



# **Introduction aux WebParts d'ASP.Net 2.0**

# *Sommaire*

I. Pré requis

II. Introduction

III. Composants Webparts

IV. Interface de base

V. Configuration des webparts

VI. Ajouter des WebPart depuis Internet Explorer

VII. Conclusion

VIII. Remerciement

## I. Pré requis

Cet article nécessite Visual Studio 2005 et SQL Express 2005. Les webparts expriment toute leur puissance avec Internet Explorer.

## II. Introduction

Les utilisateurs de SharePoint (portal server ou services) ne seront pas dépayés par les webparts. Une Webpart est un contrôle serveur spécialisé dans la création de sites portails. Les webparts permettent de découper une page principale en plusieurs zones. Asp.net 2.0 utilise le même moteur de rendu que Sharepoint mais les webparts respectives ne sont pas compatibles. Nous allons voir dans cet article comment ajouter des webparts, simplement à une page.

## III. Composants Webparts

Dans la toolbox de Visual Studio, on retrouve un onglet Webparts. On constate plusieurs contrôles :

Malheureusement, nous n'allons pas rentrer en détails dans tous les contrôles.

**WebPartManager** :  
Comme son nom l'indique, c'est le manager des webparts de votre page. Ce control gère les webparts de votre page. Sa présence est indispensable.

**WebPartZone** :  
C'est une zone ou l'on souhaite mettre des webparts. Une webpartzone peut contenir plusieurs webparts et une page peut contenir plusieurs webpartzones.

**Catalogzone** :  
C'est une zone qui permet de contenir des catalogParts. Elle permet la gestion des webparts dans les webzones.

**EditorZone** :  
C'est une zone qui permet de contenir des editorzones. C'est une zone qui configure les webparts.

## IV. Interface de base

Pour cette interface de base, nous plaçons le webpartmanager sur la page (peu importe l'emplacement). Puis nous allons placer un tableau de 4 colonnes et de 3 lignes. Dans chaque colonne de la première ligne nous allons placer un linkbutton.

Ajouter un événement OnClick sur les deux premiers Linkbutton puis ajoutez le code suivant :

```

protected void NormalButton_Click(object sender, EventArgs e)
{
    WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
}

protected void DesignButton_Click(object sender, EventArgs e)
{
    WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode;
}

```

Dans les deux premières cases de la 2eme ligne, ajoutez une webpartzone.

Dans la première webzone, placez-y un contrôle. Tout type de contrôle peut être contenue par une webpartzone, également vos usercontrols. Dans l'exemple nous placerons un label dans la webpartzone1 et un calendrier dans la webpartzone2.

Afin d'avoir un premier aperçu, lancez la page web.

On retrouve donc bien nos linkbutton et nos deux webparts. Cliquez maintenant sur Design. Les 2 webpartzones apparaissent. Ensuite, cliquez sur le titre d'un webpart et faites le glisser de webpartzone en webpartzone tout en laissant le bouton gauche de la souris enfoncé.

Puis cliquez sur normal pour repasser en mode normal. Ensuite fermez votre navigateur, puis relancez-le sur la même page. Et là, " magie ", tout est resté à la même place ! Cela signifie que votre personnalisation de la page a été sauvegardée. En effet, une fois revenu en mode normal, la disposition des webparts a été sauvegardée en base.

## V. Configuration des webparts

Ajoutez un événement OnClik au dernier Linkbutton et ajoutez ce code:

```

protected void ConfigurationButton4_Click(object sender, EventArgs e)
{
    WebPartManager1.DisplayMode =WebPartManager.EditDisplayMode;
}

```

Puis dans Visual Studio, repassez en mode Design et ajoutez depuis la toolbox, un editorzone à la dernière colonne de la seconde ligne du tableau. Ensuite, rajoutez dans l'editorzone, un AppearanceEditorPart et un LayoutEditorPart.

Puis lancez votre navigateur et cliquez sur configuration. A coté de chaque Webpart, il y a un petit triangle, cliquez dessus et un petit menu apparaît. Ce sont les Verbs, puis cliquez sur Edit.

La grille de configuration apparaît, amusez vous avec puis cliquez sur ok. La Webpart sélectionnée est ainsi configurée et enregistrée dans la base de personnalisation.

## VI. Ajouter des WebPart depuis Internet Explorer

Ajoutez dans la troisième colonne une CatalogZone, puis à l'intérieur de celle-ci un Declarative Catalog. Ensuite affichez la DeclarativeCatalog Task (en cliquant sur le petit triangle).

Puis cliquez sur Edit Templates et laissez les options par défaut. Ensuite à l'intérieur de la Declarative Catalog, ajoutez par exemple un autre calendrier.

Ensuite ajoutez un événement OnClick sur le troisième linkbutton et ajoutez ce code :

```
protected void AffichageButton3_Click(object sender, EventArgs e)
{
    WebPartManager1.DisplayMode = WebPartManager.CatalogDisplayMode;
}
```

Une fois terminé, lancez votre navigateur. Cliquez donc sur Affichage, et sélectionnez le premier check box et choisissez dans quelle zone vous souhaitez qu'il apparaisse. Une fois ceci fait, cliquez sur OK. Un deuxième calendrier est alors apparu. Bien sûr, ceci est enregistré dans la base de personnalisation. Ceci n'est qu'un petit aperçu des possibilités des catalogzone.

## VII. Conclusion

Voici un petit aperçu des possibilités des webparts. J'ai prévu de concocter un article pour chaque zone, principalement sur les editorzones. Ils seront beaucoup plus techniques.

## VIII. Remerciement

Je tiens à remercier M. Daniel Tizon sans qui cet article n'aurait jamais existé.