

# Interopérabilité avec C#/.Net

## Excel

# Comment créer un fichier Excel en C #

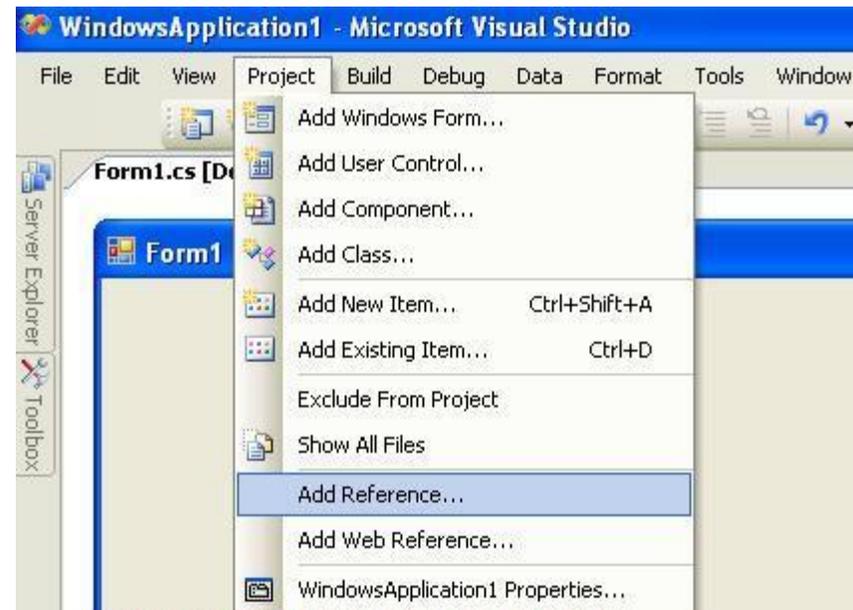
- L'exemple de code C # suivant montre comment utiliser COM Interop pour créer un fichier Excel.
- Avant d'aller pour créer un nouveau fichier Excel programmation en C #, vous devez avoir Excel installé sur votre système pour que ce code fonctionne correctement.

# Excel Bibliothèque

- Pour accéder au modèle d'objet à partir de Visual C # .NET, vous devez ajouter le X.0 Object Library Microsoft Excel.
- Créez un nouveau projet et ajoutez un bouton de commande à votre formulaire de C #.

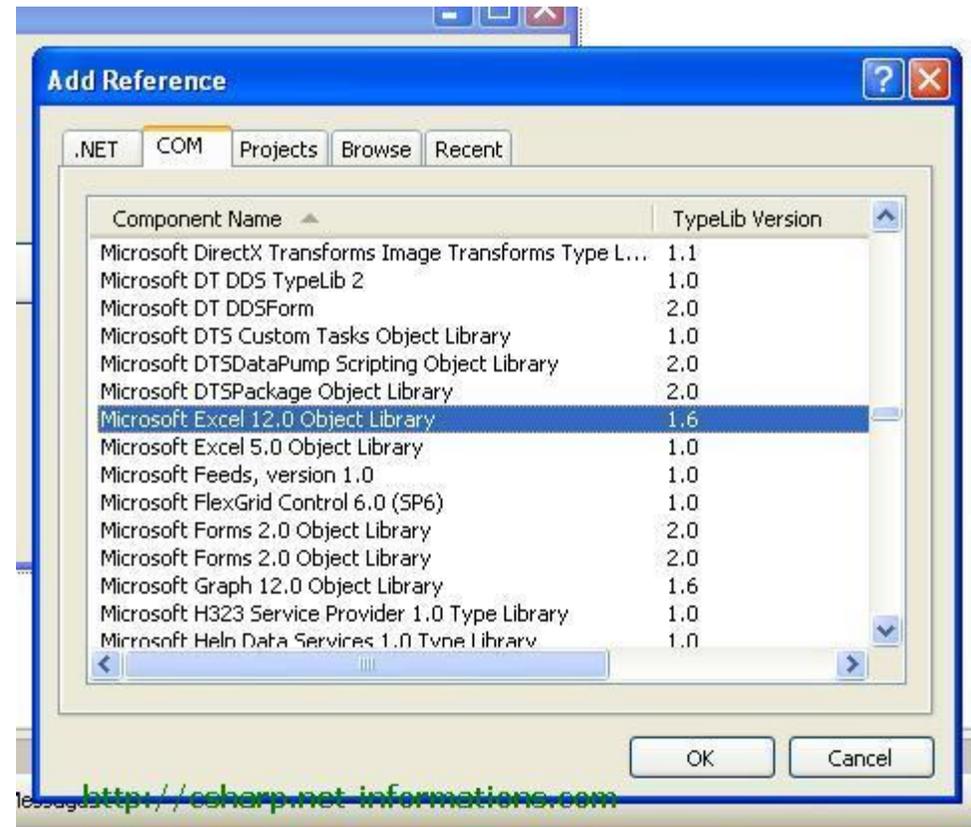
# Comment utiliser COM Interop pour créer une feuille de calcul Excel

- Former les images suivantes vous pouvez trouver comment ajouter une bibliothèque de référence Excel dans votre projet.
- Le dialogue de référence Sélectionnez dans le menu Projet

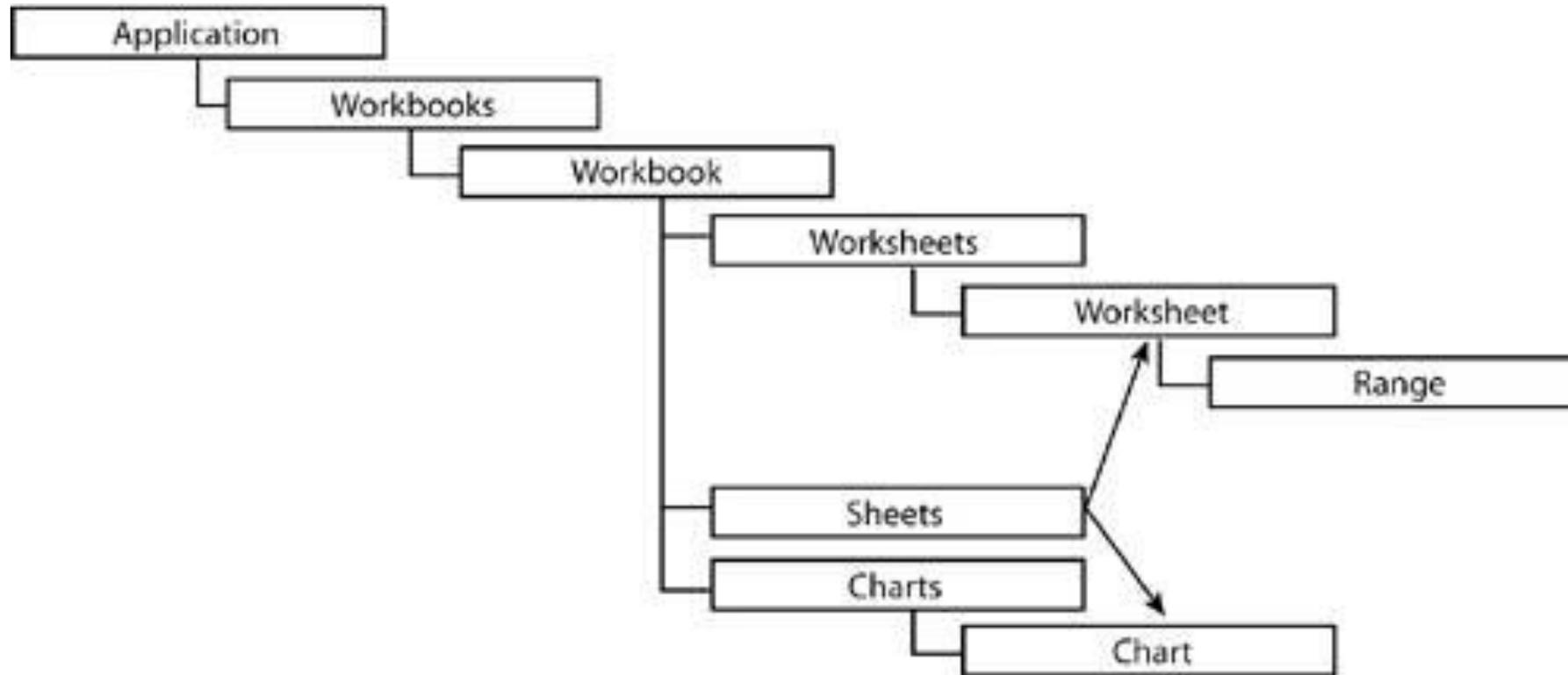


# Comment utiliser COM Interop pour créer une feuille de calcul Excel

- Sélectionnez Microsoft Excel 12.0 Object Library et cliquez sur le bouton OK



# Object Excel Model



# Comment créer et manipuler un document Excel par programme

- Nous devons d'abord initialiser l'application objet Excel.
- `Excel.Application xlApp = new Microsoft.Office.Interop.Excel.Application();`
- Avant de créer une nouvelle classeur Excel, vous devriez vérifier si Excel est installé dans votre système.
- `if (xlApp == null) { MessageBox.Show("Excel is not properly installed!!"); return; }`

# Comment créer et manipuler un document Excel par programme

- Puis créer un nouveau classeur
- `xlWorkBook = xlApp.Workbooks.Add(misValue);`
- Après avoir créé le nouveau classeur, l'étape suivante consiste à écrire le contenu de la feuille de travail
- `xlWorkSheet = xlWorkBook.Worksheets.get_Item(1);`
- `xlWorkSheet.Cells[1, 1] = "Sheet 1 content";`

# Comment créer et manipuler un document Excel par programme

- Dans le code ci-dessus nous écrivons les données dans la feuille Sheet1, Si vous voulez écrire des données dans la feuille 2, alors vous devriez coder comme ça ..

```
xlWorksheet = (Excel.Worksheet)xlWorkbook.Worksheets.get_Item(2);
```

```
xlWorksheet.Cells[1, 1] = "Sheet 2 content";
```

- Enregistrer le fichier Excel avec SaveAs () méthode
- Après écrire le contenu à la cellule, étape suivante consiste à enregistrer le fichier Excel dans votre système.
- *xlWorkbook.SaveAs("your-file-name.xls");*

# *Comment bien nettoyer des objets interop Excel*

*Comment libérer correctement les objets COM Excel c #*

*Enfin, nous devons nettoyer correctement les objets interop Excel ou Excel libération objets COM. Ici, nous écrivons une fonction pour nettoyer l'application Excel, Excel et Excel Feuille objets.*

```
private void releaseObject(object obj) {  
try { System.Runtime.InteropServices.Marshal.ReleaseComObject(obj); obj = null; }  
catch (Exception ex) {  
obj = null; MessageBox.Show("Exception Occured while releasing object " + ex.ToString()); }  
finally { GC.Collect(); } }
```