

# Créer un Site Template pour WSS V3

## Créer un modèle de site pour SharePoint



Nous avons vu dans le précédent article comment modifier la page maître d'un site SharePoint. Nous pouvons donc maintenant voir comment créer sur notre serveur un modèle de site (Site Template) SharePoint.

### Introduction

Nous avons vu dans les précédents articles comment installer une plateforme Windows SharePoint Services V3 (WSS) :

- **WSS V3 en stand-Alone**
- **Installation de WSS V3**

Nous avons ensuite abordé la question de la **page maître (master page)**. Nous pouvons donc maintenant passer au stade supérieur dans la découverte de cette plateforme.

Le concept de base de la technologie SharePoint est la création "industrielle" de sites WEB. Pour se faire, le système utilise des modèles qui permettent ensuite de créer des sites d'équipe, des Wiki, des Blogs, ...

Nous verrons donc comment créer un modèle de site et le mettre en place sur notre serveur.

---

### Présentation

Après avoir personnalisé notre site par la Master Page, nous voulons maintenant utiliser cette master dans un modèle que nous pourrions réutiliser ce modèle dans nos futures créations de site.

L'objectif de cet article est donc bien entendu d'expliquer comment créer un modèle de site à partir du modèle de base le plus utilisé :

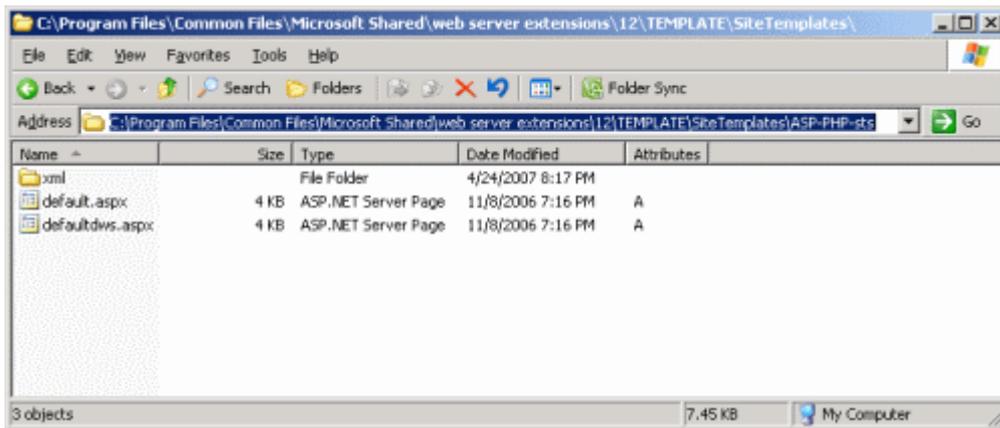
- Le site d'équipe (ou Team Site)

Vous pourrez par la suite voir pour ajouter des composants personnalisés que l'on souhaite inclure dès la création de son site. Ainsi, on finit donc par vouloir aller bien plus loin que les possibilités natives du produit.

Voyons donc la méthode de création de ce modèle de site (ou site template).



Nous allons donc commencer très simplement par faire une copie du répertoire "sts" (qui est le modèle pour les TeamSite) dans un autre répertoire "ASP-PHP-sts"



Nous voyons donc très bien à ce stade que nous avons les deux fichiers ASPX de base des TeamSites, ainsi qu'un sous-répertoire XML contenant un fichier :

- **ONET.XML**

Ce fichier est le fichier "différentiel" de définition de tout site SharePoint.

En effet dans WSS V3, Microsoft a affiné la notion d'industrialisation et de décomposition. Ainsi, on factorise un maximum d'information dans un répertoire de base :

- [ProgramFiles]\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\GLOBAL\

Celui-ci contient aussi le répertoire "XML" avec le fichier "ONET.XML" de base (il fait pour information 218 Ko).

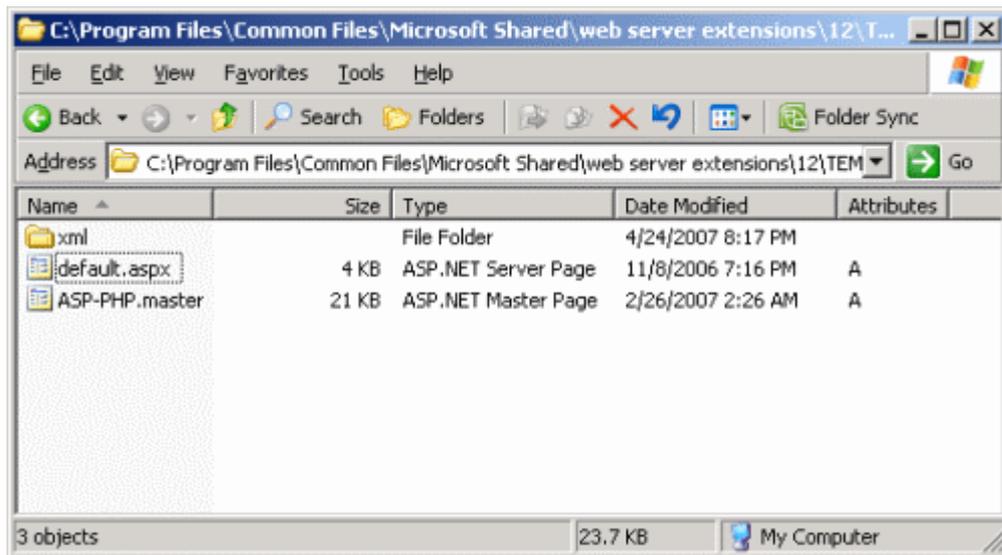
Il ne faut strictement jamais toucher au contenu de ce répertoire "GLOBAL". Si cela arrive, vous risquez deux choses :

- Que vos modifications soient écrasées lors de la mise en place de patches ou de Service Pack
- De perdre le support de Microsoft en cas de problème découvert

Revenons donc dans notre répertoire de travail "ASP-PHP-sts". Nous pouvons déjà copier à la racine le fichier master "ASP-PHP.master" de notre précédent article :

- **Modifier la Master Page de SharePoint**

Nous pouvons aussi supprimer le fichier "defaultdws.aspx", qui ne nous servira pas dans cet exemple. Nous avons donc maintenant deux fichiers (default.aspx, ASP-PHP.master) et un répertoire "XML" (contenant le fichier "ONET.XML").



Nous devons maintenant modifier notre fichier ONET.XML pour la gestion de cette nouvelle master page.

## Modification du fichier ONET.XML

Nous allons travailler sur ce fichier avec notre meilleur ami :

- Visual Studio .NET 2005

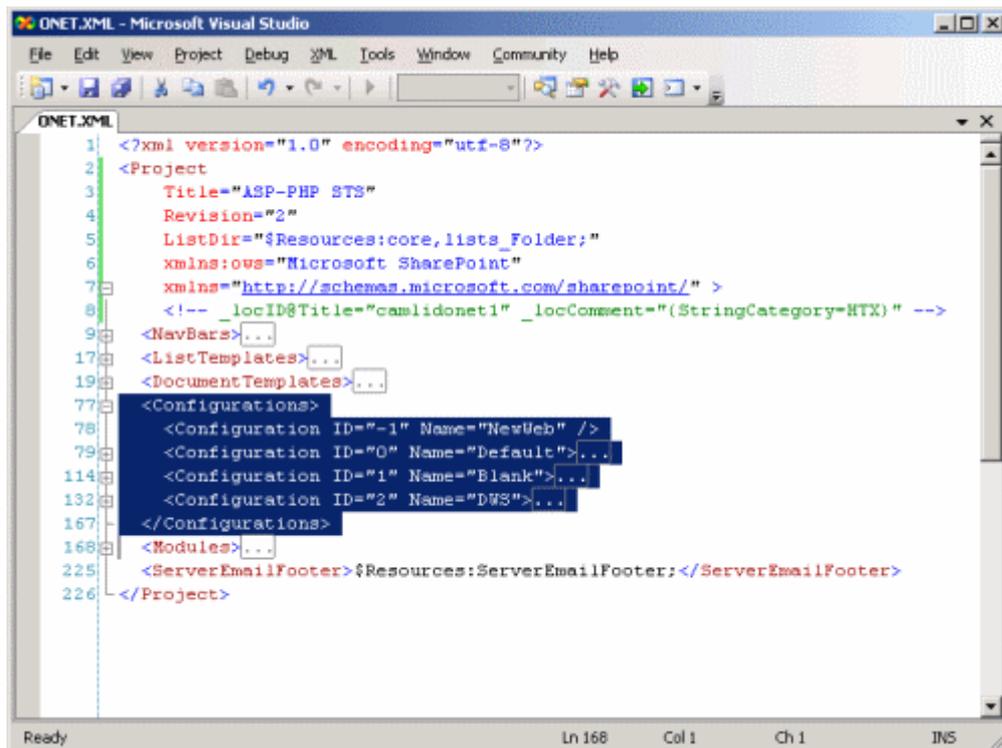
Une des premières choses à faire est d'ajouter dans la définition de l'entête XML le NameSpace de SharePoint afin de bénéficier du XSD de SharePoint (**wss.xsd**). Ce fichier est installé avec WSS V3 et se trouve dans le répertoire :

- [ProgramFiles]\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\XML\

Ainsi, nous modifierons le nœud de base "Project" comme suit :

```
<Project
  Title="ASP-PHP STS"
  Revision="2"
  ListDir="$Resources:core,lists_Folder;"
  xmlns:ows="Microsoft SharePoint"
  xmlns="http://schemas.microsoft.com/sharepoint/" >
```

Nous laisserons tous les nœuds suivant jusqu'au nœud "Configurations" :



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project
3   Title="ASP-PHP STS"
4   Revision="2"
5   ListDir="$Resources:core,lists_Folder;"
6   xmlns:ows="Microsoft SharePoint"
7   xmlns="http://schemas.microsoft.com/sharepoint/" >
8   <!-- _locID@Title="camlidonet1" _locComment="(StringCategory=MTX)" -->
9   <NavBars>...
17  <ListTemplates>...
19  <DocumentTemplates>...
77  <Configurations>
78    <Configuration ID="-1" Name="NewWeb" />
79    <Configuration ID="0" Name="Default">...
114   <Configuration ID="1" Name="Blank">...
132   <Configuration ID="2" Name="DWS">...
167 </Configurations>
168 <Modules>...
225 <ServerEmailFooter>$Resources:ServerEmailFooter;</ServerEmailFooter>
226 </Project>
```

Dans ce nœud, nous voyons plusieurs sous-nœud "Configuration" qui sont différenciés par leur ID. Nous pouvons donc déjà supprimer les nœuds ayant l'ID 1 et 2 puis modifier l'ID du 0 avec une autre valeur (dans notre exemple, nous prendrons "69"), nous laisserons le "-1" :

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <Project
3     Title="ASP-PHP STS"
4     Revision="2"
5     ListDir="$Resources:core,lists_Folder;"
6     xmlns:ows="Microsoft SharePoint"
7     xmlns="http://schemas.microsoft.com/sharepoint/" >
8     <!-- _locID@Title="camlidonet1" _locComment="(StringCategory=HTX)" -->
9     <NavBars>...
10
11
12
13
14
15
16
17 <ListTemplates>...
18
19 <DocumentTemplates>...
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69 <Configuration ID="69" Name="Default">...
70
71
72
73
74
75
76
77 </Configurations>
78
79 <Configuration ID="-1" Name="NewWeb" />
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114 </Configurations>
115 <Modules>...
116
117
118
119
120
121
122 <ServerEmailFooter>$Resources:ServerEmailFooter;</ServerEmailFooter>
123 </Project>

```

A ce stade, nous devons modifier ce nœud "Configuration" avec les nouveaux paramètres à prendre en compte, notamment la spécification de la master page.

Les sous-nœuds de ce nœud "configuration" définissent les composants à créer ou activer lors de la création d'un site utilisant ce nouveau modèle. On retrouve donc :

- Les listes (Nœud "Lists") : Cela spécifie toutes les listes à créer (Documents, contacts, Taches, ...), on peut d'ailleurs voir les listes qui sont créées par défaut pour un Team Site (Document, Discussion, Contact, Annonce, Calendrier et Tache)
- Les Fichiers (Nœud "Modules") : Ceci pointe vers un autre nœud "Modules" qui liste les fichiers à uploader lors de la création
- Les "SiteFeatures" (Nœud "SiteFeatures") : Ce sont les composants SharePoint (les Features) à activer au niveau du site, on y retrouve par exemple le WorkFlow 3 états de base des TeamSites et la WebPart de base. Elle sont identifiées par le FeatureID qui est le GUID généré lors de son développement.
- Les "WebFeatures" (Nœud "WebFeatures") : Ce sont les composants SharePoint à activer au niveau du site Web créé

Dans notre exemple, nous modifierons la définition de la configuration afin de définir la master page à utiliser :

```

<Configuration
  ID="69"
  Name="Default"
  MasterUrl="_catalogs/masterpage/ASP-PHP.master">

```

Puis nous ajouterons un module qui pointerait vers le fichier Master :

```

<Modules>
  <Module Name="Default" />
  <Module Name="ASP-PHP-MasterPage" />
</Modules>

```

Nous laisserons toutes les listes et features comme elles sont de base.  
Nous pouvons passer au dernier nœud "Modules" afin de définir les fichiers qui devront être uploadés dès la création de notre site. Nous pouvons déjà supprimer les modules d'origine :

- DefaultBlank
- DWS

Nous laissons celui de base "Default" et ajoutons un nouveau "ASP-PHP-MasterPage".

```
<Module
  Name="ASP-PHP-MasterPage"
  List="116"
  Url="_catalogs/masterpage"
  RootWebOnly="FALSE">
  <File Url="ASP-PHP.master"
    Type="GhostableInLibrary"
    IgnoreIfExists="TRUE" />
</Module>
```

A ce stade, notre fichier ONET.XML est correct, et nous pouvons maintenant créer un fichier XML qui référence notre modèle de site :

- WEBTEMP\_xxxx.XML

Voyons comment faire cela.

## Création du WEBTEMP

Lors du chargement de SharePoint, celui-ci parcourt un répertoire spécifique (suivant votre Language ID) :

- [ProgramFiles]\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\LANG-ID\XML\

Ce répertoire contient notamment plusieurs fichiers XML ayant le nom commençant par WEBTEMP. Nous allons donc créer le notre :

- WEBTEMP\_ASP-PHP-ST.S.XML

Nous devons définir d'un nom pour ce modèle et surtout un ID, celui-ci sera le numéro d'identification pour SharePoint de votre modèle, il ne doit donc pas être en conflit avec d'autres déjà existant. La règle est donc de prendre une valeur supérieure à 10 000 (afin de ne pas entrer en conflit avec ceux que peuvent distribuer Microsoft).

Nous prendrons donc dans cet exemple "69690" et le nom "ASP-PHP-sts". Nous aurons aussi créé un sous-répertoire "ASP-PHP" dans le répertoire de stockage des images :

- [ProgramFiles]\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\IMAGES\

Dans ce dossier, nous ajoutons directement une image qui deviendra le logo de notre Modèle de site. Nous devons maintenant compléter notre fichier XML comme suit.

```
<?xml version="1.0" encoding="utf-8"?>
<Templates
  xmlns:ows="Microsoft SharePoint">
  <Template Name="ASP-PHP-sts" ID="69690">
    <Configuration ID="69"
      Title="ASP-PHP Team Site"
      Hidden="FALSE"
      ImageUrl="/_layouts/images/ASP-PHP/ASP-PHP-Logo.png"
      Description="ASP-PHP Team Site"
      DisplayCategory="ASP-PHP" >
    </Configuration>
  </Template>
</Templates>
```

Nous retrouvons dans le nœud configuration le Numéro (69) de configuration que nous avons fourni précédemment dans le fichier ONET.XML.

Nous pouvons maintenant lancer la commande "iisreset" afin que SharePoint recharge toute sa configuration.

Nous pouvons désormais utiliser ce modèle pour créer un nouveau site.

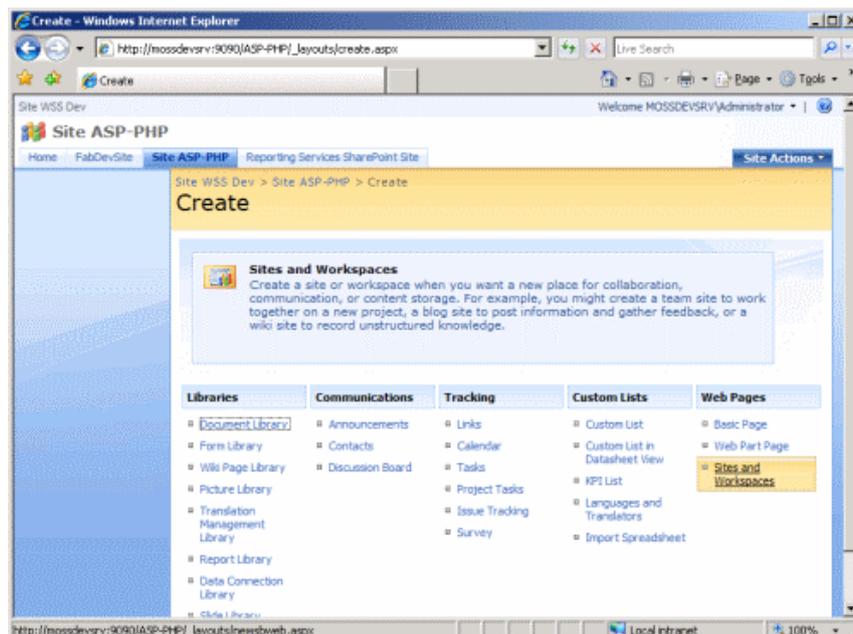
## Création d'un site utilisant ce nouveau modèle

A partir de maintenant depuis n'importe quel site de notre serveur SharePoint, nous pouvons créer un site utilisant ce nouveau modèle.

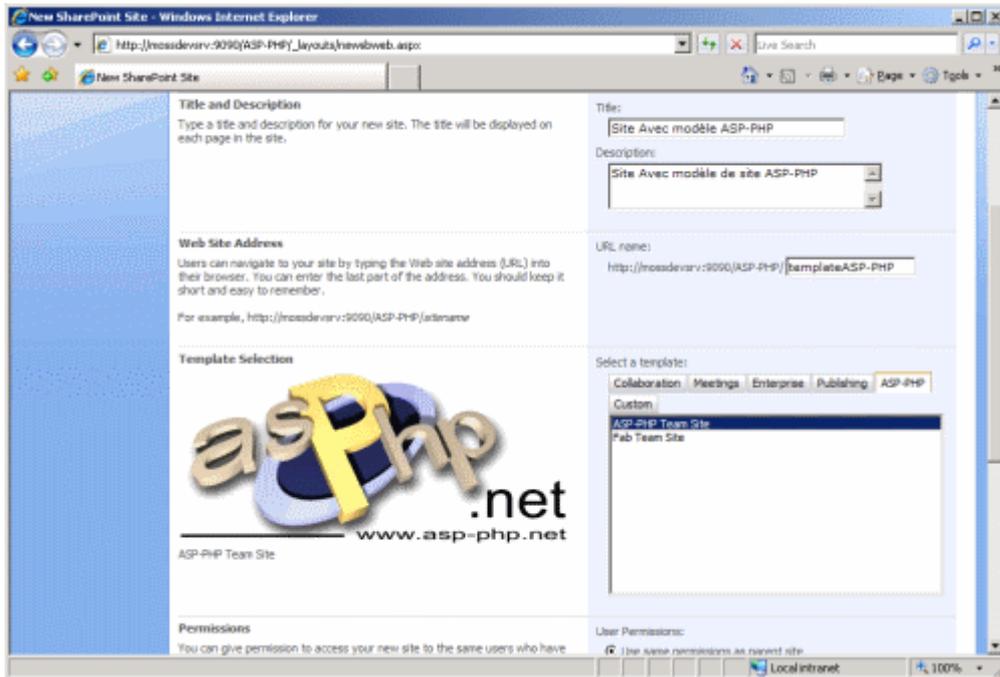
Pour s'en convaincre, nous allons en créer un. Ceci se fait avec le menu de droite "Site Actions" (on doit avoir les droits d'administration pour cela), puis "Create".



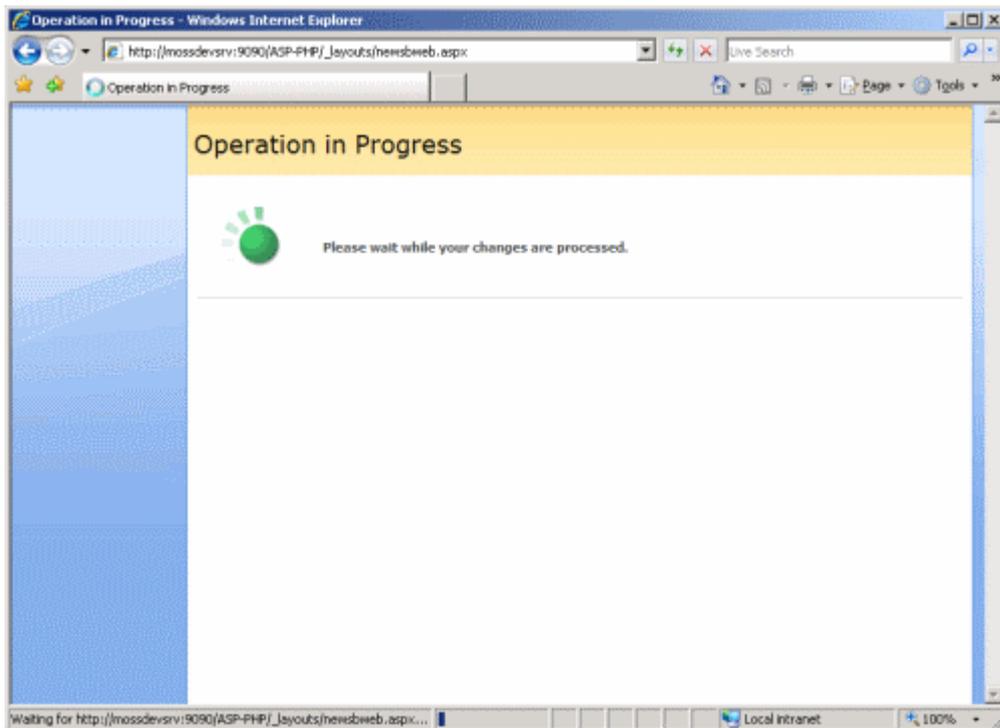
Dans le menu suivant on sélectionne "Sites and WorkSpaces".



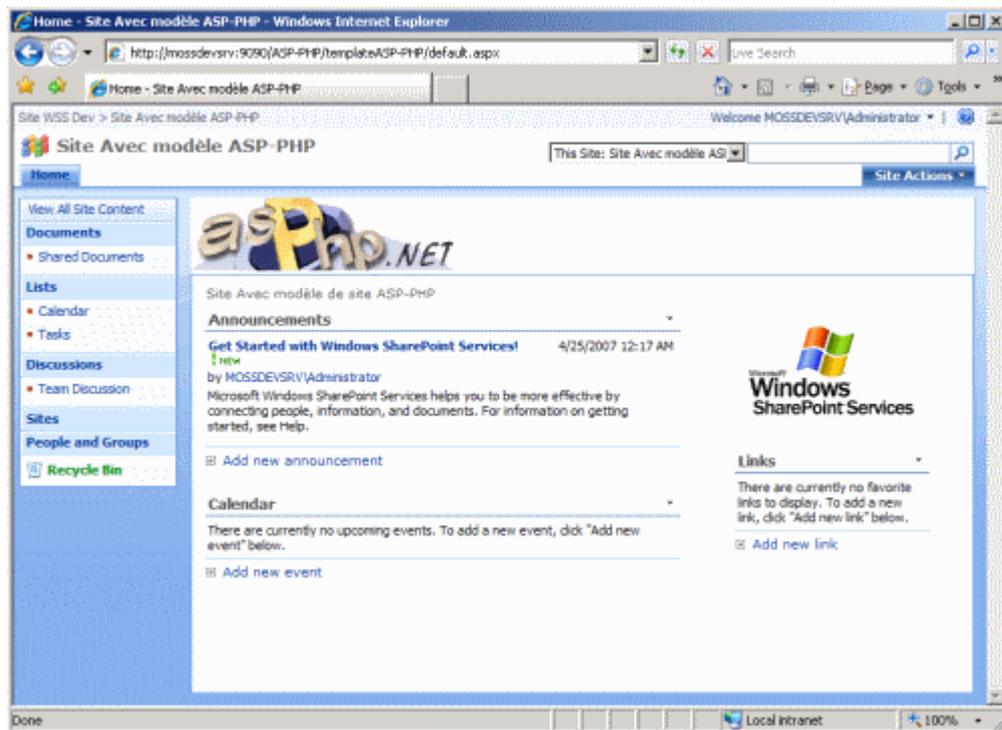
On donne alors les différents renseignements dans les champs et on sélectionne le modèle que l'on souhaite dans la catégorie "ASP-PHP".



On valide par "Create". On voit alors la création se faire.



Et on obtient enfin le nouveau site avec la master page sélectionnée.



Nous avons terminé cette création de modèle.

## Conclusion

Ce second article permet d'approcher une nouvelle vision du produit SharePoint. Cette présentation est valable aussi bien pour WSS V3 que MOSS 2007. Vous pouvez donc adapter votre plateforme SharePoint à votre besoin.

Vous pouvez aller bien plus loin que simplement définir une master page particulière, choisir des WebParts particulières, définir des listes personnalisées, ajouter des types de contenu spécifiques, ...

---

Voici quelques liens utiles si cet article vous a intéressé :

- [\*\*Welcome to the Windows SharePoint Services 3.0 SDK\*\*](#)
  - [\*\*Installation de WSS V3 \(FR\)\*\*](#)
  - [\*\*Le developpement avec Sharepoint MOSS 2007\*\*](#)
  - [\*\*Modifier la Master Page de SharePoint\*\*](#)
  - [\*\*SharePoint Server 2007 SDK: Software Development Kit and Enterprise Content Management Starter Kit\*\*](#)
  - [\*\*Windows SharePoint Services 3.0: Software Development Kit \(SDK\)\*\*](#)
  - [\*\*Windows SharePoint Services 3.0 Tools: Visual Studio 2005 Extensions\*\*](#)
  - [\*\*Club SPS MOSS FRANCE\(FR\)\*\*](#)
- 

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F\_\_\_)  
Consultant Technique **ilem SA**