

# Sécuriser les composants d'un Webpart

par Ludovic Lefort ([Site web](#)) ([Blog](#))

Dernière mise à jour :

Comment afficher/cacher des contrôles dans un webpart suivant les permissions de l'utilisateur connecté.

- 1 - Introduction
- 2 - Développement d'un webpart de base
- 3 - Premier déploiement
- 4 - Ajout de la sécurité
- 5 - Liste des permissions
- 6 - Référence
- 7 - Liens
- 8 - Sources du projet

## 1 - Introduction

Il est parfois nécessaire de pouvoir afficher les contrôles contenus dans un webpart suivant les droits que possède l'utilisateur connecté.

L'objet **SPSecurityTrimmedControl** est justement là pour ça. Dans ce document, je vais par un exemple simple, vous expliquer comment l'utiliser.

## 2 - Développement d'un webpart de base

Pour notre exemple il nous faut un webpart de base qui contiendra deux lignes de textes : la première devra être visible par tout le monde et la deuxième uniquement réservée aux administrateurs du site.

Première étape : Créons le webpart et laissons les deux textes être vus par tout le monde.

Dans Visual Studio nous allons créer un projet de type **Class Library** et l'appeler **WebpartSecurity**.

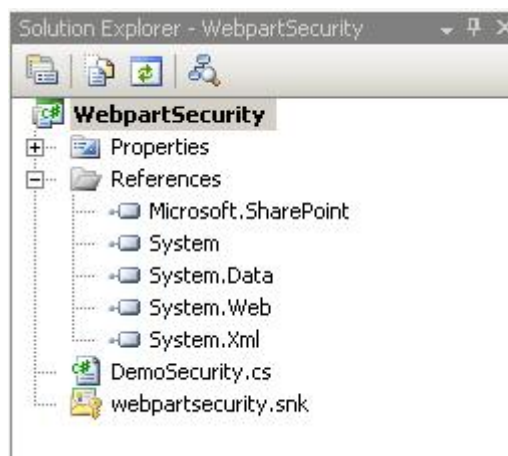
Renommez le fichier class1.cs en DemoSecurity.cs et ajoutez les références suivantes à votre projet:

- System.web
- Microsoft.Sharepoint (C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\ISAPI\Microsoft.Sharepoint.dll)

Nous devons maintenant signer notre assembly afin de pouvoir la copier dans la GAC lors du déploiement.

- Clic droit sur le projet
- propriétés
- Signing
- Choisir new dans la liste déroulante
- Entrez le nom **webpartsecurity.snk**
- décochez **Protect my key with a password** et cliquez sur OK

Après compilation vous devez obtenir ceci :



Pour être utilisé dans Sharepoint un Webpart doit hériter de la classe **System.Web.UI.WebControls.WebParts.WebPart**

Ajoutons donc un using de l'assembly **System.Web.UI.WebControls.WebParts** et définissons l'héritage au niveau de l'entête de classe:

```
using System.Web.UI.WebControls.WebParts;  
public class DemoSecurity : WebPart  
{
```

```
}
```

Prochaine étape surcharger la méthode **CreateChildControls** pour créer nos deux lignes de textes sur le webPart:

```
public class DemoSecurity : WebPart
{
    protected override void CreateChildControls()
    {
        base.CreateChildControls();
        Controls.Add(new LiteralControl("Public Text

"));
        Controls.Add(new LiteralControl("Administrator Text

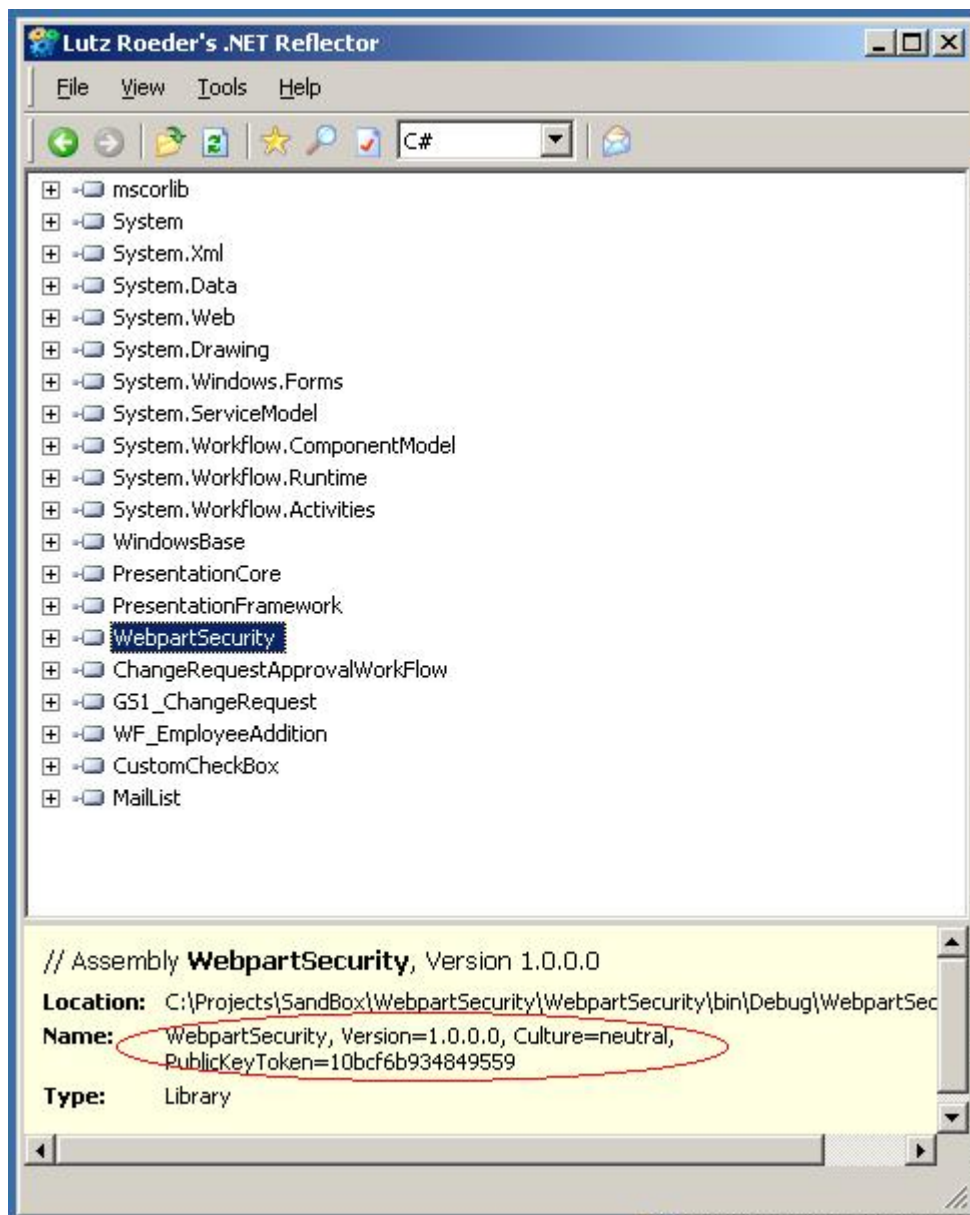
"));
    }
}
```

### 3 - Premier déploiement

Déployons et testons maintenant notre webpart dans Sharepoint.

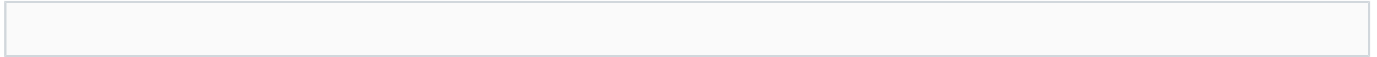
Nous allons commencer par déployer notre assembly dans la GAC pour cela "drag and droppez" le fichier WebpartSecurity.dll dans le répertoire **c:\windows\assembly**

Une fois que cela est fait nous allons récupérer le strong name de notre assembly à l'aide du très pratique **.net reflector**



Modifions le fichier web.config de notre application Sharepoint (par défaut : C:\inetpub\wwwroot\wss\VirtualDirectories\...) afin de truster notre assembly.

Il faut ajouter une ligne dans la section <Safecontrols> :



Le webpart est maintenant connu par Sharepoint et prêt à être ajouté dans la Webpart gallery

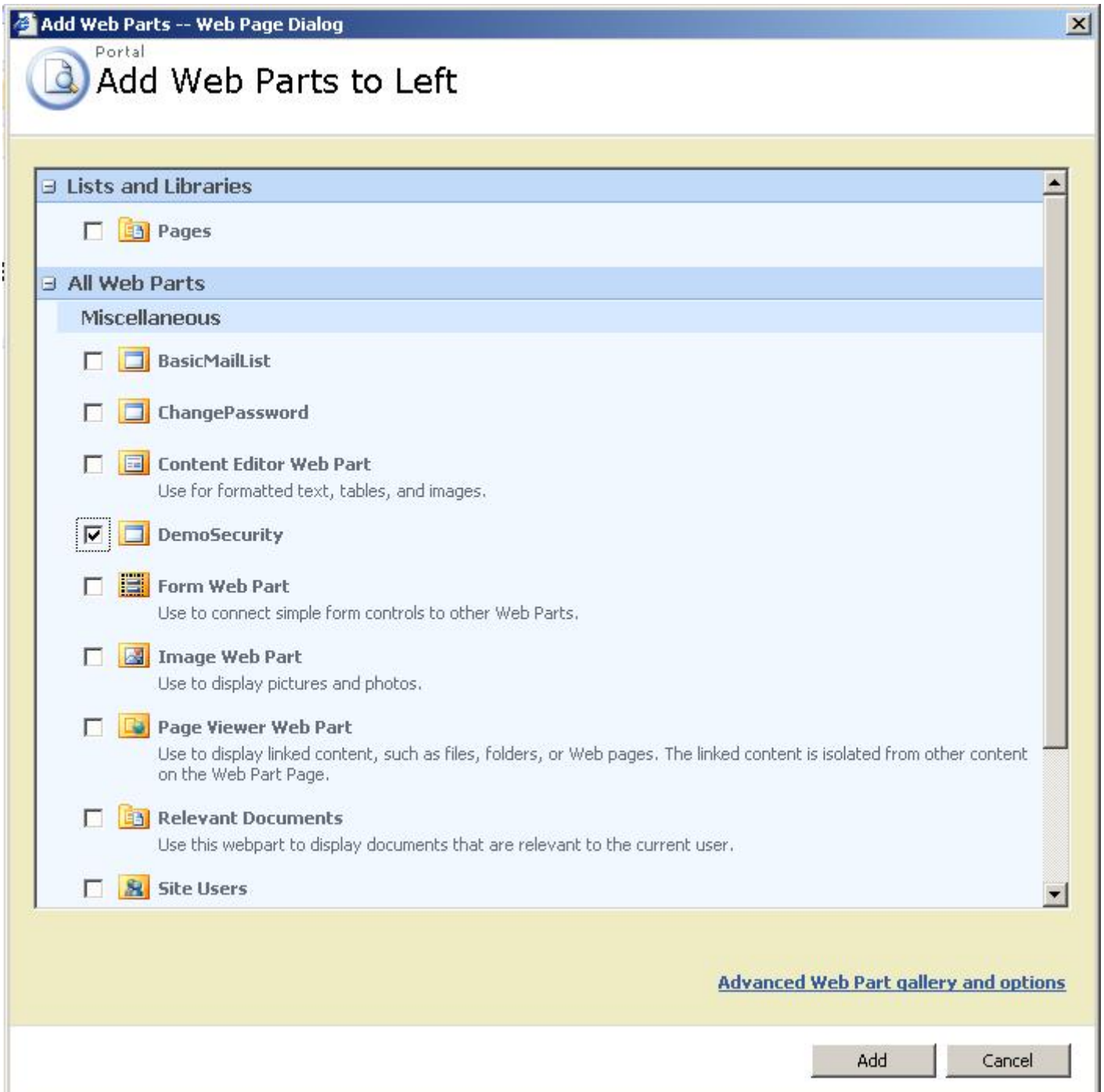
Pour cela connectez vous en tant qu' administrateur sur votre site Sharepoint et faites :

- Site Actions
- Site Settings
- Galleries > Webparts
- New
- Cochez la ligne concernant notre WebPart

rWebPart	<input type="checkbox"/>	TitleBarWebPart	.dwp	Microsoft.SharePoint, Version=12.0.0.0, Culture=fr-FR
ocsWebPart	<input type="checkbox"/>	UserDocsWebPart	.dwp	Microsoft.SharePoint, Version=12.0.0.0, Culture=fr-FR
isksWebPart	<input type="checkbox"/>	UserTasksWebPart	.dwp	Microsoft.SharePoint, Version=12.0.0.0, Culture=fr-FR
bPart	<input type="checkbox"/>	XmlWebPart	.dwp	Microsoft.SharePoint, Version=12.0.0.0, Culture=fr-FR
	<input type="checkbox"/>	DemoSecurity	.webpart	WebpartSecurity, Version=1.0.0.0, Culture=fr-FR

- Cliquez sur le bouton **Populate Gallery** en haut de la page

Nous pourrons alors ajouter notre Webpart sur une WebpartPage:



DemoSecurity

## **Public Text**

## **Administrator Text**

## 4 - Ajout de la sécurité

Nous voulons que la deuxième ligne de texte ne soit visible que par les administrateurs du site, pour cela nous allons utiliser un objet de type **SPSecurityTrimmedControl**

. Voici le nouveau code :

```
using System.Web.UI.WebControls.WebParts;
using System.Web.UI;
using Microsoft.SharePoint;
using Microsoft.SharePoint.WebControls;
namespace WebpartSecurity
{
    public class DemoSecurity : WebPart
    {
        protected override void CreateChildControls()
        {
            base.CreateChildControls();
            SPSecurityTrimmedControl sptrim = new SPSecurityTrimmedControl();
            sptrim.PermissionsString = "ManageWeb";
            Controls.Add(new LiteralControl("Public Text

"));
            sptrim.Controls.Add(new LiteralControl("Administrator Text

"));
            Controls.Add(sptrim);
        }
    }
}
```

Quels sont les changements ?

- Ajout d'un using pour **Microsoft.SharePoint.WebControls**
- Création d'un objet **SPSecurityTrimmedControl** avec la propriété `PermissionsString` à "ManageWeb"

```
SPSecurityTrimmedControl sptrim = new SPSecurityTrimmedControl();
sptrim.PermissionsString = "ManageWeb";
```

- Nous n'ajoutons plus la deuxième ligne de texte directement sur le webpart mais dans l'objet `sptrim` que nous ajoutons sur le webpart

```
sptrim.Controls.Add(new LiteralControl("Administrator Text

"));
Controls.Add(sptrim);
```

Recompilez votre projet, re-déployez la dll dans la GAC et faites un IISreset

Si vous testez maintenant votre webpart en tant qu'administrateur vous obtenez :

---

DemoSecurity

## Public Text

## Administrator Text

alors que en tant que non-administrateur :

---

DemoSecurity

## Public Text

**Remarque** : Pour cumuler plusieurs permissions (ET logique) il faut les séparer par une virgule:

```
sprim.PermissionsString = "ManageWeb, AddAndCustomizePages" ;
```

## 5 - Liste des permissions

### Permissions sur les listes :

- ManageLists
- CancelCheckout
- AddListItems
- EditListItems
- DeleteListItems
- ViewListItems
- ApproveItems
- OpenItems
- ViewVersions
- DeleteVersions
- CreateAlerts
- ViewFormPages

### Permissions sur les sites

- ManagePermissions
- ViewUsageData
- ManageSubwebs
- ManageWeb
- AddAndCustomizePages
- ApplyThemeAndBorder
- ApplyStyleSheets
- CreateGroups
- BrowseDirectories
- CreateSSCSite
- ViewPages
- EnumeratePermissions
- BrowseUserInfo
- ManageAlerts
- UseRemoteAPIs
- UseClientIntegration
- Open
- EditMyUserInfo

### Permissions personnelles

- ManagePersonalViews
- AddDelPrivateWebParts
- UpdatePersonalWebParts

## 6 - Référence

MSDN : **SPSecurityTrimmedControl Class (Microsoft.SharePoint.WebControls)**

## 7 - Liens

Mon site : <http://lefortludovic.developpez.com>

Mon blog : <http://blogs.ezos.com/blog/lle>

## 8 - Sources du projet

Cliquez [ici](#) pour télécharger les sources de ce projet.

Je tiens à remercier [stephane eyskens](#) pour la correction orthographique de cet article.

