

Création d'une solution de déploiement SharePoint

Créer la solution pour notre composant SharePoint



Après avoir suivi toutes les étapes pour la création de notre composant (Feature) SharePoint, il faut maintenant voir comment le déployer.

On pourrait faire ce travail en faisant la copie des différents fichiers mais ceci est très fastidieux. Nous verrons donc comment créer une solution de déploiement SharePoint.

Introduction

Nous avons vu dans l'article précédent comment créer très rapidement un composant SharePoint. Ce composant est destiné à effectuer une tâche d'administration au travers d'une page ASPX :

- **Créer une Feature d'administration sous SharePoint**

La question se pose alors pour la mise en place de ce composant. Microsoft a répondu à cette question pour SharePoint 2007 avec les solutions de déploiement (fichiers WSP).

Présentation

Pour installer un composant (une feature), on peut copier tous les fichiers nécessaires dans les différents sous-répertoires de SharePoint, puis effectuer un IISRESET afin de recharger toute la couche applicative.

Cette solution, bien que fonctionnelle, n'est pas du tout la plus pratique et efficace. En effet, dans ce choix d'installation, il faut effectuer ce travail sur chaque serveur SharePoint dans le cas d'une ferme. Il faut ensuite effectuer ce même travail lors de l'ajout d'une nouvelle machine dans notre ferme.

Ce travail de copie devient dès lors complexe dans les architectures complexes.

On peut aussi ajouter la gestion éventuelle de ce composant qui n'est plus faisable et qui nécessite de toucher les fichiers de chaque machine en cas de désinstallation éventuelle.

Nous verrons dans cet article les différentes étapes nécessaires pour rassembler les différents fichiers de notre composant dans un seul fichier solution. Nous verrons ensuite comment déployer ou supprimer cette solution sur notre ferme SharePoint.

Création des fichiers de définition

Pour créer notre solution, nous devons fournir différents fichiers qui vont servir à différentes étapes :

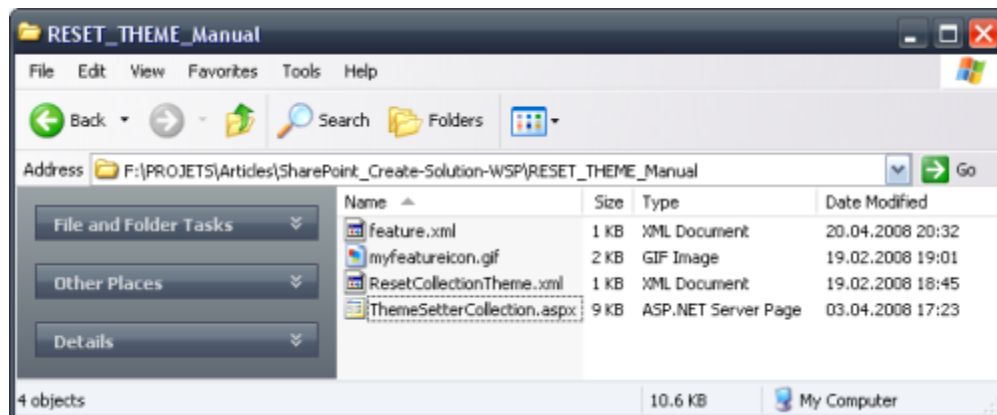
- **Un fichier "manifest.xml"** : qui va fournir à SharePoint le placement de chaque fichier dans la ferme
- **Un fichier DDF** : permettant la création du fichier Compressé WSP (CAB renommé en WSP)

Voyons comment créer ces fichiers un par un. Nous mettrons pour cela tous les fichiers de notre composant à la racine de notre répertoire de travail.

Fichier Manifest.XML

Nous devons fournir les différents renseignements pour chaque fichier de notre composant qui sont :

- feature.xml : fichier de définition de notre composant
- ResetCollectionTheme.xml : Fichier de définition des éléments de notre composant
- myfeatureicon.gif : Image icône pour notre composant
- ThemeSetterCollection.aspx : Page ASPX de notre composant



Nous devons tout d'abord générer un GUID pour notre solution, nous pouvons par exemple utiliser Visual Studio pour cela, comme c'est expliqué dans le message :

- **SharePoint : Comment retrouver le Create GUID dans Visual Studio 2008**

Notre fichier XML va contenir la définition de notre solution (avec le GUID d'identification pour le champs SolutionID) puis la définition de chaque composant :

```
<Solution
  SolutionId="F5C71641-661C-49a9-A027-62009793D04C"
  xmlns="http://schemas.microsoft.com/sharepoint/">
  ...
```

Puis nous retrouverons la spécification de notre fichier de composant (Feature.xml) dans un bloc XML "FeatureManifests" :

```
...
<FeatureManifests>
  <FeatureManifest Location="ResetAllSiteCollectionTheme\feature.xml" />
</FeatureManifests>
...
```

On doit maintenant ajouter la spécification de chaque fichier additionnel (Image et fichier ASPX) dans un bloc XML "TemplateFiles" :

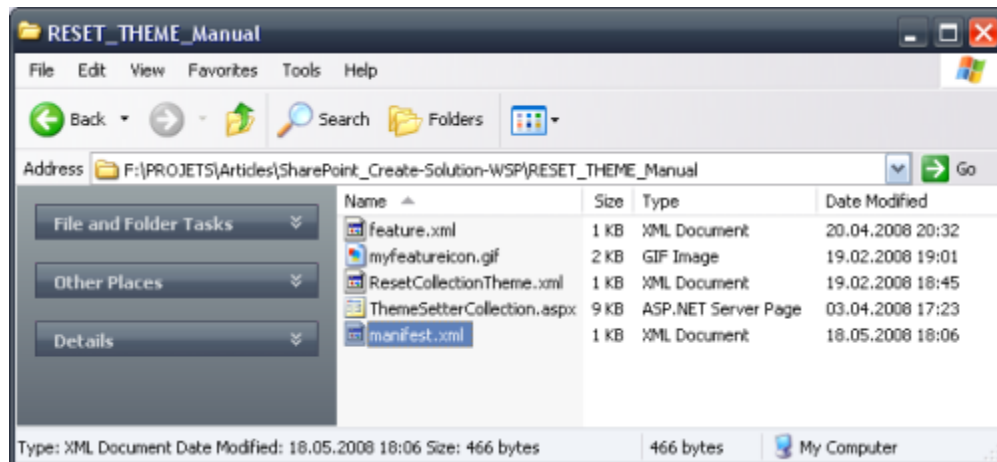
```
...
<TemplateFiles>
  <TemplateFile Location="IMAGES\RESETHEME\myfeatureicon.gif" />
  <TemplateFile Location="LAYOUTS\ThemeSetterCollection.aspx" />
</TemplateFiles>
...
```

Notre composant est très simple et ne contient pas de fichier DLL, nous n'avons donc pas besoin du bloc XML "Assemblies".

Notre fichier Manifest.XML complet est le suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<Solution
  SolutionId="F5C71641-661C-49a9-A027-62009793D04C"
  xmlns="http://schemas.microsoft.com/sharepoint/">
  <FeatureManifests>
    <FeatureManifest Location="ResetAllSiteCollectionTheme\feature.xml" />
  </FeatureManifests>
  <TemplateFiles>
    <TemplateFile Location="IMAGES\RESETHEME\myfeatureicon.gif" />
    <TemplateFile Location="LAYOUTS\ThemeSetterCollection.aspx" />
  </TemplateFiles>
</Solution>
```

Notre fichier de définition est donc maintenant OK.



Si vous souhaitez de plus amples informations, je vous invite à lire le texte de Microsoft :

- **Creating a Solution**

Maintenant, nous devons créer le second fichier qui nous permettra d'effectuer la compression.

Création du fichier DDF

Le fichier DDF est un fichier qui va permettre à l'outil interne de Microsoft de générer le fichier compressé avec le format CABINET (CAB).

Pour notre exemple, la création de ce fichier est assez simple. Nous devons d'abord fournir les renseignements de base de notre fichier compressé.

```
.OPTION EXPLICIT ;Génère les erreurs
.Set CabinetNameTemplate=RESET_THEME_Manual.wsp ;fournit le nom du fichier final
.set DiskDirectoryTemplate=CDROM ;Tous les fichiers seront dans un seul répertoire
.Set CompressionType=MSZIP ;Tous les fichiers seront compressés dans le fichier archive
.Set UniqueFiles="ON" ;Un seul fichier d'archive
.Set Cabinet=on ;Type de fichier cabinet
```

Maintenant que les paramètres de base sont définis, nous allons fournir les informations pour chaque fichier de notre composant.

On peut regrouper ses fichiers par destination de copie. Commençons par le fichier manifest :

```
.Set DestinationDir=
manifest.xml
```

Puis nous fournissons les informations pour les fichiers de définition de Feature :

```
.Set DestinationDir=ResetAllSiteCollectionTheme
feature.xml
ResetCollectionTheme.xml
```

Mais on peut aussi fournir pour chaque fichier le point de départ et le point de destination, comme pour l'image ou le fichier ASPX :

```
myfeatureicon.gif "IMAGES\RESETHEME\myfeatureicon.gif"
ThemeSetterCollection.aspx "LAYOUTS\ThemeSetterCollection.aspx"
```

On obtient donc le fichier que l'on nommera "makecab.ddf" avec le contenu :

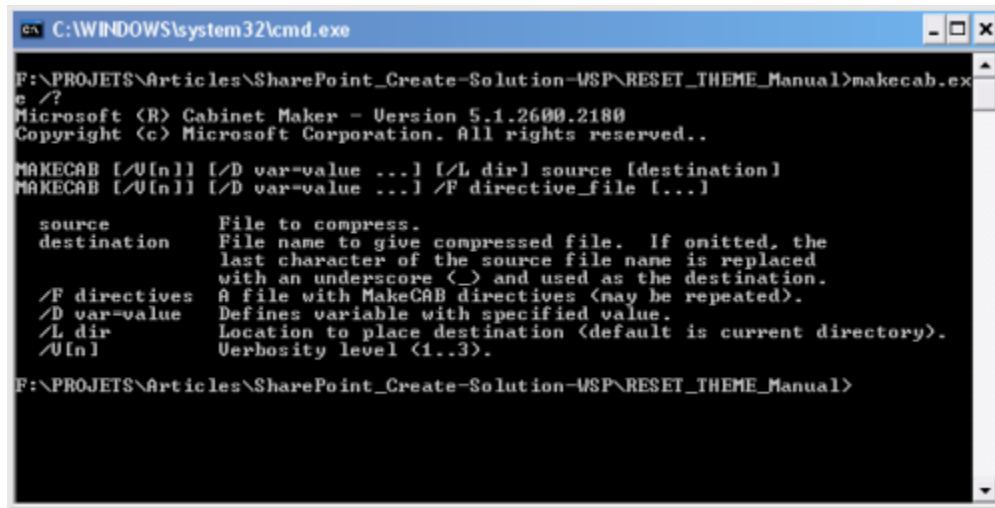
```
;Départ du fichier - MakeCab.ddf
.OPTION EXPLICIT ;Génère les erreurs
.Set CabinetNameTemplate=RESET_THEME_Manual.wsp ;fournit le nom du fichier final
.set DiskDirectoryTemplate=CDROM ;Tous les fichiers seront dans un seul répertoire
.Set CompressionType=MSZIP ;Tous les fichiers seront compressés dans le fichier archive
.Set UniqueFiles="ON" ;Un seul fichier d'archive
.Set Cabinet=on ;Type de fichier cabinet

.Set DestinationDir=
manifest.xml
.Set DestinationDir=ResetAllSiteCollectionTheme
feature.xml
ResetCollectionTheme.xml
myfeatureicon.gif "IMAGES\RESETHEME\myfeatureicon.gif"
ThemeSetterCollection.aspx "LAYOUTS\ThemeSetterCollection.aspx"

;Fin du fichier - MakeCab.ddf
```

Maintenant que notre fichier de définition MakeCab.DDF, nous devons utiliser l'outil fourni avec Windows :

- **MAKECAB.EXE**



```
C:\WINDOWS\system32\cmd.exe
F:\PROJETS\Articles\SharePoint_Create-Solution-WSP\RESET_THEME_Manual>makecab.exe /?
Microsoft (R) Cabinet Maker - Version 5.1.2600.2180
Copyright (c) Microsoft Corporation. All rights reserved..

MAKECAB [/U{nl}] [/D var=value ...] [/L dir] source [destination]
MAKECAB [/U{nl}] [/D var=value ...] /F directive_file [...]

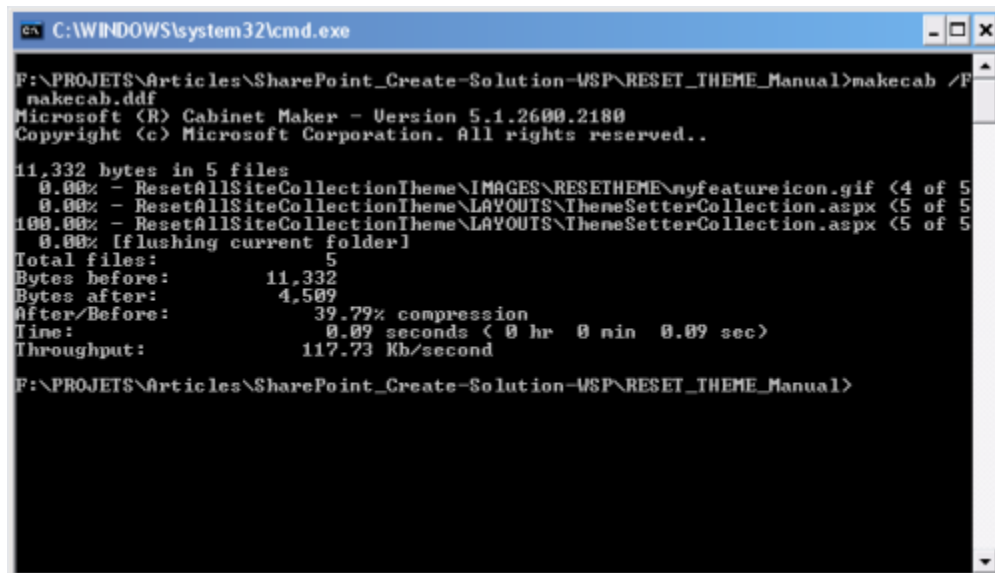
    source      File to compress.
    destination File name to give compressed file. If omitted, the
                  last character of the source file name is replaced
                  with an underscore (_) and used as the destination.

    /F directives A file with MakeCAB directives (may be repeated).
    /D var=value  Defines variable with specified value.
    /L dir        Location to place destination (default is current directory).
    /U{nl}        Verbosity level (1..3).

F:\PROJETS\Articles\SharePoint_Create-Solution-WSP\RESET_THEME_Manual>
```

Il suffit dès lors d'exécuter la commande MAKECAB avec les paramètres adaptés :

- **makecab.exe /F makecab.ddf**

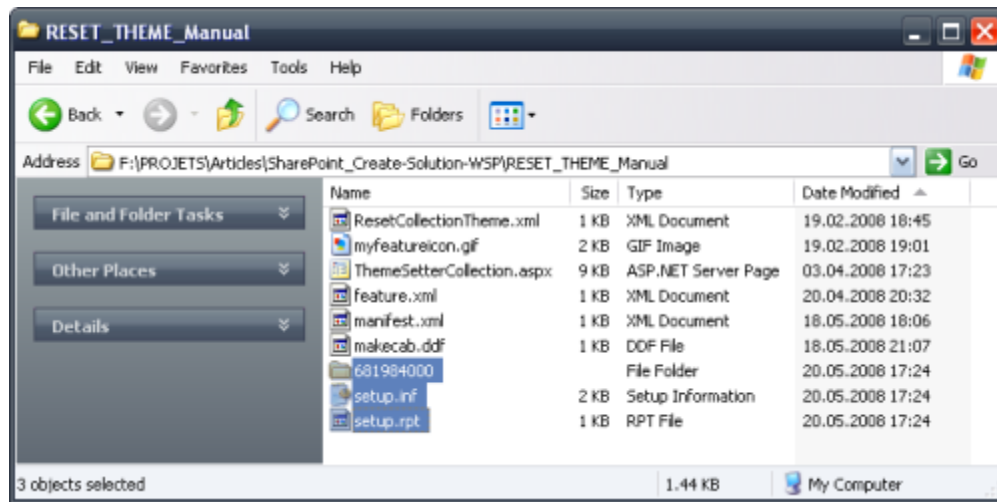


```
C:\WINDOWS\system32\cmd.exe
F:\PROJETS\Articles\SharePoint_Create-Solution-WSP\RESET_THEME_Manual>makecab /F
makecab.ddf
Microsoft (R) Cabinet Maker - Version 5.1.2600.2180
Copyright (c) Microsoft Corporation. All rights reserved..

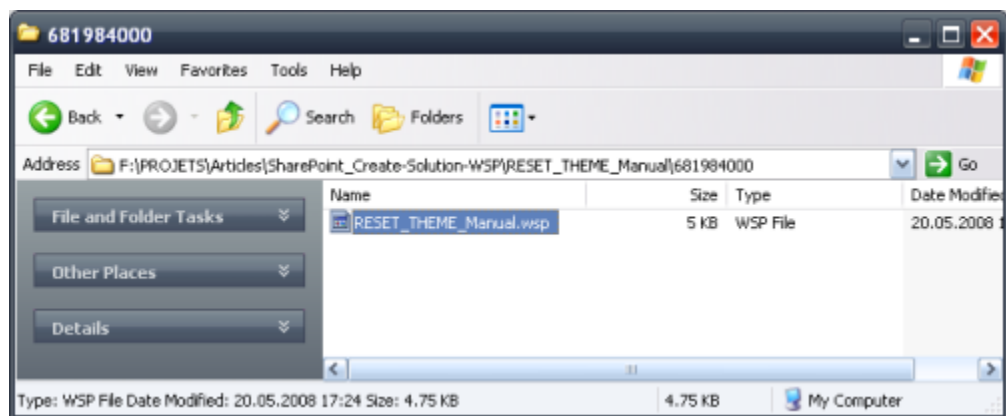
11,332 bytes in 5 files
  0.00% - ResetAllSiteCollectionTheme\IMAGES\RESETTHEME\myfeatureicon.gif <4 of 5
  0.00% - ResetAllSiteCollectionTheme\LAYOUTS\ThemeSetterCollection.aspx <5 of 5
100.00% - ResetAllSiteCollectionTheme\LAYOUTS\ThemeSetterCollection.aspx <5 of 5
  0.00% [flushing current folder]
Total files:      5
Bytes before:     11,332
Bytes after:      4,509
After/Before:     39.79% compression
Time:             0.09 seconds < 0 hr  0 min  0.09 sec>
Throughput:       117.73 Kb/second

F:\PROJETS\Articles\SharePoint_Create-Solution-WSP\RESET_THEME_Manual>
```

On retrouve alors des documents de rapports pour cette exécution.



Le fichier WSP est placé dans un sous-répertoire.



On peut dès lors utiliser ce fichier pour installer notre composant sur la ferme.

Mais voyons avant cela comment effectuer ce même travail de packaging avec un outil externe : WSPBuilder.

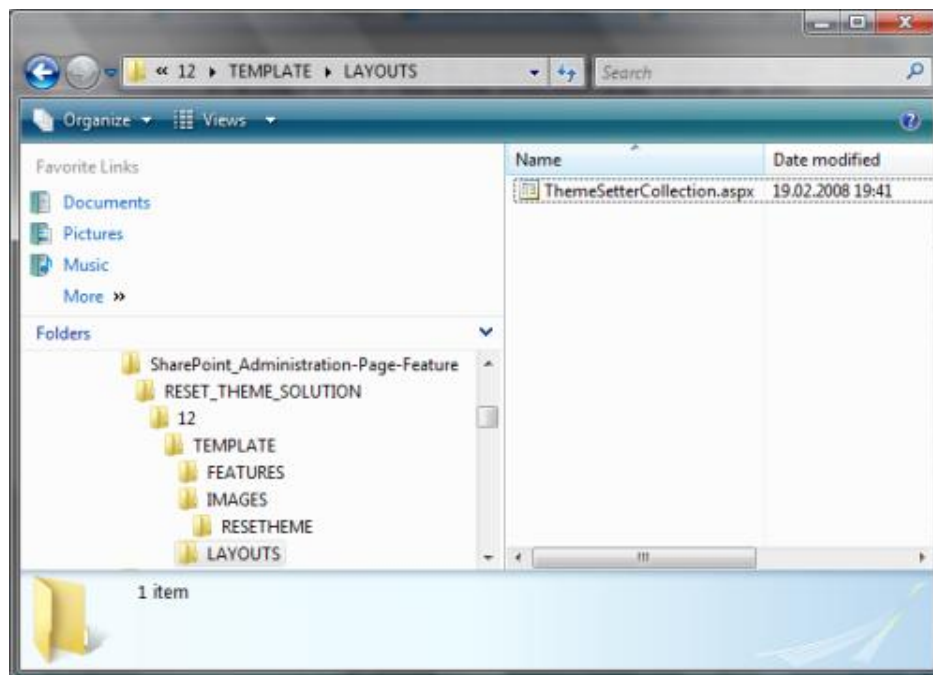
Outil d'automatisation : WSPBuilder

WSPBuilder est un outil très simple, mais très puissant permettant de créer le fichier de solution en s'affranchissant de toutes les étapes manuelles. Cet outil est développé par un développeur danois : **Carsten Keutmann**

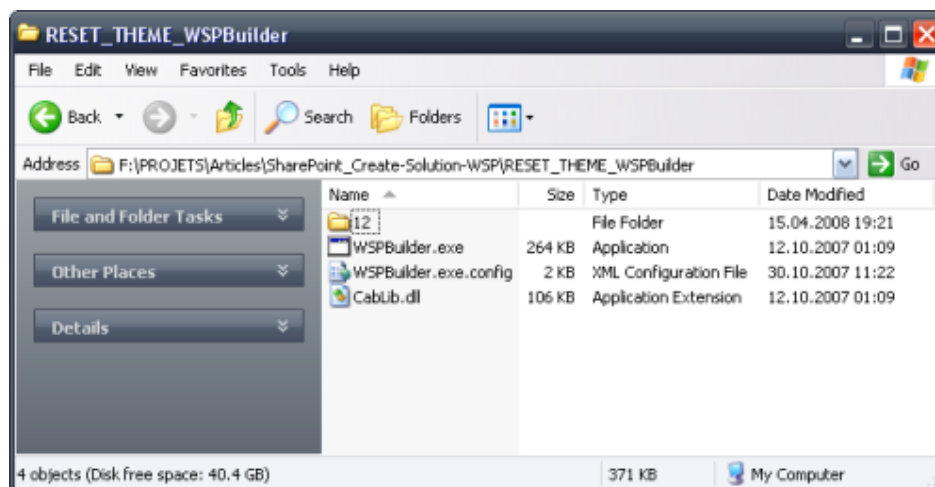
On peut retrouver le site de ce projet sur **CODEPLEX** :

- **CODEPLEX - WSPBuilder**

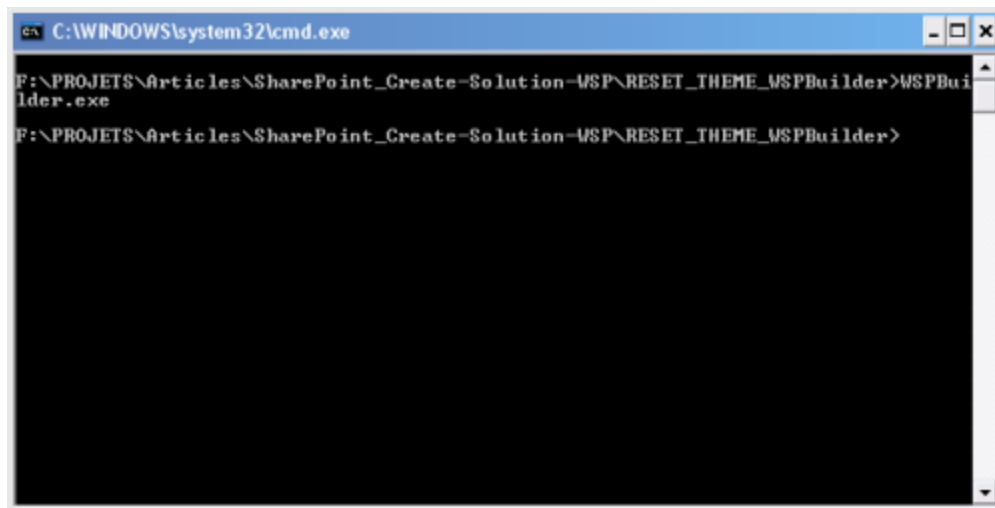
Pour notre exemple de développement, la mise en place de WSPBuilder est très simple, il suffit de reproduire dans un répertoire de travail la structure du 12 de SharePoint avec nos fichiers placés dans les répertoires adéquat :



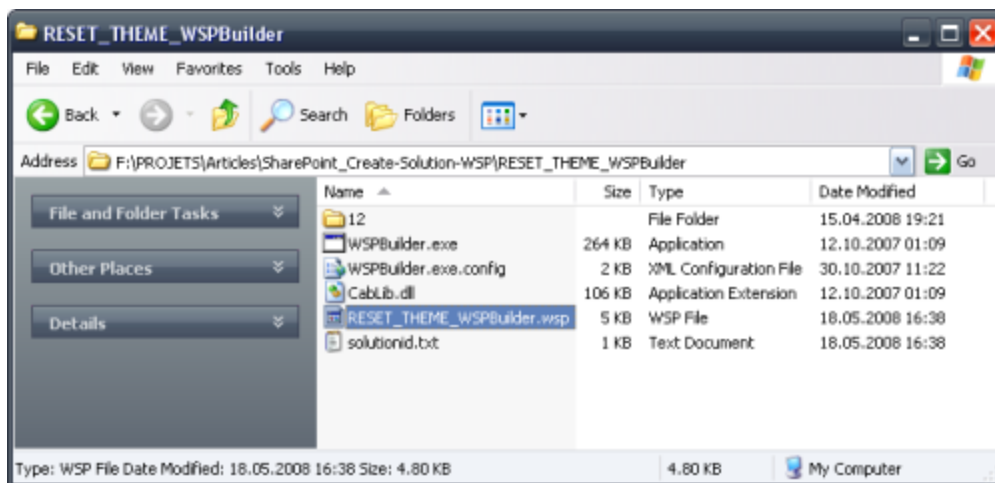
Maintenant, il nous faut utiliser l'exécutable de l'outil "WSPBuilder.exe" qui doit être placé au même niveau que le répertoire 12 de notre projet :



Il suffit alors d'exécuter ce fichier pour que la génération du fichier de solution s'effectue :



On obtient alors le fichier WSP possédant le nom du répertoire dans lequel nous sommes :



Pour de plus amples informations, je vous invite à aller sur [le site du projet](#).

Nous pouvons donc utiliser directement ce fichier pour installer le composant dans notre ferme.

Voyons maintenant comment faire cela.

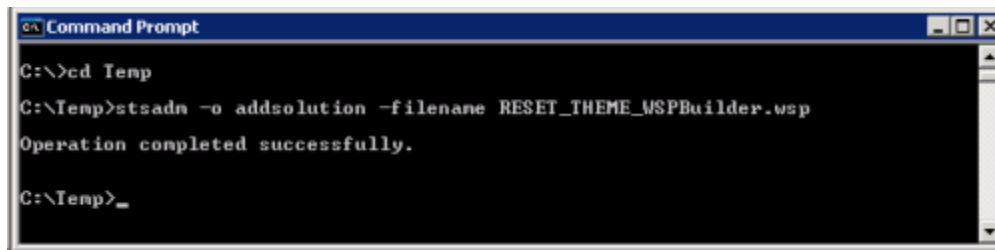
Déploiement de la solution

Nous partons dans cette partie du fichier de solution créé précédemment avec WSPBuilder (RESET_THEME_WSPBuilder.wsp).

Il faut tout d'abord copier le fichier WSP dans un répertoire de travail d'un des serveurs SharePoint de notre ferme (le plus simple est de prendre un des frontaux web).

On ouvre alors une commande DOS pour utiliser l'outil de gestion de SharePoint **STSADM.EXE** avec une opération spécifique :

- **stsadm -o addsolution -filename RESET_THEME_WSPBuilder.wsp**



```
CA\ Command Prompt
C:\>cd Temp
C:\Temp>stsadm -o addsolution -filename RESET_THEME_WSPBuilder.wsp
Operation completed successfully.
C:\Temp>_
```

Cette opération permet d'ajouter le fichier de déploiement dans la ferme SharePoint, en revanche il ne déploie pas cette solution.

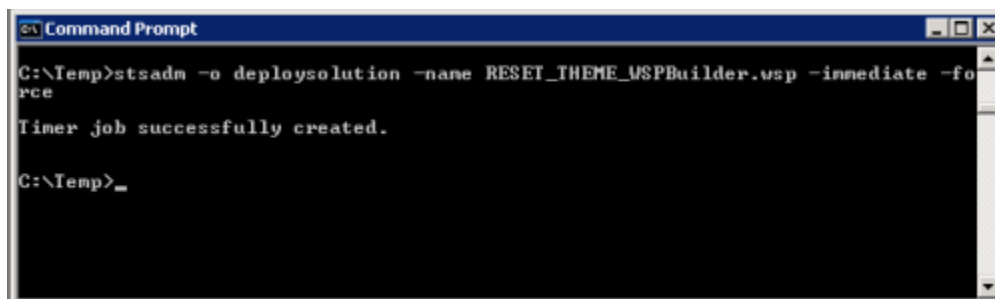
Pour cela, il faut utiliser une autre commande permettant de demander au Timer de SharePoint d'effectuer ce déploiement avec les différentes copies ou modifications.

Cette partie peut se faire avec la commande STSADM ou avec l'interface d'administration WEB.

Avec STSADM

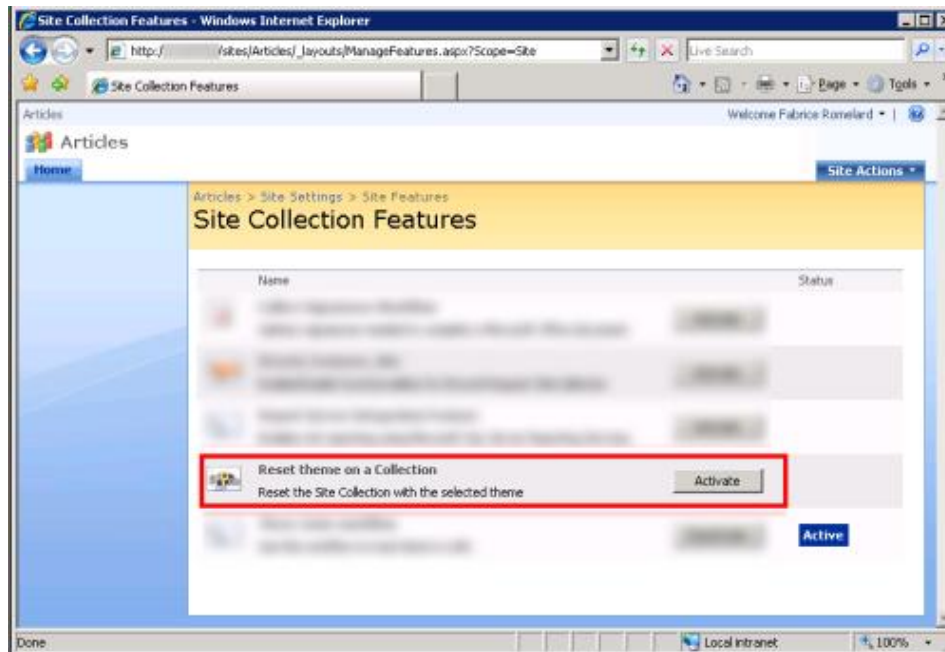
N'ayant pas de DLL dans notre composant mais juste quelques fichiers à placer, il suffit d'exécuter la commande suivante :

- **stsadm -o deploysolution -name RESET_THEME_WSPBuilder.wsp -immediate -force**



```
CA\ Command Prompt
C:\Temp>stsadm -o deploysolution -name RESET_THEME_WSPBuilder.wsp -immediate -force
Timer job successfully created.
C:\Temp>_
```

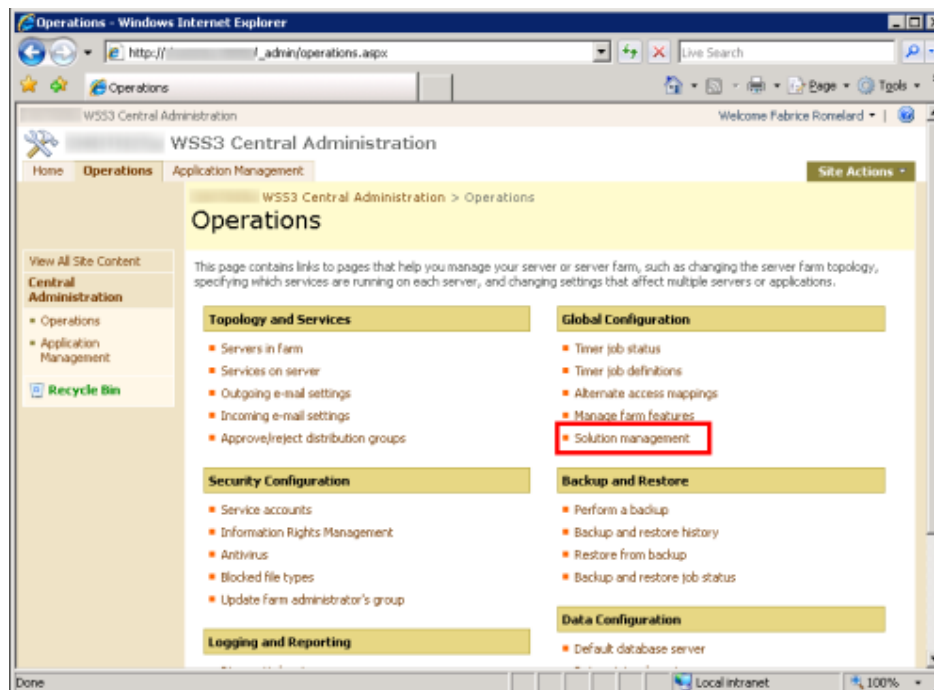
On peut alors aller dans les composants disponibles pour la collection de sites (Sites Settings > Site Collection Features) afin de trouver notre composant activable.



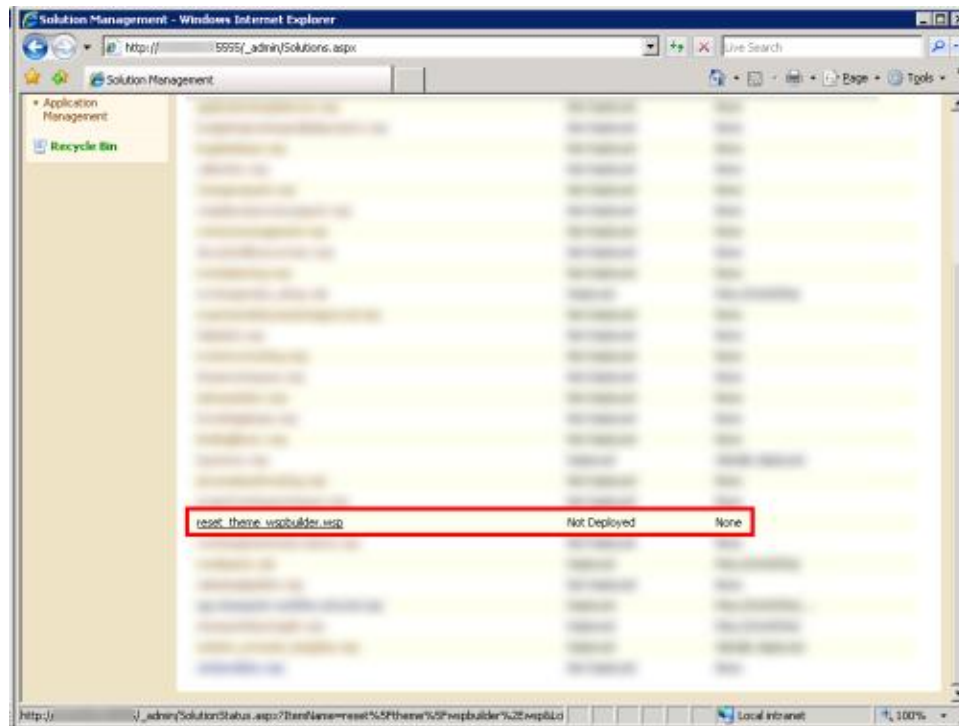
Avec la Central Administration

La Central Admin est le site Web d'administration de toute ferme SharePoint. On peut donc depuis ce site WEB déployer ou rétracter les solutions SharePoint ajoutées dans notre ferme.

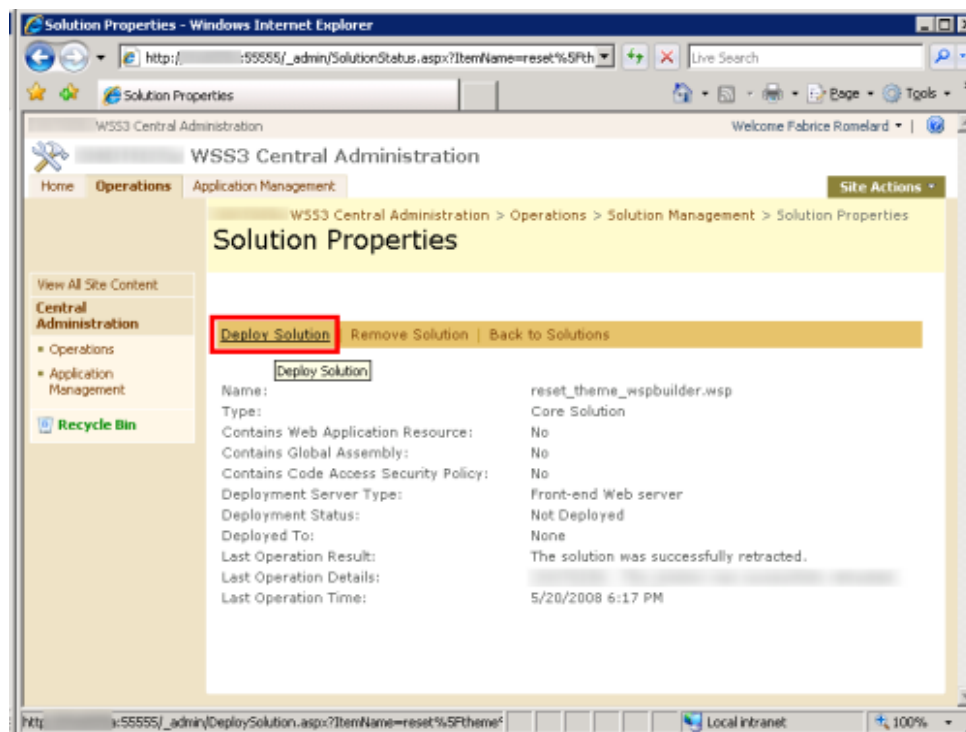
Pour effectuer cela, il faut aller dans la rubrique "Operations" de ce site et cliquer sur "Solution Management"



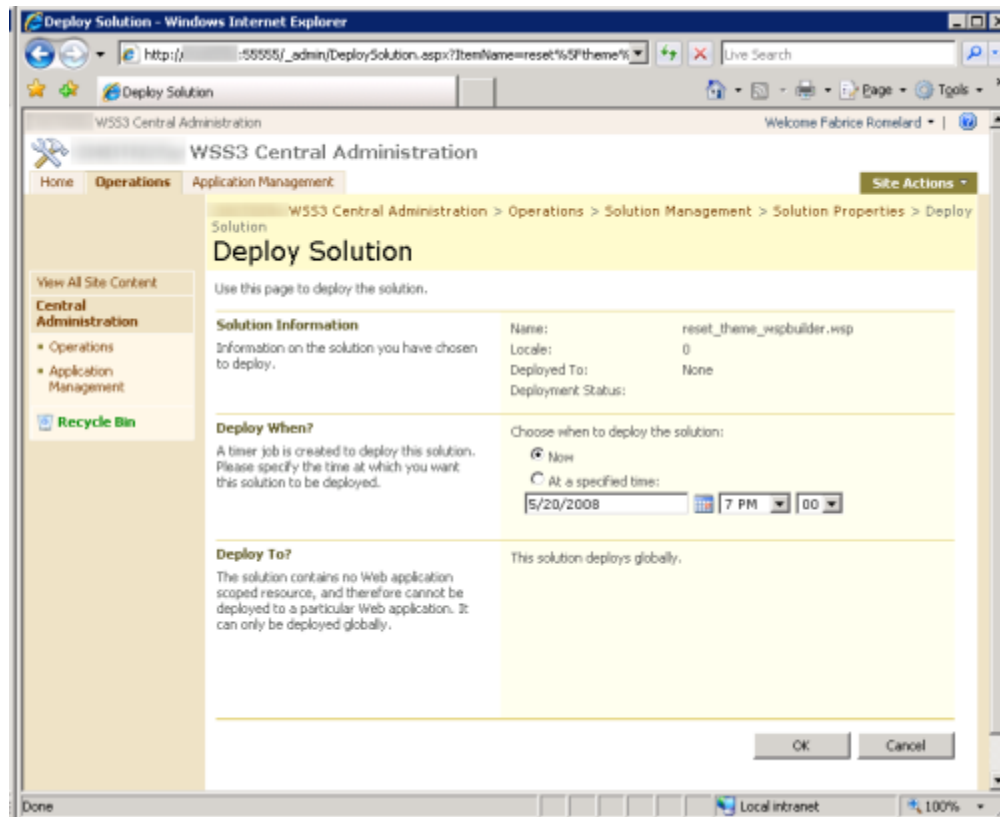
On retrouve alors l'ensemble des solutions déployées (ou non) sur notre ferme, dont celle que nous avons uploadée



On peut cliquer sur notre solution pour obtenir plus de détails sur cette solution ainsi que les boutons de déploiement ou de suppression. Nous allons cliquer sur "Deploy Solution".



Du fait que notre solution est très simple, nous avons juste le moment voulu pour déployer notre solution à définir "Now" et cliquer sur OK.



La tâche est alors créée et notre composant sera activable sur les collections de sites.

Voyons ce qu'il faut faire pour supprimer une solution.

Suppression d'une solution

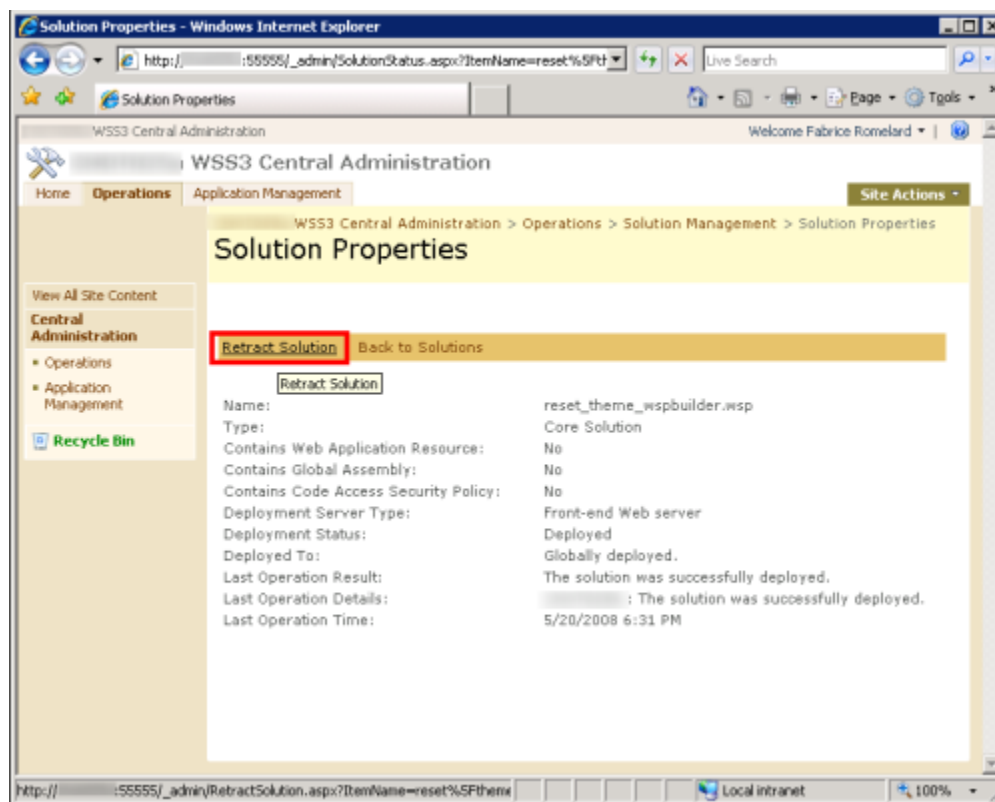
La suppression d'une solution implique forcément la désinstallation des composants de cette solution. Ceci est effectivement simple pour notre exemple.

En revanche, lorsque ce composant contient une WebPart ou des fichiers qui seront uploadés dans les listes (WebParts, Site Templates, Page Master, ...), il faut penser à ces fichiers qui peuvent devenir inutilisables (comme pour les WebParts).

Pour notre cas, nous avons juste à supprimer depuis la Central Administration ou en commande.

Avec la Central Administration

Il faut aller dans la même page de listing des solutions et cliquer sur notre solution pour avoir le détail et les boutons de suppression ou retrait.



On définit alors de la même façon le moment que l'on souhaite pour retirer cette solution et on clique sur OK.

WSS3 Central Administration

Home Operations Application Management Site Actions

WSS3 Central Administration > Operations > Solution Management > Solution Properties > Retract Solution

Retract Solution

Use this page to retract the solution.

Solution Information Information on the solution you have chosen to retract.	Name: reset_theme_wspbuilder.wsp Locale: 0 Deployed To: Globally deployed. Deployment Status:
Retract When? A timer job is created to retract this solution. Please specify the time at which you want this solution to be retracted.	Choose when to retract the solution: <input checked="" type="radio"/> Now <input type="radio"/> At a specified time: 5/20/2008 7 PM 00
Retract From? The solution contains no Web application scoped resource, and therefore cannot be retracted from a particular web application. It can only be retracted globally.	This solution should be retracted globally.

OK Cancel

Il ne reste alors que d'attendre le moment défini et ensuite de cliquer sur cette solution pour cette fois cliquer sur "Remove Solution".

WSS3 Central Administration

Home Operations Application Management Site Actions

WSS3 Central Administration > Operations > Solution Management > Solution Properties

Solution Properties

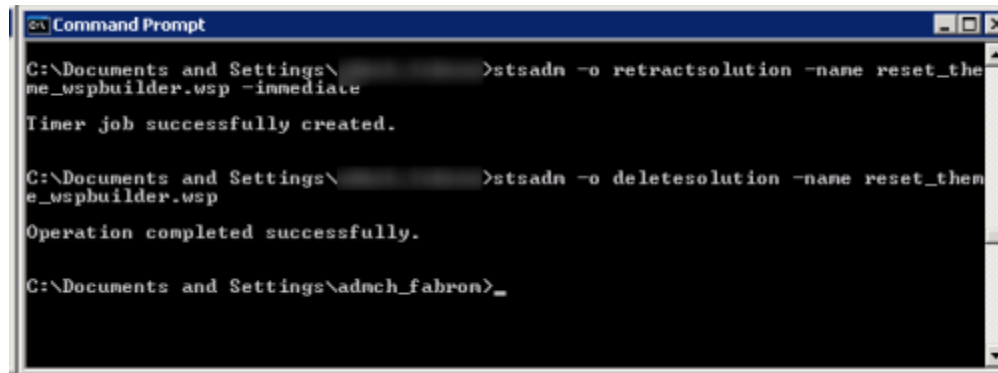
Deploy Solution Remove Solution Back to Solutions

Name:	reset_theme_wspbuilder.wsp
Type:	Core Solution
Contains Web Application Resource:	No
Contains Global Assembly:	No
Contains Code Access Security Policy:	No
Deployment Server Type:	Front-end Web server
Deployment Status:	Not Deployed
Deployed To:	None
Last Operation Result:	The solution was successfully retracted.
Last Operation Details:	The solution was successfully retracted.
Last Operation Time:	5/20/2008 6:45 PM

En ligne de commande

La ligne de commande est simple, puisqu'elle correspond à retirer la solution et supprimer celle-ci de la liste. On traduit ça avec deux opérations STSADM :

- stsadm -o retractsolution -name reset_theme_wspbuilder.wsp -immediate
- stsadm -o deletesolution -name reset_theme_wspbuilder.wsp



```
Command Prompt
C:\Documents and Settings\adnch_fabron>stsadm -o retractsolution -name reset_theme_wspbuilder.wsp -immediate
Timer job successfully created.

C:\Documents and Settings\adnch_fabron>stsadm -o deletesolution -name reset_theme_wspbuilder.wsp
Operation completed successfully.

C:\Documents and Settings\adnch_fabron>
```

Notre solution est maintenant supprimée de la ferme.

Conclusion

Cet article nous a permis au travers d'un exemple simple de comprendre comment transformer son développement en réel composant au sens SharePoint 2007. Cela nous permet de nous affranchir de la structure de la ferme et de laisser celle-ci effectuer toutes les tâches nécessaires pour rendre notre composant utilisable.

Cette architecture logicielle aussi primordiale pour assurer la continuité de service en cas d'évolution de l'architecture physique. Ainsi, on peut ajouter un nouveau serveur Frontal WEB dans notre ferme et celle-ci se chargera automatiquement de déployer les fichiers nécessaires pour chaque solution déployée dans celle-ci.

Bien que non parfaite, le produit fourni par Carsten Keutmann : **WSPBuilder** permet de créer les solutions dans une grande partie des cas rencontrés. En revanche, dans certaines situations (surtout avec certaines DLL), l'outil peut devenir délicat à manipuler. Je vous invite donc à suivre ce projet, car il évolue constamment.

Voici quelques liens utiles si cet article vous a intéressé :

- [Creating a Solution \(US\)](#)
 - [Creating a SharePoint Solution Package \(.wsp\) in 5 steps \(US\)](#)
 - [WSPBuilder \(US\)](#)
 - [SharePoint 2007 : Features et Solutions, les outils de Création et de Déploiement \(partie 2\) \(FR\)](#)
 - [Installation de WSS V3 \(FR\)](#)
 - [Office Online](#)
 - [Club SPS MOSS FRANCE\(FR\)](#)
-

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F____)
Intranet/Extranet CTO - **SGS**