

Sharepoint 2007 : Excel Services

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

Dernière mise à jour :

Administration, configuration et développement autour des Excel Services de Sharepoint 2007

0 - Introduction.....	3
1 - Configuration d'une "Trusted Location".....	4
2 - Création de notre classeur Excel.....	5
3 - Publication du classeur.....	6
4 - Les fonctions personnalisées (User-defined functions).....	9
5 - Excel Services Web Part.....	12
6 - Référence.....	14
7 - Liens.....	15

0 - Introduction

Dans beaucoup d'entreprises chaque département possède des tableaux Excel plus ou moins complexes utilisant des graphiques, des tableaux croisés dynamiques, etc ...

Anciennement le développeur qui voulait intégrer cela dans une application devait réécrire tout les algorithmes de calcul existants ... un vrai casse tête.

Grâce à Sharepoint 2007 et aux Excel Services il est possible d'utiliser directement les formules se trouvant dans un tableau.

Excel Services n'est disponible que dans la version MOSS de Sharepoint 2007

Les objectifs de cet article :

- Apprendre les bases de l'administration et de la configuration
- Apprendre à préparer et publier des feuilles Excel
- Apprendre à créer des fonctions personnalisées

1 - Configuration d'une "Trusted Location"

Pour pouvoir être utilisés les classeurs Excel doivent se trouver dans une "Trusted Location". Autrement dit vous devez définir un emplacement comme étant sûr.

Cette configuration se fait dans la "Central Administration" de votre serveur MOSS 2007 dans la partie **Shared Services**

Cliquez sur le menu **Trusted File Location** du groupe **Excel Services Settings**



Cliquez sur **Add Trusted File Location** pour ajouter un nouvel emplacement

L'adresse peut être une liste Sharepoint, un webfolder ou un répertoire partagé sur le réseau. Dans notre exemple nous allons utiliser une Document Library.

Entrez l'adresse d'une Document Library :

The image shows the 'Add Trusted File Location' configuration form. It has the following sections:

- Address**: The full Windows SharePoint Services location, network file share or Web folder address of this trusted location. The text box contains 'http://xxxxxx/DemoExcelServices'.
- Location Type**: Storage type of this trusted location. Radio buttons are selected for 'Windows SharePoint Services', 'UNC', and 'HTTP'.
- Trust Children**: Trust child libraries or directories. The checkbox 'Children trusted' is unchecked.
- Description**: The optional description of the purpose of this trusted location. There is an empty text area below.

Nous allons également ajouter un autre paramètre qui nous sera utile plus tard.

- **Allow User-Defined Functions** --> Cochez la case

2 - Création de notre classeur Excel

Tous les classeurs Excel ne peuvent pas être utilisés, il y a certaines limitations :

- Le classeur ne peut pas contenir de code embarqué comme du VBA par exemple
- Certains graphiques complexes ne sont pas affichés correctement
- Il est impossible d'utiliser l'API d'excel dans Excel Services

Pour cet article nous allons créer une application Excel plus que basique qui nous permettra simplement de calculer une ristourne :

	A	B	C	D	E	F
1	Excel Services Demonstration					
2						
3	Prix de base :		100			
4	Ristourne (%) :		10			
5	Prix :		90			

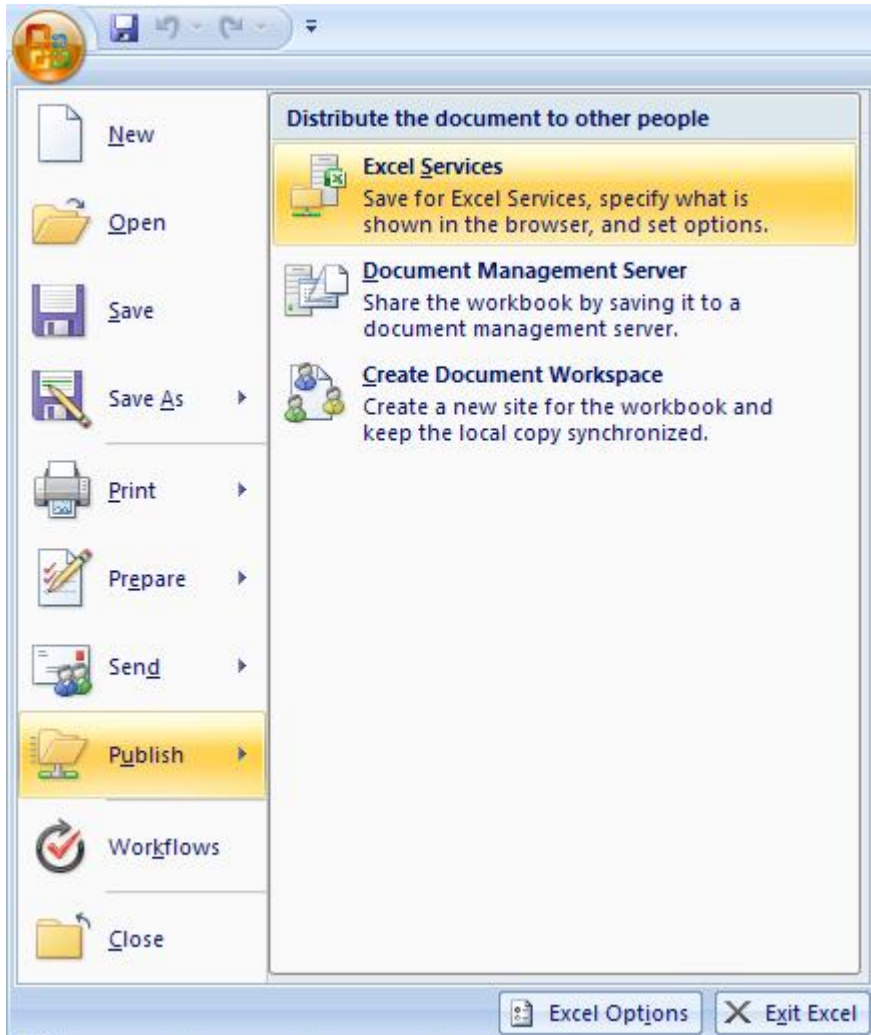
Ce classeur est téléchargeable [ici](#)

La calcul est basé sur deux paramètres passés à la feuille : **Le prix de base** et le montant de la **ristourne**, pour autoriser cela nous devons nommer les cellules qui contiendront ces valeurs.

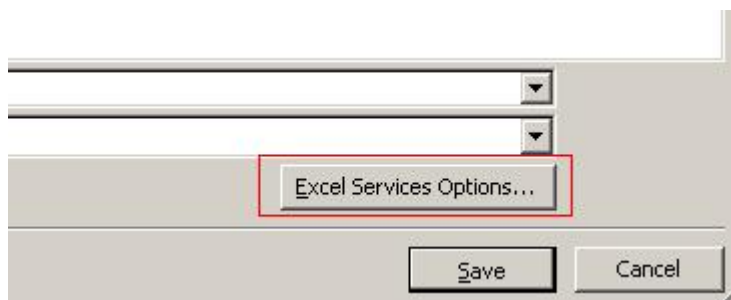
3 - Publication du classeur

Notre fichier est créé, nous devons maintenant le publier dans la document library que nous avons ajoutée aux **Trusted Location** dans le premier chapitre.

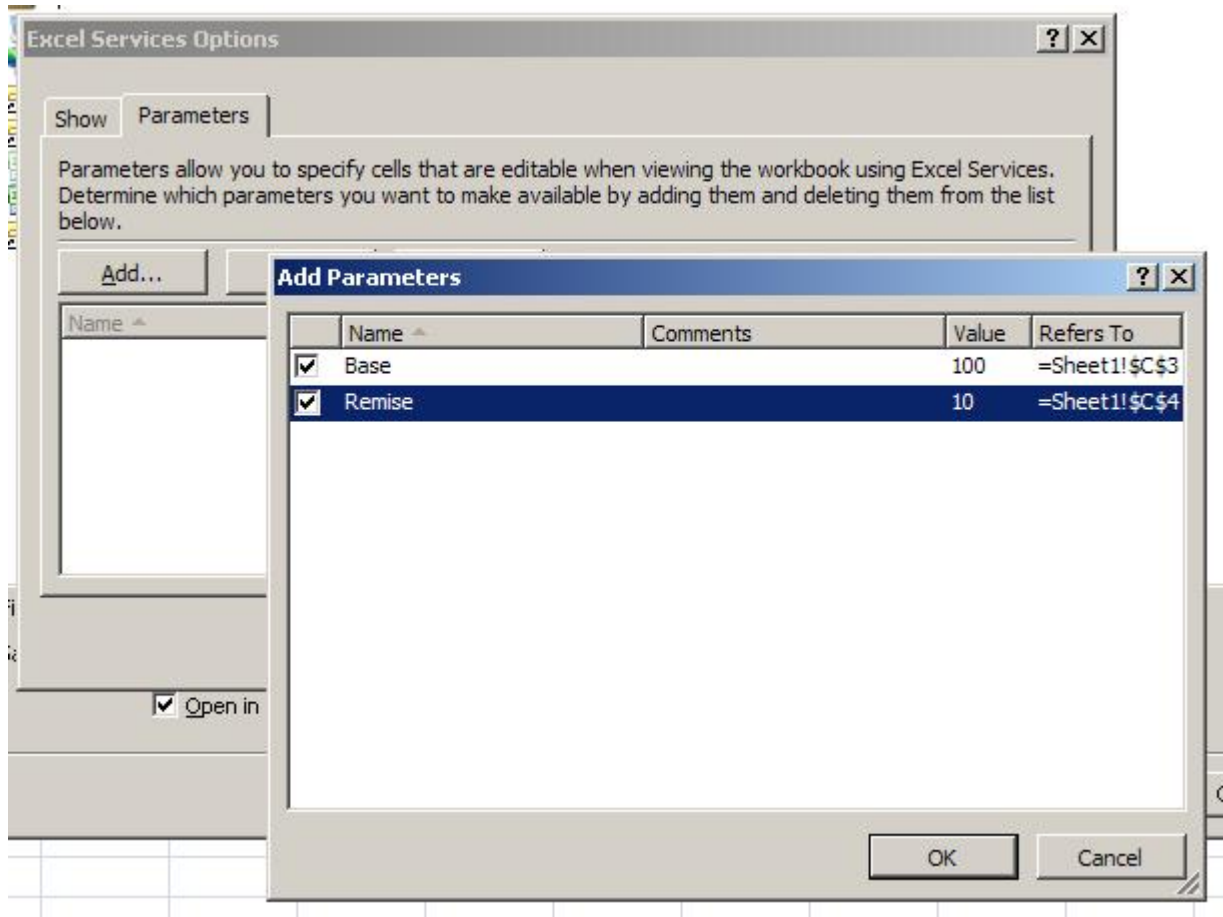
Dans le menu **Publish** d'Excel choisissez **Excel Services**



Avant de publier notre fichier nous devons définir les paramètres de la feuille de calcul, pour cela utilisez le bouton **Excel Services Options**



Ajoutez les paramètres dans l'onglet **Parameters**



Entrez l'adresse de votre document library et cliquez sur **Save** pour publier votre fichier



Après quelques secondes d'attente votre fichier Excel sera chargé dans votre navigateur avec un volet à droite pour les paramètres.

Cela nous permet de découvrir un autre avantage d'Excel Services : Les utilisateurs peuvent utiliser les applications publiées sans avoir Excel installé sur leur machine.

The screenshot displays the Excel Services interface. On the left, a spreadsheet is visible with the following data:

	A	B	C
1	Excel Services Demonstration		
2			
3	Prix de base :		450
4	Ristourne (%) :		15
5	Prix :		382.5
6			
7			
8			
9			
10			

On the right, the 'Parameters' pane is open, showing the following parameters used in the workbook:

- Base: 450
- Remise: 15

Buttons for 'Apply' and 'Clear' are located at the bottom of the Parameters pane.

4 - Les fonctions personnalisées (User-defined functions)

Excel Services offre la possibilité d'écrire des fonctions personnalisées en .Net.

C'est un moyen très performant de contourner l'impossibilité d'utiliser du VBA dans la feuille

Grâce au **User-defined functions**(UDF) vous pouvez par exemple vous connecter à une base de données ou appeler un Web service.

Pour illustrer l'utilisation des UDF nous allons créer une fonction identique à celle utilisée dans la feuille pour calculer une ristourne mais au travers d'une library .net

Première étape : créons un projet de type Class Library dans Visual Studio.net. Je décide d'appeler mon projet **CalculRistourne**

Ajoutez une référence vers la library : **Microsoft.Office.Excel.Server.Udf.dll** qui par défaut se trouve dans **C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\ISAPI** ainsi que le using suivant :

```
using Microsoft.Office.Excel.Server.Udf
```

La classe doit être marquée de l'attribut **[UdfClass]** et les fonctions par **[UdfMethod]** voici à quoi doit ressembler votre classe une fois la méthode implémentée :

```
using System;
using System.Collections.Generic;
using System.Text;
using Microsoft.Office.Excel.Server.Udf;

namespace CalculRistourne
{
    [UdfClass]
    public class CalculRistourne
    {
        [UdfMethod]
        public double PrixAvecRistourne(double Prix,double ristourne)
        {
            return (Prix * (1-(ristourne/100)));
        }
    }
}
```

Prochaine étape : signer l'assembly pour permettre son déploiement dans Sharepoint :

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

L'emplacement de la dll est totalement libre, il faut juste que l'administrateur Sharepoint ajoute une référence vers celle ci.

Pour cela ouvrez la **Central Administration** et allez dans les Shared Services ensuite cliquez sur le menu **User-defined function assemblies**

Excel Services Settings

- ▣ Edit Excel Services settings
- ▣ Trusted file locations
- ▣ Trusted data connection libraries
- ▣ Trusted data providers
- ▣ User-defined function assemblies

Cliquez sur **Add-User Defined function Assembly**, Entrez le chemin complet vers votre dll, choisissez FilePath (il est également possible de déployer la dll dans la GAC) et cliquez sur OK

Assembly
Strong name or full path of an assembly that contains user-defined functions, which Excel Calculation Services can call. Examples:
SampleCompany.SampleApplication.SampleUdf, C:\UDFs\SampleUdf.dll, \\MyNetworkServer\UDFs\SampleUdf.dll.

Assembly Location
Location of the assembly:
 Global assembly cache
 File path

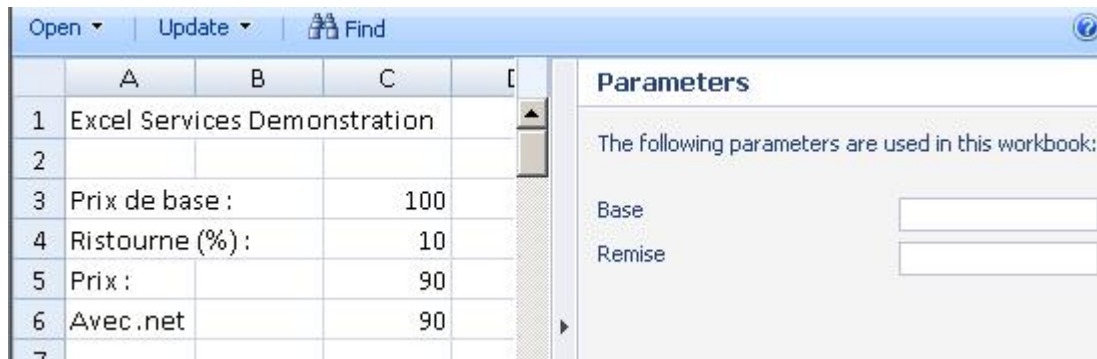
Enable Assembly
Allow this user-defined function assembly to be loaded and used by Excel Calculation Services. Turning off this option disables the assembly without having to completely remove the entry from the list.
 Assembly enabled

Description
The optional description of this user-defined function assembly.

Il nous reste maintenant à tester notre fonction, pour cela ouvrez votre classeur excel et ajoutez l'appel à la fonction dans une des cellules : **=PrixAvecRistourne(Base,Ristourne)**

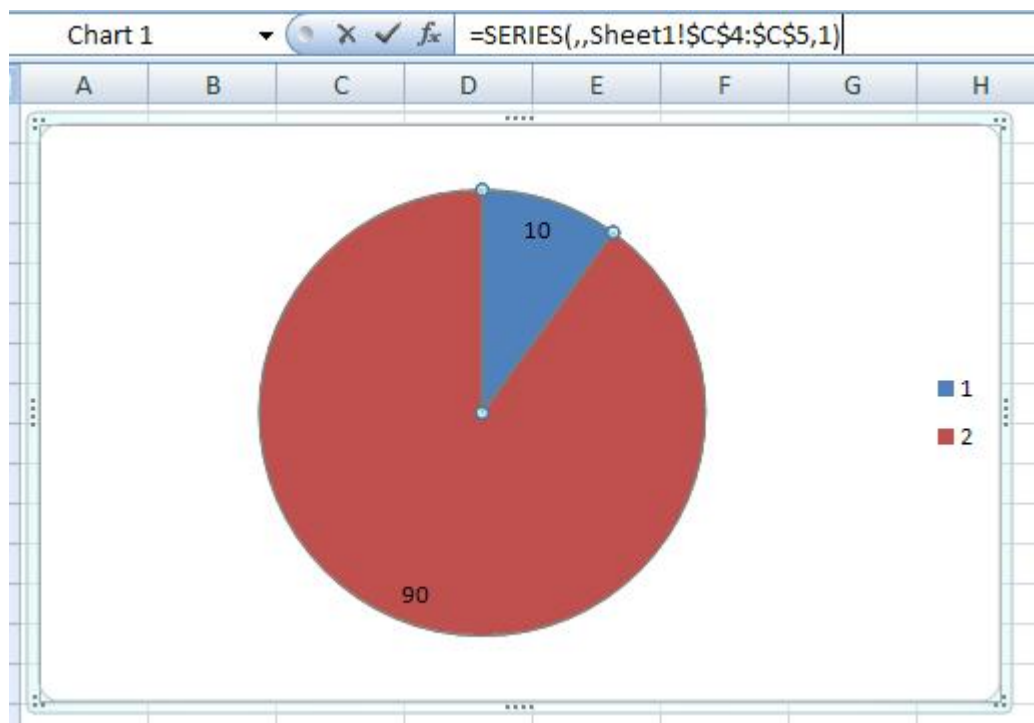
	A	B	C	D	E	F	G
1	Excel Services Demonstration						
2							
3	Prix de base :		100				
4	Ristourne (%) :		10				
5	Prix :		90				
6	Avec .net		=PrixAvecRistourne(B3,B4)				

Le résultat obtenu est **#NAME?** ... pas de panique les UDF ne fonctionnent que dans le contexte Sharepoint, il faut donc re-publier le classeur et le tester dans Sharepoint :



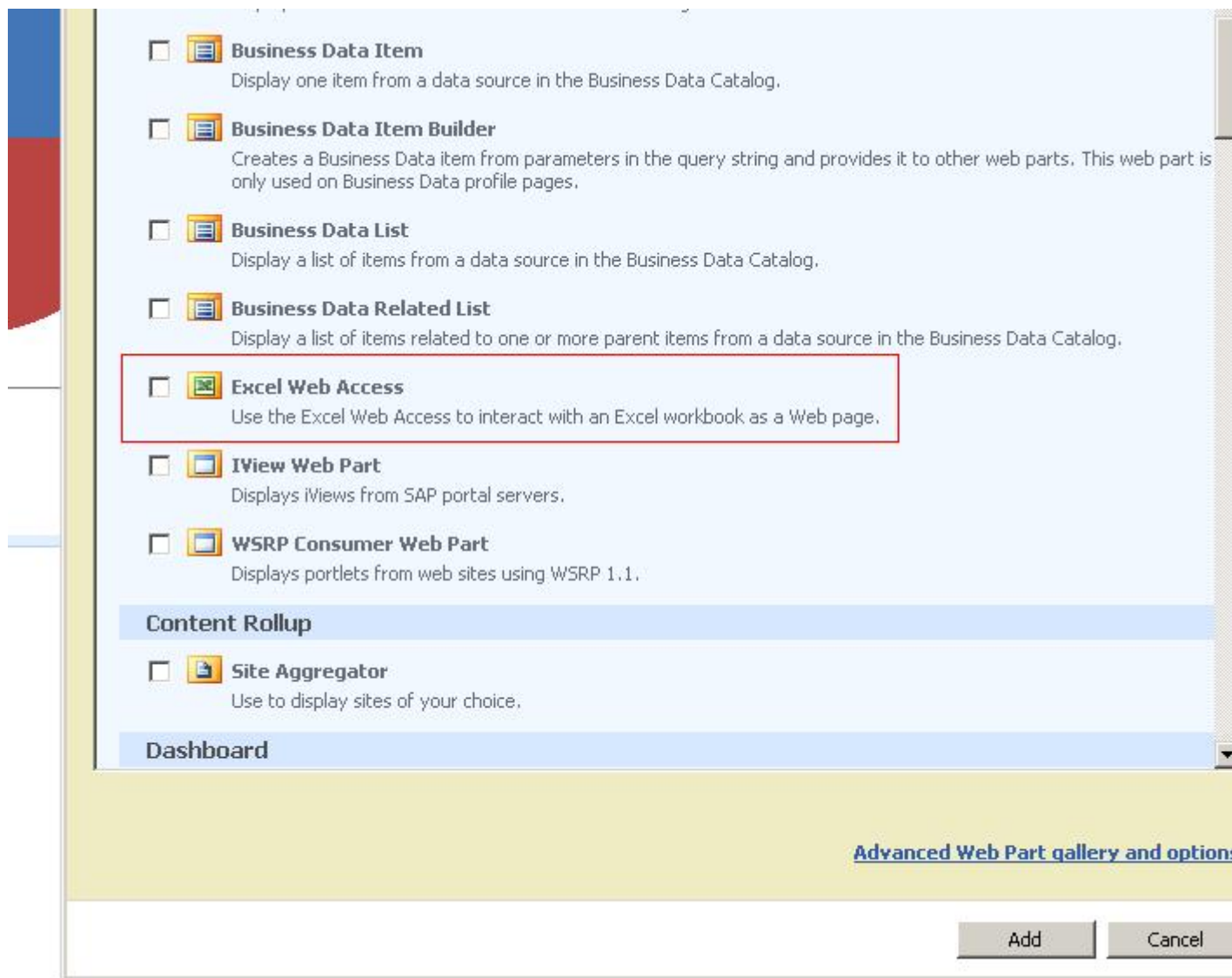
5 - Excel Services Web Part

Plutôt que d'afficher tout le classeur Excel il est possible de n'afficher qu'une partie de celui ci dans un WebPart. Modifions notre classeur excel afin d'ajouter un graphique sur notre deuxième Sheet et publions le à nouveau :

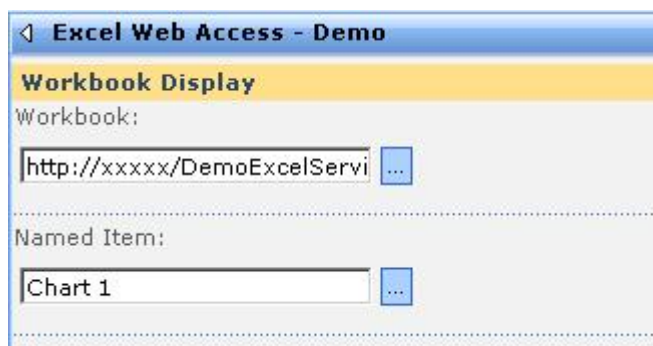


Ajoutons maintenant un webpart de type **Excel Web Access** sur une des pages de notre site

*Remarque : Si le webpart n'est pas présent dans votre liste c'est que la feature **Office Sharepoint Server Enterprise** n'est pas activée*



Modifiez les propriétés du web part pour n'afficher que le graphique. Choisissez votre classeur Excel et l'élément à afficher : **Chart 1** dans notre cas



6 - Référence

Livre : Inside Microsoft Office Sharepoint Server 2007 par Patrick Tisseghem

7 - Liens

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

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

Je tiens à remercier **RomainVALERI** pour la correction orthographique de cet article.

