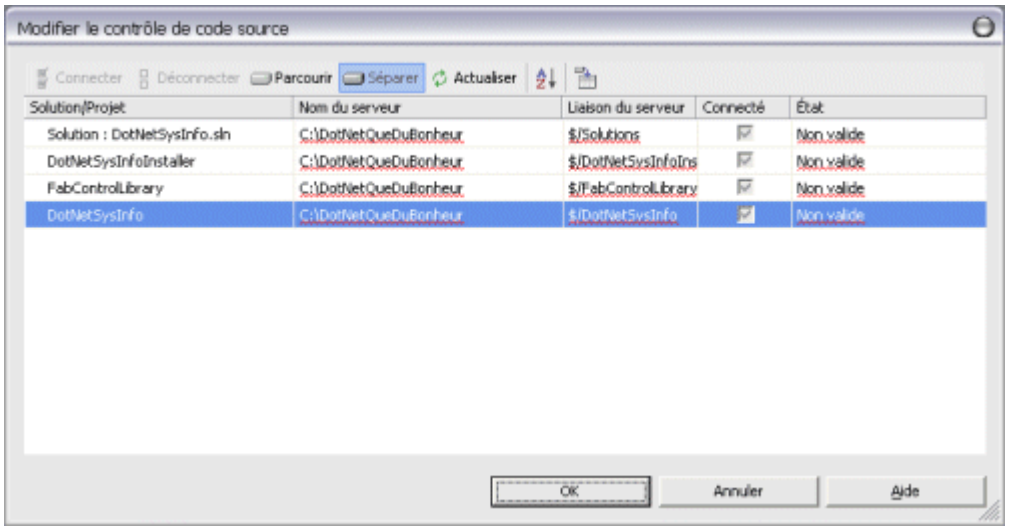

Ensuite, il faut aller dans 'Fichier', puis 'Contrôle de code source' et choisir alors 'Modifier le contrôle de code source...' :



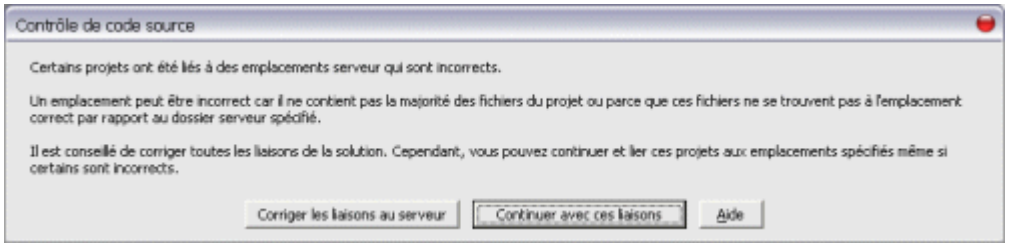
A ce moment, on obtient une fenêtre présentant la liste des projets de la solution courante ainsi que le fichier SLN :



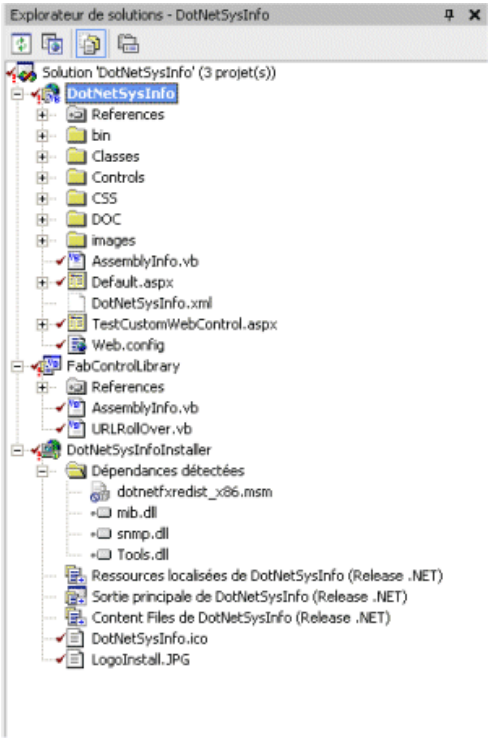

On va alors choisir projet par projet, le répertoire dans lequel nous archiverons celui-ci. Ainsi on clique sur le projet voulu pour le sélectionner et on clique sur 'Lier'. On choisit alors le répertoire voulu pour le Projet. On finit alors par obtenir l'écran suivant :



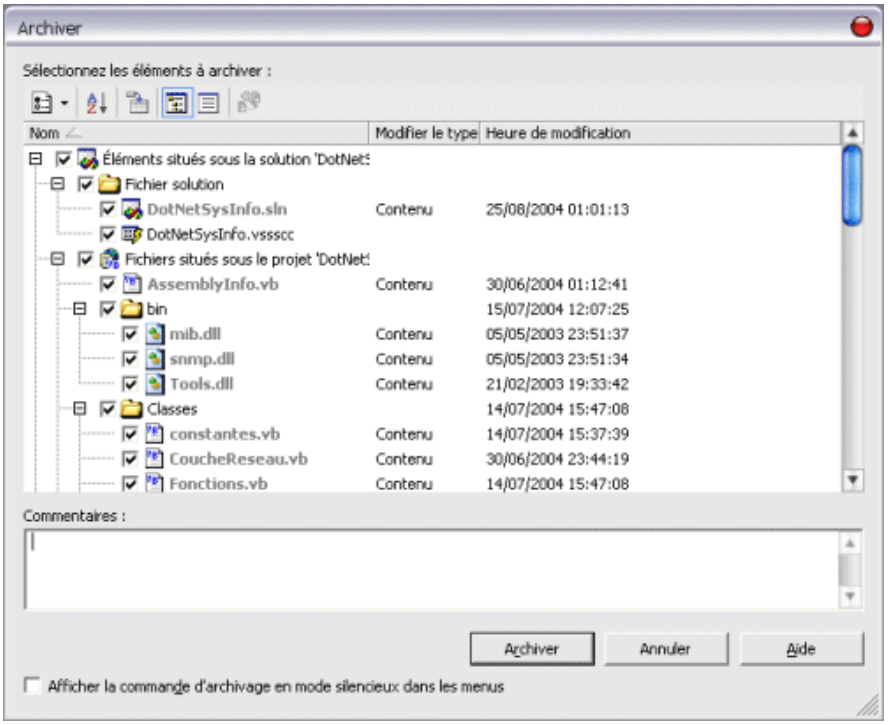

Les projets sont alors soulignés, car ils n'ont pas été archivés et donc la synchronisation avec la base SourceSafe n'est pas possible. A ce moment, on va donc cliquer sur 'OK' afin de valider cet archivage. A ce moment, on voit apparaître un message nous rappelant que les emplacements spécifiés dans la liste ne sont pas corrects. Il faut alors sélectionner 'Continuer avec ces liaisons' :



A ce moment, on voit apparaître des Symboles sur les fichiers :




On n'a plus alors qu'à archiver toute la solution (donc tous les projets) en cliquant avec le bouton droit et cliquant sur 'Archiver'. A ce moment, on obtient la fenêtre avec la liste des fichiers qui ont été modifiés et qui doivent être archivés :




Une fois cliqué sur 'Archiver', tous les projets et fichiers sont archivés. Désormais, à chaque fois que vous devrez modifier un fichier, il vous demandera de l'extraire et il faudra alors l'archiver par la suite.

Pour les autres options, je vous invite à les découvrir par vous même ou grâce aux différents sites traitant de Visual Studio .NET et Visual SourceSafe.

---

**Attention :** Il faut penser à ajouter absolument les DLL où autres fichiers extérieurs que vous utilisez dans vos projets. En effet, lorsque vous référencez une DLL externe ou un composant COM de l'OS (Office par exemple) vous trouverez ces DLL soit dans le répertoire que vous avez choisi (cas de la DLL externe que vous référencez manuellement) ou dans le répertoire BIN (cas d'une référence sur un composant COM).

Ces DLL sont donc correctes mais ne sont pas ajoutées au projet ce qui fait qu'elles ne seront pas archivées dans la base VSS.

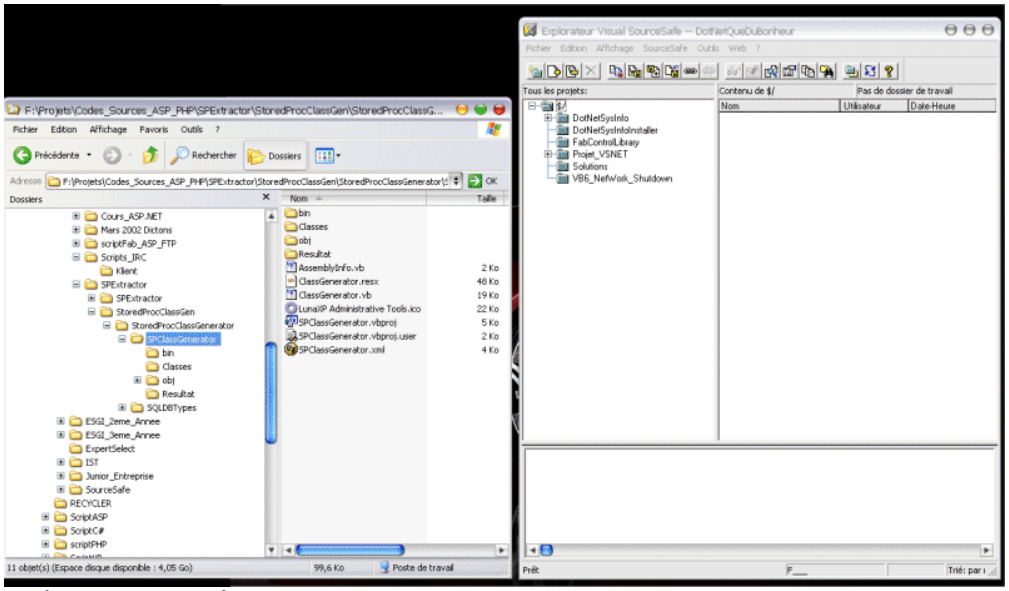
Le principal risque est donc la perte de ces fichiers en cas de panne de la machine de développement ou dans le cas de travail en XP (Extreme Programming).

En effet, en cas de développement de type XP ou collaboratif, plusieurs développeurs travaillent sur la même base VSS pour développer des modules différents, mais si un des développeur ajoute une référence externe mais n'ajoute pas la DLL dans la base VSS alors les autres développeurs ne récupéreront jamais cette DLL et la compilation plantera.

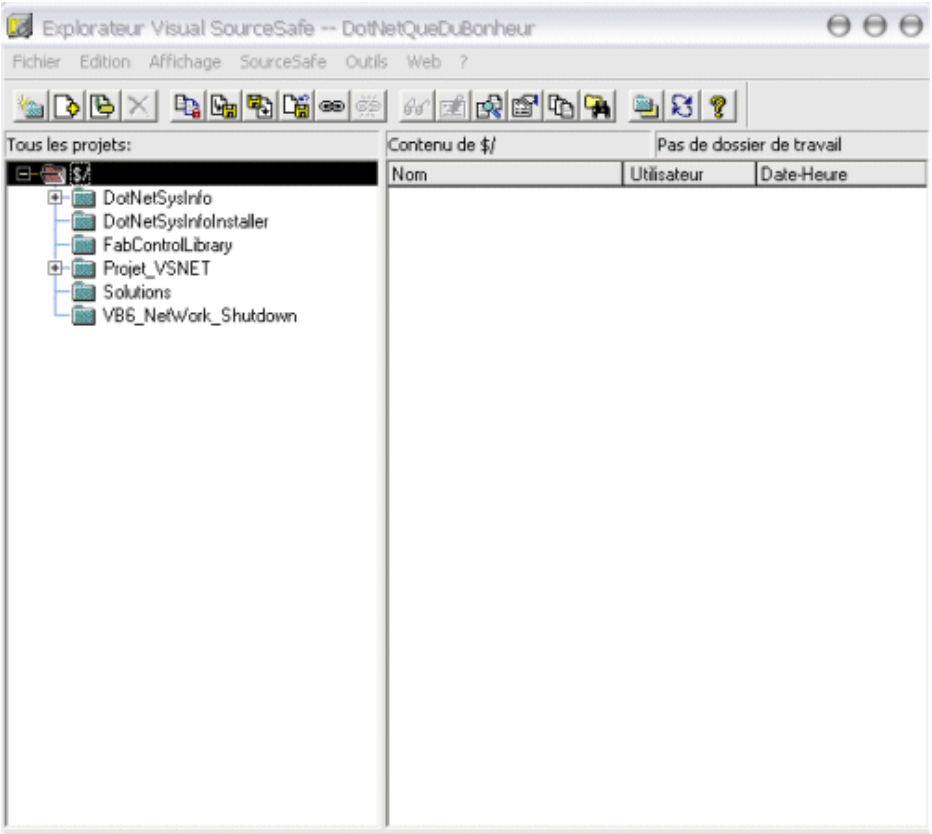
### 3) L'Explorateur de Fichiers et Visual SourceSafe

Cette solution n'est pas spécifique à Visual Studio, mais plutôt est une possibilité d'interaction directe entre Windows et VSS.

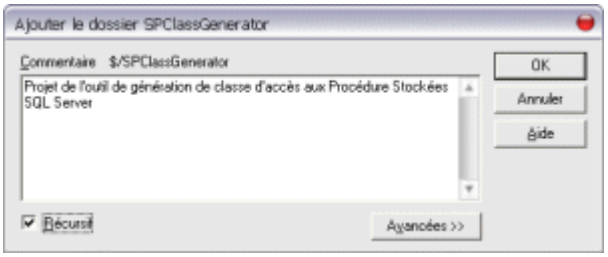
En effet, pour montrer cela, nous allons faire un archivage simple d'un projet .NET développé avec VS.NET. Tout d'abord ouvrons un explorateur de fichiers Windows dans une fenêtre juste au coté d'une seconde fenêtre avec le client VSS sur notre base SourceSafe.



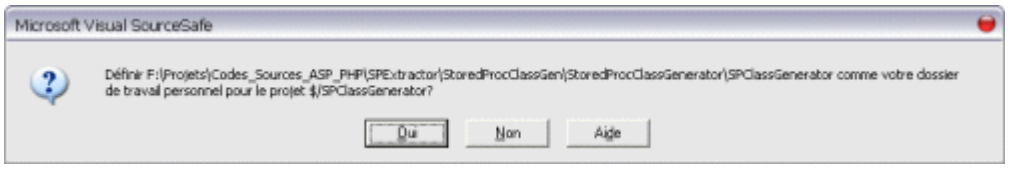

Maintenant sélectionnons le répertoire contenant les sources que nous souhaitons archiver, en maintenant le click de souris, et glissons ce répertoire dans la fenêtre VSS au point voulu (dans notre cas à la racine de la base SourceSafe).



Une fenêtre s'ouvre alors demandant de renseigner les détails sur ce nouvel archivage. Il faut cocher le récursif dans le cas où l'on a des sous répertoires dans le répertoire que l'on veut archiver.




Les fichiers sont alors tous remontés, puis une boite de dialogue apparaît demandant si l'on souhaite utiliser le répertoire d'où vient le projet comme dossier de travail pour ce projet.




Ainsi tous les fichiers et répertoires du répertoire que l'on a remonté sont maintenant dans cette base.

Il est donc à noter que cette remontée n'étant pas pilotée par Visual Studio.NET permet un archivage complet. Il n'y a pas les filtres utilisés par VS.NET pour ne remonter que les fichiers de la solution sans remonter les fichiers temporaires de compilation, etc. ...

Ce mode peut être utilisé si on a déjà une arborescence plutôt complexe que l'on souhaite archiver en une fois. C'est aussi celui à utiliser lorsque l'on utilise un outil de développement qui ne s'interface pas avec Visual SourceSafe, comme par exemple UltraEdit en ASP ou #Develop en .NET.

## Conclusion

Cet article vous a permis d'aborder Visual Studio.NET dans le but du travail collaboratif. Ainsi, le grand intérêt de VSS est de pouvoir archiver les fichiers d'un projet sur une base qui peut être déportée sur un serveur de fichiers par exemple. Ainsi tous les utilisateurs ayant accès à ce partage pourront charger la base et donc les fichiers. Ainsi, vous pouvez monter une équipe de développeurs qui travailleront sur une même solution.

Vous approcherez alors du travail collaboratif et de son aboutissement le plus poussé l'Extreme Programming (XP) dont une présentation est disponible ici :

- **Présentation de l'Extreme Programming sur IdealX (FR)**

---

Si vous souhaitez en savoir plus sur cet outil d'archivage de Microsoft, vous pouvez aller sur les sites dédiés :

- **Site Microsoft France pour Visual SourceSafe (FR)**
- **Tutorial concernant l'archivage des Procédures Stockées SQL Server (FR)**
- **Utilisation de Visual SourceSafe avec ASP.NET : procédure détaillée (FR)**

---

Je remercie Elise Dupont (MVP .NET) pour sa contribution dans les commentaires constructifs fournis sur cet article.

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)