Publier un site WEB .NET 2.0 avec VS 2005

La publication d'un site WEB .NET 2.0 avec VS 2005

Il est intéressant de savoir comment mettre en ligne son application WEB développée en ASP.NET 2.0 avec Visual Studio 2005.
Nous allons voir les possibilités offertes par l'outil de développement.

Introduction

Le développement de site WEB ASP.NET a changé depuis la version 2003 de Visual Studio avec la nouvelle version de Visual Studio (VS 2005). Ainsi, on peut par exemple développer son site en utilisant le serveur WEB intégré (CASSINI). Mais de nombreux outils ont été ajoutés.

Nous allons voir dans cet article comment publier son site WEB sur le serveur de destination.

Présentation

Lorsque l'on développe un site ASP.NET et que l'on utilise l'éditeur Visual Studio .NET 2005, on sait bien que les utilisateurs finaux ne travailleront pas sur notre machine de développement. Ainsi, on doit prévoir de déployer le site développé sur un serveur.

Dans de nombreux cas, nous devons avoir au moins une phase intermédiaire, c'est-à-dire passer par un serveur de test (qui peut être aussi celui d'intégration) avant la publication sur le serveur final (la production).

Nous n'évoquerons pas cette stratégie dans cet article, car elle doit être intégrée dans la stratégie de développement logiciel de l'équipe et de l'entreprise.

Nous verrons donc les deux outils intégrés à Microsoft Visual Studio .NET 2005 ainsi qu'un outil en ligne de commande :

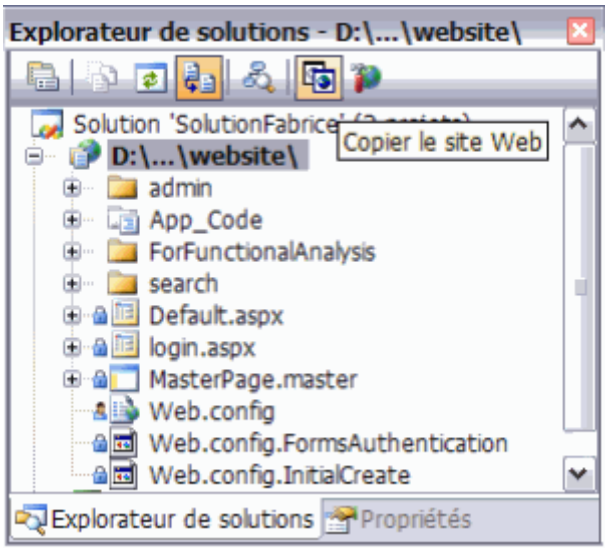
- La copie de site WEB
- La publication de site WEB
- La commande aspnet_compiler

Copier le site WEB

Le premier outil disponible est accessible via le menu (lorsqu'on est dans le projet WEB) :

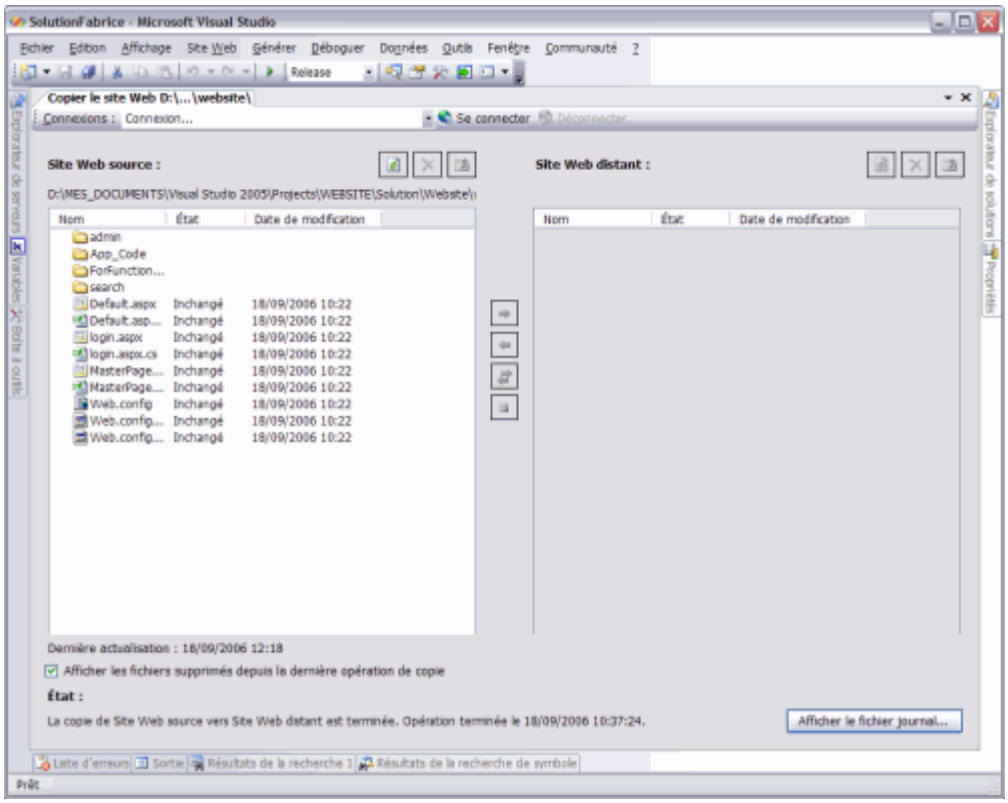
- Site WEB > Copier le site WEB

On le trouve aussi sous forme d'icone dans la fenêtre "Explorateur de solution" :



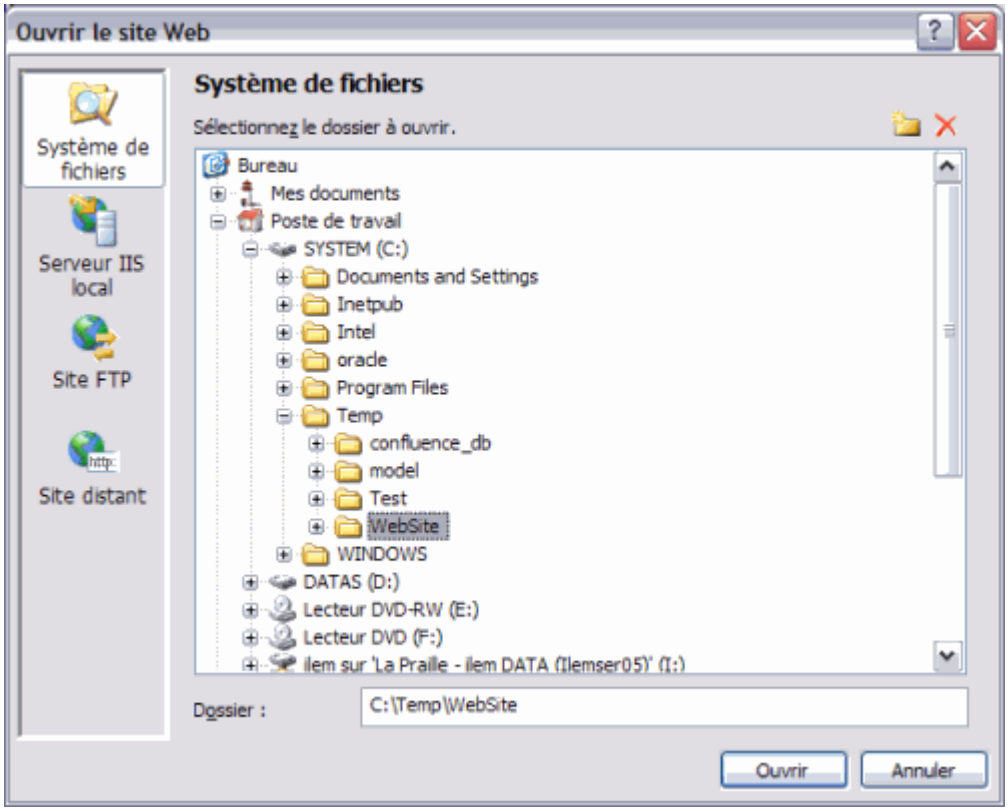
Une fois que l'on a cliqué sur le lien, une fenêtre se charge au centre avec deux parties distinctes :

- Répertoire local du site WEB (Site WEB source)
- Répertoire distant (Site WEB distant)



Nous devons dans un premier temps définir la connexion sur le site distant en cliquant sur le bouton "Se connecter". A cet instant un écran se charge nous donnant la possibilité de nous connecter sur un serveur distant via :

- Système de fichier (en local ou partage réseau)
- Serveur IIS local
- Site FTP
- Site distant (utilisant les extensions Frontpage)

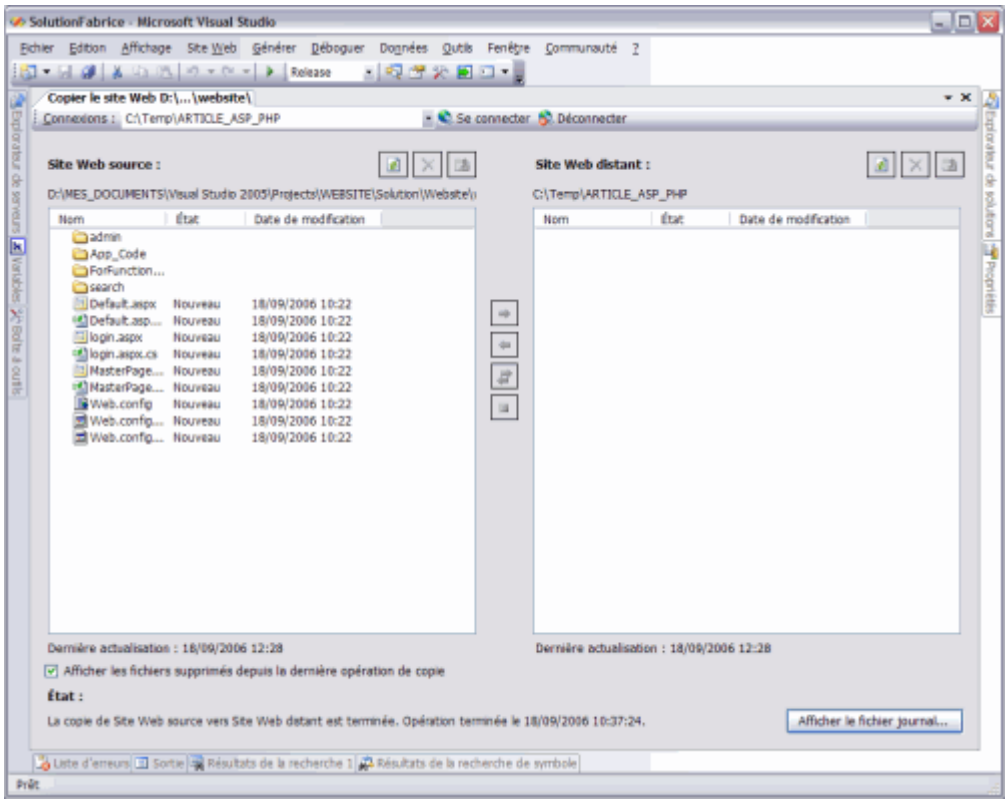


Nous utiliserons dans notre exemple le système de fichiers, mais les autres protocoles utilisent le même concept, il suffit donc d'avoir les paramètres et les droits de connexion afin d'adapter l'exemple à votre cas.

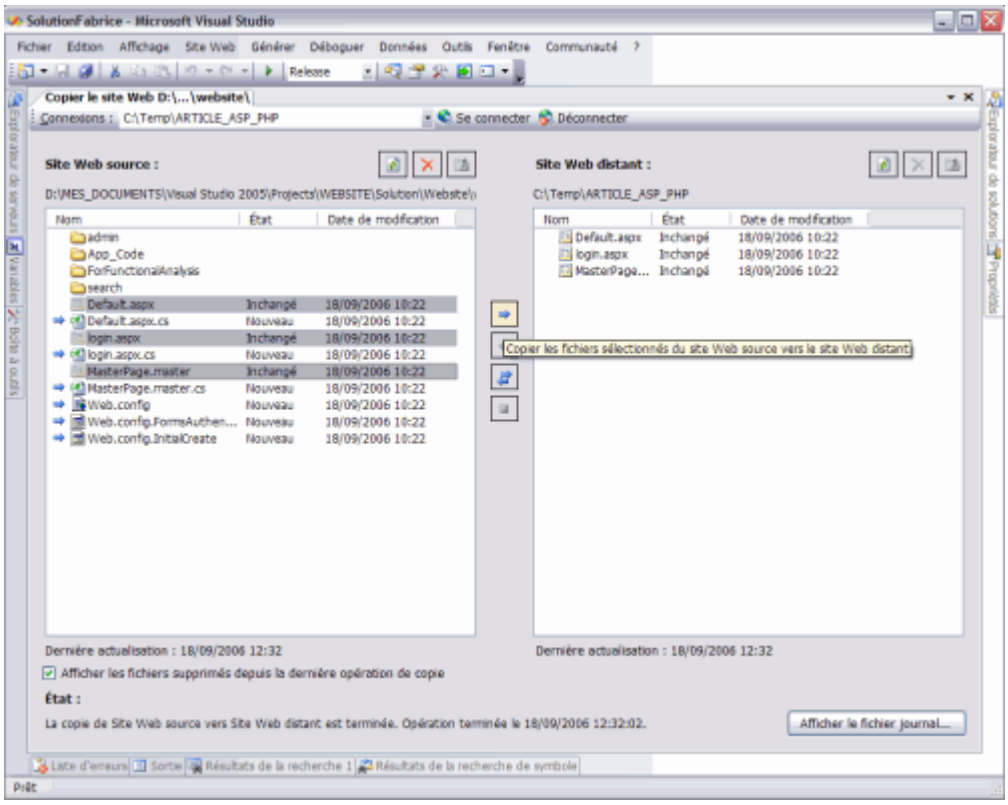
Donc, nous sélectionnons le répertoire destination :

- C:\Temp\ARTICLE_ASP_PHP\

Et cliquons sur Ouvrir. Nous voyons alors les deux répertoires (source et destination).



Nous sélectionnons alors simplement les fichiers que nous souhaitons passer de la source vers la destination par les flèches présentes entre les deux fenêtres de navigation.



On voit alors apparaître dans la colonne "Etat", le statut du fichier depuis le dernier transfert. Ainsi les fichiers transférés apparaissent comme "inchangés" alors que les autres sont détectés comme "nouveaux".

Attention :

Cet outil ne permet pas de faire la compilation préalable. Il doit être considéré de la même façon qu'un client FTP. Ainsi, on peut publier les fichiers souhaités vers le site WEB distant. Il ne permet pas en revanche de faire le nettoyage des fichiers qui ne sont pas nécessaires sur le site de destination (comme les fichiers Code Behind ou les fichiers de Visual Studio).

Pour effectuer cette tache, il faut passer par la publication de site WEB.

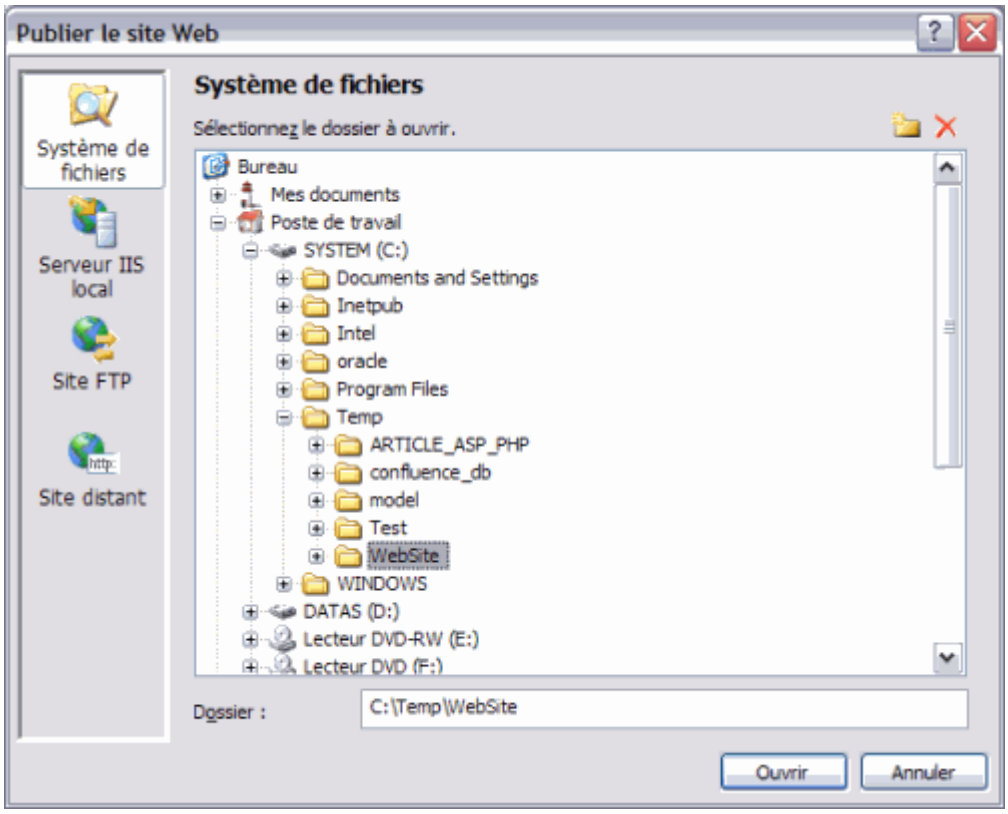
Publier le site WEB

L'assistant de publication de site WEB est accessible depuis le menu :

- Générer > Publier le site WEB

Une fois l'assistant lancé, il nous demande le site de destination qui peut être (comme pour la copie de site WEB) :

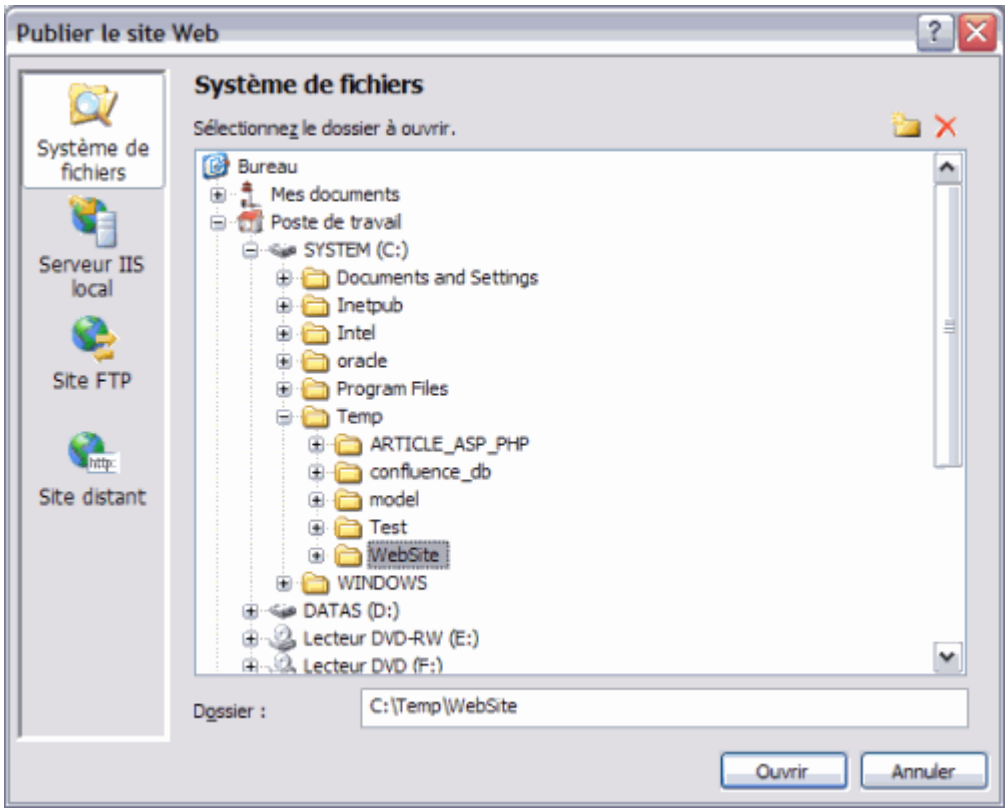
- Le système de fichier (local ou partage réseau)
- Le serveur IIS local
- Un site FTP
- Un site distant (via les extensions FrontPage)



Nous prendrons (comme pour la copie de site WEB) le système de fichiers avec le répertoire :

- C:\Temp\WebSite\

Nous devons alors spécifier les options de cette publication :



- La première option **"Autoriser ce site précompilé à être mis à jour"** permet de spécifier à l'assistant qu'il doit bien compiler tout le code behind dans la DLL, mais qu'il doit mettre les fichiers ASPX sans les modifier dans le site distant. Cette option est particulièrement utile si vous passez par un fournisseur externe pour la charte graphique.

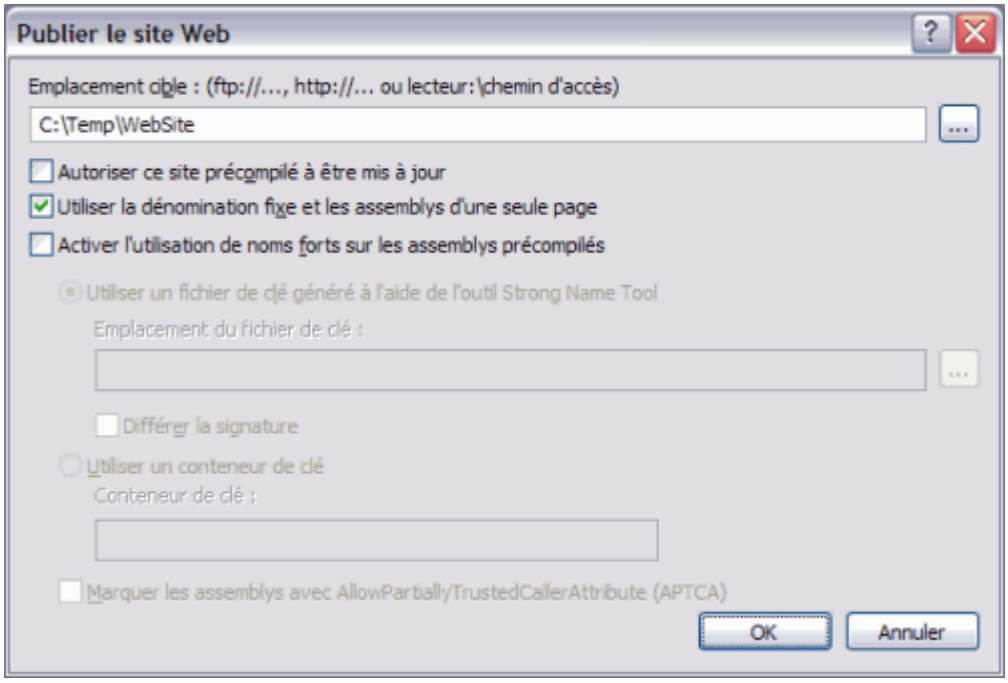
  Dans notre exemple, nous décocherons cette option.

- La seconde option **"Utiliser la dénomination fixe et les assemblys d'une seule page"** permet d'effectuer la compilation complète de tout le projet WEB dans des DLL. Ainsi tous les fichiers ASPX seront présents sur le site de destination, mais seront vides. Cette option est particulièrement intéressante sur les serveurs de production afin d'accélérer la visualisation du site WEB, car celui-ci ne recompile pas le contenu des pages ASPX.

  Dans notre exemple, nous cocherons cette option.

- La partie du bas "Activer l'utilisation de noms forts sur les assemblys précompilés" permet d'intégrer l'usage de SN.EXE (Strong Name Tool) dans le déploiement du site WEB afin de ne bloquer la modification des sources par un tiers.

  Dans notre exemple, nous décocherons cette option.



Lorsque l'on clique sur OK, Visual Studio effectue la compilation de tout le projet du site WEB, puis publie les fichiers résultat dans le répertoire défini.

Dans le cas où on a sélectionné un répertoire local (comme notre exemple), il nous suffit de placer tous les fichiers dans le répertoire destination (en n'oubliant pas de modifier les fichiers de configuration si besoin).

Mais voyons maintenant l'usage de l'outil en ligne de commande, dans le cas de passage par des outils de Build ou des batchs personnalisés.

La commande aspnet_compiler

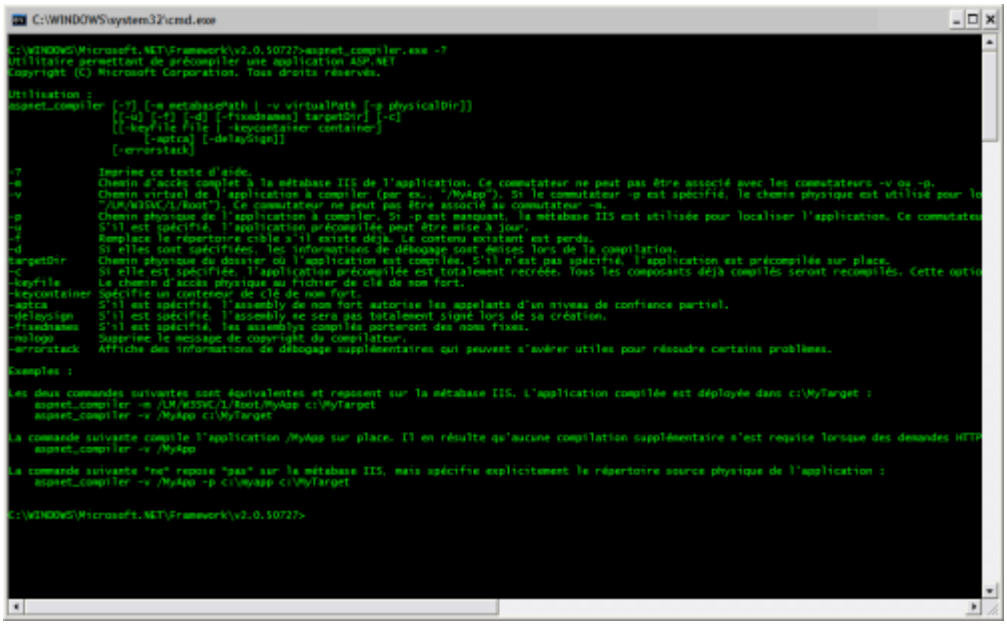
Cet outil est fourni avec le FrameWork .NET 2.0, on le trouve dans le répertoire :

- %windir%\Microsoft.NET\Framework\v2.0.50727\

Ainsi, on trouve dans ce répertoire l'exécutable :

- aspnet_compiler.exe

La commande "aspnet_compiler.exe -?" nous permet d'obtenir l'aide sur son usage.



Dans notre cas, nous n'utilisons pas IIS comme serveur WEB pour notre projet. De ce fait, nous devons spécifier le répertoire source de notre site WEB ainsi que le répertoire de destination.

La commande à utiliser sera donc :

- aspnet_compiler.exe -p C:\MonRepertoire\ProjetDotNet\website -v -f C:\Temp\ASPNET_Compiler

Cette commande va alors effectuer le même travail que l'assistant précédent avec la seconde option cochée. Nous obtenons alors tous les fichiers compilés en DLL et les fichiers ASPX sont vidés de leur contenu.

Attention :

Je vous invite à lire l'aide de la ligne de commande afin d'adapter celle-ci à votre configuration.

Conclusion

Ce sujet nous a permis d'optimiser le temps de déploiement par l'usage des assistants intégrés dans Visual Studio.

Vous pouvez aussi aller plus loin dans la démarche en utilisant des outils d'industrialisation de développement comme Visual Studio Team System qui intègre son propre système de compilation et de déploiement.

Vous pouvez aussi voir le projet Open Sources NAnt qui permet aussi d'automatiser ces taches de déploiement.

Voici quelques liens utiles si cet article vous a intéressé :

- **Comment : publier des sites Web (Visual Studio)**
- **Procédure pas à pas : publication d'un site Web**
- **Publication de sites Web**
- **Gestion des fichiers pendant une précompilation ASP.NET**
- **Comment : précompiler des sites Web ASP.NET à des fins de développement**
- **Centre de développement ASP.NET**
- **Découvrez & Formez-vous avec le Coach ASP.NET**
- **Centre de développement Visual Studio Team System**
- **NAnt Home Page**

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)

Consultant Technique **ilem SA**