

Les Delegate Controls dans SharePoint

Qu'est-ce que c'est ?

Les delegates controls sont une nouvelle fonctionnalité de MOSS 2007/WSS v3 et permettent l'affichage de User Controls asp.net 2.0 sur une page SharePoint. Dans cet article je propose de présenter le fonctionnement des Delegate controls, et d'expliquer la façon de les inclure sur une page d'un site SharePoint.

A quoi ça sert ?

Si je demande à toute personne ayant déjà développé une Webpart, le principal défaut, elle répondra spontanément qu'il n'existe pas de designer pour webpart dans Visual Studio, et que par conséquent, une webpart se code dynamiquement en créant les objets à la volée.

Dès lors, il s'avère très intéressant de pouvoir développer un User Control .Net 2.0 de façon traditionnelle, en y ajoutant des contrôles de manière graphique, et de le déployer ensuite sur un site SharePoint.

Comment ça marche ?

Tout l'intérêt des Delegate Controls réside dans le fait qu'il s'agit de simple User Controls asp.net, et que l'on puisse les déployer sur un site SharePoint à l'aide d'une feature.

Feature, « Le terme est lâché ! » », et c'est bien là tout l'intérêt. Grâce à la feature, on peut décider d'afficher ou pas un Delegate Control comprenant un User Control en activant ou en désactivant la feature, et ceci aussi bien au niveau Web, Site que Farm.

Et en pratique ?

Maintenant que nous avons vu à quoi servent les Delegate Controls, et comment ils fonctionnent, rien ne vaut une petite démonstration.

Dans cet exemple, nous allons créer un UserControl asp.net classique, et grâce à un Delegate Control, et à une feature, nous allons l'afficher sur la page d'accueil d'un site SharePoint.

1° Création d'un User control

Depuis Visual Studio 2005 nous créons d'abord un nouveau site web « Test ». En cliquant « Bouton de droite » sur le projet, nous sélectionnons ensuite « Ajouter un nouvel élément » puis « Contrôle utilisateur Web ». Nous remplissons alors le nom du fichier .ascx (par ex : MyUserControl.ascx) et désélectionnons l'option « Placer le code dans un fichier distinct » ; de cette façon le code se trouvera en ligne dans un seul fichier.

Nous commutons ensuite l'affichage en mode « Design » et plaçons dans notre contrôle, un bouton et une textbox. Après un double-click sur le bouton, nous saisissons le code dans l'événement « Click » du bouton. Le code final doit ressembler à ceci :

```

<%@ Control Language="C#" ClassName="WebUserControl" %>

<script runat="server">

    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox1.Text = DateTime.Now.ToString();
    }
</script>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Text="Button" />

```

Tableau 1 : Code du UserControl

Finalement nous pouvons inclure ce contrôle dans la page Default.aspx du site web afin de vérifier que tout fonctionne correctement.

2° Création d'un Delegate Control sur la page d'accueil d'un site SharePoint

A partir du répertoire des « Site Templates » (12/TEMPLATE/SiteTemplates) nous ouvrons la page default.aspx de notre site (Rappel : Ne JAMAIS modifier un template original, mais en faire une copie) et ajoutons la ligne suivante :

```

<SharePoint:DelegateControl runat="server" ControlId="MyUserControl" />

```

Tableau 2 : Création du Delegate Control

Cette balise correspond à l'emplacement du Delegate control dans la page. Nous le plaçons à un endroit qui ne dérange pas, par exemple au dessus des zones à webparts (aux environs de la ligne 56 du fichier default.aspx).

3° Création de la feature permettant le déploiement

Afin de réaliser cette feature de déploiement, nous créons 2 fichiers :

- Feature.xml : contenant les paramètres de la feature
- Elements.xml : contenant le nom du contrôle .ascx à utiliser

Une fois créés, ces fichiers devraient ressembler à ceci :

```

<?xml version="1.0" encoding="utf-8"?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/" Id="AF4CB471-
950A-4a68-A918-50FFD2703BE6" Scope="Web"
    Title="MySts.Demo.DelegateControls" Description="Description de
ma feature" Hidden="FALSE" Version="1.0.0.0">
    <ElementManifests>
        <ElementManifest Location="elements.xml"/>
    </ElementManifests>
</Feature>

```

Tableau 3 : Fichier Feature.xml

```

<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
    <Control Id="MyUserControl" Sequence="1000"
ControlSrc="~/_ControlTemplates/Custom/TestDelegate.ascx"/>
</Elements>

```

Tableau 4 : Fichier elements.xml

Ces deux fichiers devront être placés dans un sous-répertoire de « FEATURES » (12\TEMPLATE\FEATURES) du serveur SharePoint.

Quelques explications s'imposent :

- Dans le fichier « Feature.xml » l'Id est l'identifiant unique (GUID) de la feature. Dans Visual Studio 2005 il existe un outil permettant de générer des GUIDs. Dans le menu « Outil » sélectionner « Create GUID ». Un nouveau GUID est créé à chaque click sur « New GUID ». Il suffit ensuite de le copier dans le fichier XML
- Le « Scope » du fichier « Feature.xml » indique le niveau de visibilité de la feature. Comme mentionné précédemment il peut être « Web », « Site » ou « Farm »
- Le fichier elements.xml contient un tag « Control » qui permet de spécifier le User control à associer au Delegate control

Une fois la feature installée avec stsadm (stsadm -o installfeature -name feature), nous pouvons l'activer depuis le menu « Site features » du site SharePoint.

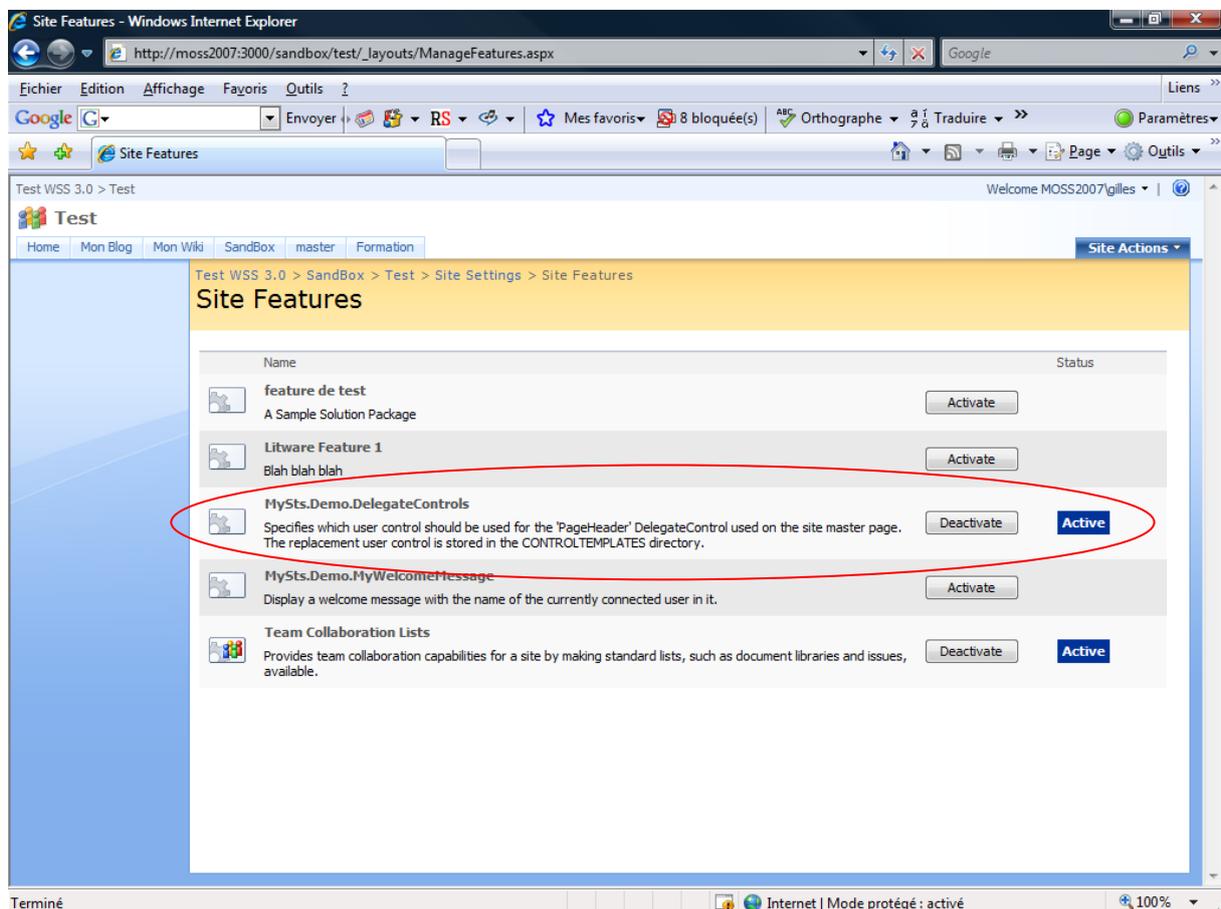


Figure 1 : Activation d'une feature depuis un site SharePoint

Lorsque la feature est activée nous pouvons retourner sur la page d'accueil du site et voir que notre Delegate control est apparu.

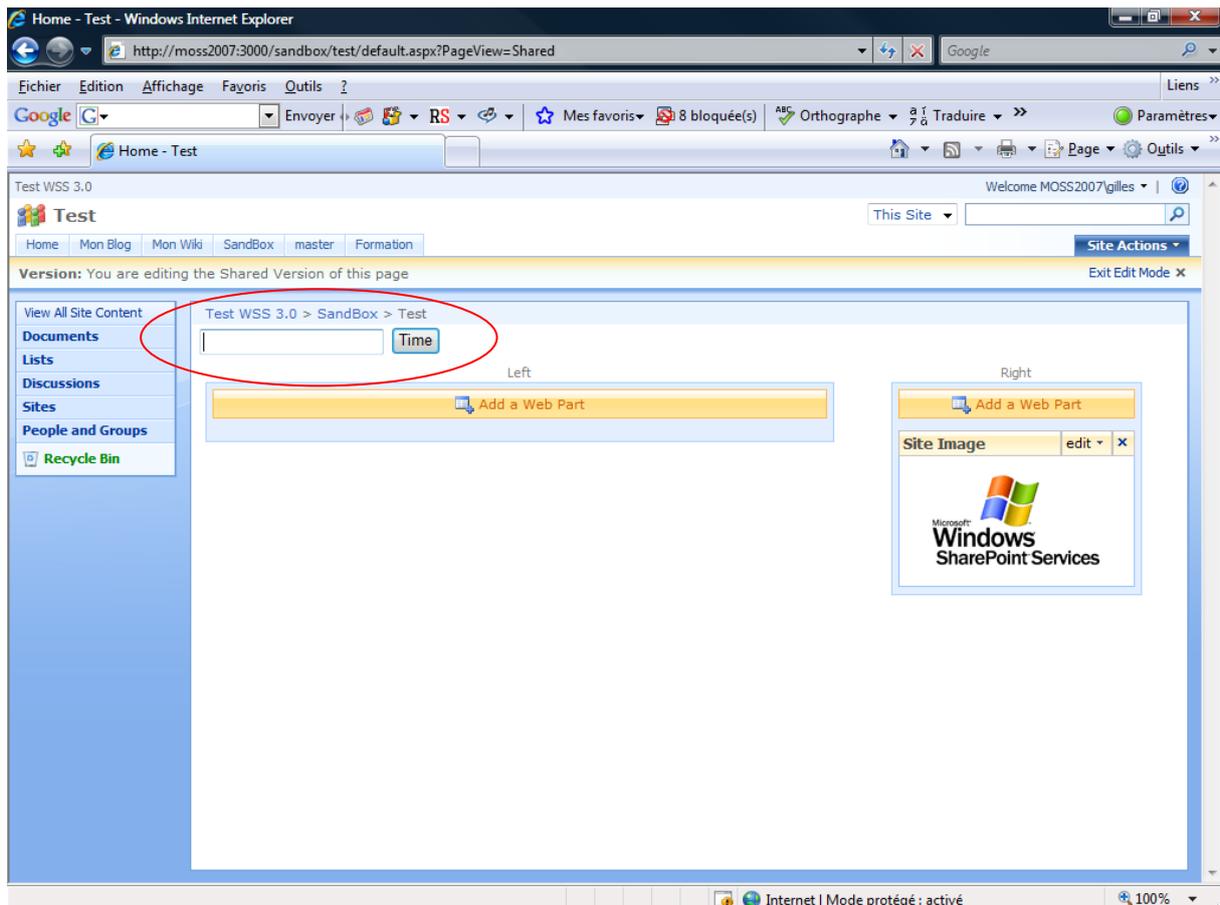


Figure 2 : Affichage du Delegate control sur la page d'accueil

Sur la page d'accueil de la Figure 2 en mode édition, nous pouvons constater qu'il ne s'agit pas d'une webpart qui aurait été insérée, mais bien d'un contrôle faisant partie intégrante de la page.

Surcharge des Delegate Controls

Une autre propriété des Delegate controls est le fait qu'ils peuvent se superposer lors du rendu d'affichage. Lorsque deux User controls différents utilisent le même Delegate control pour s'afficher, il existe un numéro de séquence qui permet d'identifier celui des deux qui s'affichera.

Si nous reprenons la feature que nous avons vu dans l'exemple précédent, dans le fichier « elements.xml » du Tableau 4, il existe le tag « Control » qui contient une propriété « Sequence ». Ce numéro de séquence permet de spécifier quel control va s'afficher. Plus le numéro de séquence est bas, plus la priorité est élevée.

Afin d'illustrer cet effet, nous allons créer un second User control que nous allons faire s'afficher dans le même Delegate control qu'auparavant mais avec un numéro de séquence plus bas.

4° Création d'un second User control

Ce User control est d'une simplicité enfantine, il prend le nom de l'utilisateur courant et l'affiche dans un message de bienvenue. Bien que très simple ce User control est intéressant car il permet de montrer comment utiliser les objets SharePoint dans nos contrôles.

```
<%@ Control Language="C#" ClassName="WelcomeMessage" %>
<%@ Register Tagprefix="SharePoint"
Namespace="Microsoft.SharePoint.WebControls"
Assembly="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="Utilities"
```

```

Namespace="Microsoft.SharePoint.Utilities" Assembly="Microsoft.SharePoint,
Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce11e9429c" %>
<%@ Import Namespace="Microsoft.SharePoint" %>

<script runat="server">

    public override void RenderControl(HtmlTextWriter writer)
    {
        try
        {
            SPWeb web = SPControl.GetContextWeb(Context);

            string userName = web.CurrentUser.Name;
            writer.Write("Welcome " + userName + ", have a nice day");
        }
        catch
        {
        }
        finally
        {
            base.RenderControl(writer);
        }
    }
}

</script>

```

Tableau 5 : Accès aux objets SharePoint depuis un User control

5° Création de la feature de déploiement

Comme dans l'exemple précédent nous devons créer 2 fichiers : Feature.xml et elements.xml.

```

<?xml version="1.0" encoding="utf-8"?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/" Id="F6B14481-
4755-4e1d-9A27-5EA23087787B" Scope="Web"
    Title="MySts.Demo.MyWelcomeMessage" Description="Display a
welcome message with the name of the currently connected user in it."
    Hidden="FALSE" Version="1.0.0.0">
    <ElementManifests>
        <ElementManifest Location="elements.xml"/>
    </ElementManifests>
</Feature>

```

Tableau 6 : Feature.xml

```

<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
    <Control Id="MyUserControl" Sequence="900"
ControlSrc="~/_ControlTemplates/Custom/WelcomeMessage.ascx"/>
</Elements>

```

Tableau 7 : elements.xml

Dans le fichier elements.xml le numéro de séquence (900) est inférieur à celui de l'exemple précédent, c'est lui par conséquent qui devrait s'afficher.

Après installation et activation de cette feature nous pouvons voir le résultat sur la page d'accueil de notre site.

The screenshot shows the 'Site Features' management interface. The browser address bar indicates the URL: `http://moss2007:3000/sandbox/test/_layouts/ManageFeatures.aspx`. The page title is 'Test WSS 3.0 > Test'. The breadcrumb trail is 'Test WSS 3.0 > SandBox > Test > Site Settings > Site Features'. The main heading is 'Site Features'. Below this, there is a table of features:

Name	Status
 feature de test A Sample Solution Package	Activate
 Litware Feature 1 Blah blah blah	Activate
 MySts.Demo.DelegateControls Specifies which user control should be used for the 'PageHeader' DelegateControl used on the site master page. The replacement user control is stored in the CONTROLTEMPLATES directory.	Deactivate Active
 MySts.Demo.MyWelcomeMessage Display a welcome message with the name of the currently connected user in it.	Deactivate Active
 Team Collaboration Lists Provides team collaboration capabilities for a site by making standard lists, such as document libraries and issues, available.	Deactivate Active

The status bar at the bottom of the browser shows 'Terminé', 'Internet | Mode protégé : activé', and '100%' zoom level.

Figure 3 : Activation de la feature avec un numéro de séquence plus bas

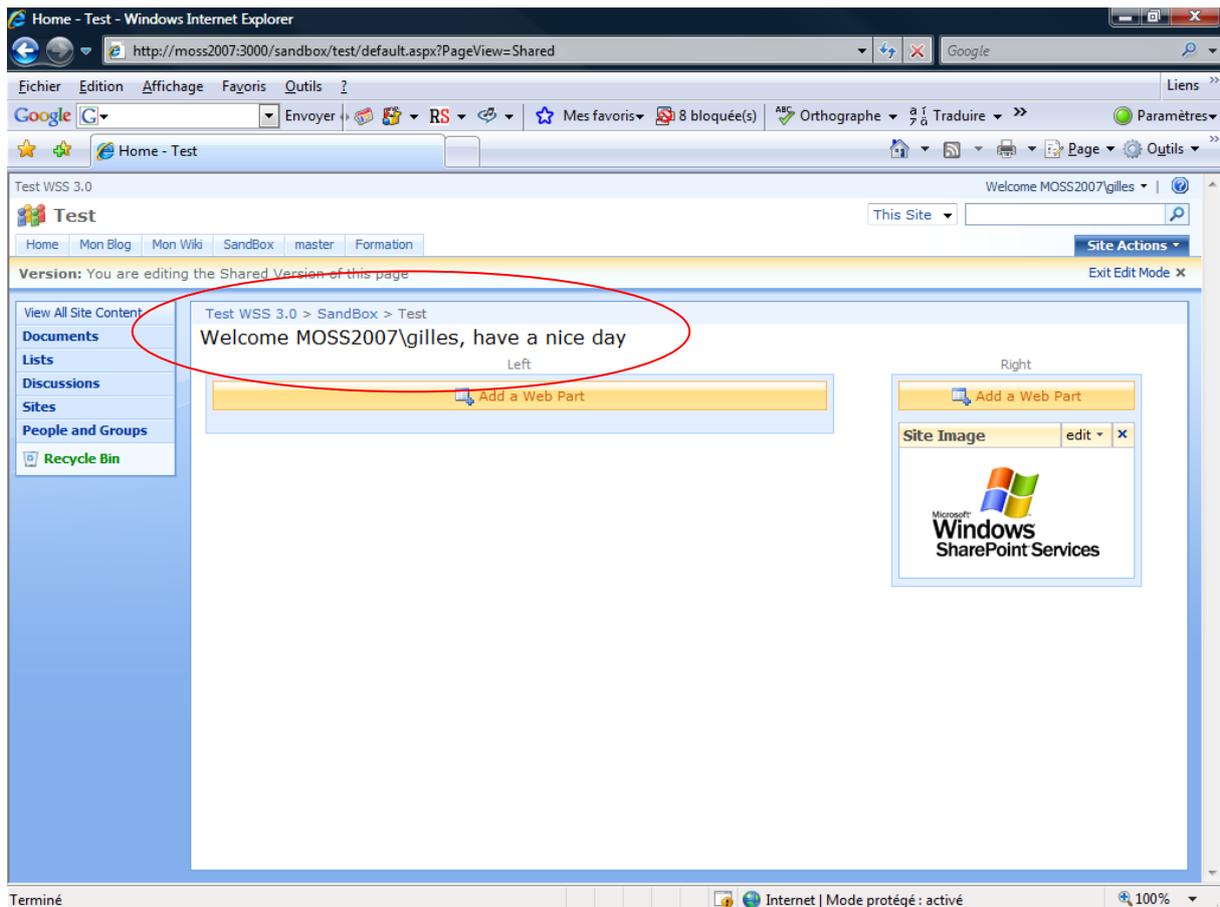


Figure 4 : surcharge du Delegate control

C'est bien le nouveau control qui s'affiche !

Delegate Controls et Master Pages

Dans notre exemple, afin de bien comprendre ce qui se passait nous avons inclus le Delegate control directement sur la page d'accueil du site. En réalité ce n'est pas là qu'il faudrait le placer, mais plutôt sur la Master Page du site. Les Master Pages de SharePoint contiennent déjà par défaut plusieurs Delegate Controls : le lien sur « Mon Site » (ou « My Site » pour MOSS), la barre de lancement rapide, la saisie pour le moteur de recherche. Dès lors, avec ce que nous avons vu, il est tout à faire envisageable de surcharger ces contrôles natifs SharePoint par nos propres contrôles !

Cela nous permet par exemple de masquer des composants que l'on ne souhaite pas mettre à disposition des utilisateurs, tel que le « My Site ».

Je vous invite à consulter l'article sur [la personnalisation des Master Page](#).

En conclusion

Les Delegate controls offrent de nouvelles possibilités quand au développement de sites SharePoint. En les alliant avec les webparts traditionnelles, ils permettent de créer des interfaces plus riches et réutilisables à plusieurs niveaux par le biais de features.

En les plaçant sur une Master Page, ils peuvent être déployés de manière globale sur l'ensemble d'un portail SharePoint et même surcharger les contrôles natifs existants.

Toutes ces fonctionnalités ne visent qu'un seul but : répondre au mieux aux besoins de nos clients.

Voici quelques liens utiles si cet article vous a intéressé :

- [Créer une page personnalisée pour WSS V3 \(FR\)](#)
- [Modifier la Master Page de SharePoint \(FR\)](#)
- [Créer un Site Template pour WSS V3 \(FR\)](#)
- [Création d'un User Control et utilisation de celui-ci en ASP.NET \(sous Visual Studio.NET et en VB.NET\) \(FR\)](#)
- [Installation de WSS V3 \(FR\)](#)
- [WSS V2 et SQL Server 2005 Express Edition \(FR\)](#)
- [SQL Server 2005 Express Edition \(FR\)](#)
- [Office Online](#)
- [Club SPS MOSS FRANCE\(FR\)](#)

Kern Gilles

Développeur .NET et SharePoint [ilem SA](#)