

Csharp : compléments

PARTIE 1 : CREATION D'UN COMPOSANT	2
I- INTRODUCTION	2
II- CREATION D'UN CONTROLE	2
1. CREATION DU PROJET	2
2. AJOUT DES CONTROLES WINDOWS ET DE COMPOSANTS	3
3. AJOUT DE PROPRIETES	3
III- TEST DU COMPOSANT	4
4. CREATION D'UN PROJET DE TEST	5

Partie 1 : Création d'un composant

I- Introduction

La création de contrôles permet de réutiliser des interfaces graphiques personnalisées. Ainsi, sous Visual Studio, il est possible de créer ses propres composants pour :

- Ses propres besoins applicatifs,
- Leur commercialisation ou leur distribution libre.

De tels composants sont appelés **contrôles composites**

Un contrôle composite est pour l'essentiel un composant doté d'une *représentation visuelle*, mais ce n'est pas une obligation.

Il peut être constitué :

- d'un ou de plusieurs contrôles Windows Forms,
- de composants ou de blocs de code,

Les blocs de code peuvent développer des fonctionnalités comme :

- La validation des entrées de l'utilisateur,
- La modification des propriétés d'affichage
- La mise en œuvre d'autres tâches imposées par le créateur du contrôle.

Les contrôles composites peuvent être placés dans les Windows Forms de la même façon que les autres contrôles. On peut aussi les retrouver dans les web Forms, ou encore dans les applications Smart Device.

II- Création d'un contrôle

1. Création du projet

Lorsque l'on crée un projet, on spécifie son nom pour définir l'espace de noms racine, le nom de l'assembly ainsi que celui du projet pour avoir la garantie que le composant par défaut sera placé dans l'espace de noms adéquat.

On va créer une bibliothèque de contrôles *CtlDeuxListeLib* et le contrôle *CtlDeuxListe*

Dans le menu *Fichier*, pointer sur *Nouveau*, puis cliquez sur *Projet* pour ouvrir la boîte de dialogue *Nouveau projet*.

Dans la liste de projets Visual C#, sélectionner le modèle de projet *Bibliothèque de contrôles Windows*, tapez *CtlDeuxListeLib* dans la zone *Nom*, puis cliquer sur **OK**.

Le nom de projet, *CtlDeuxListeLib*, est également assigné à l'espace de noms racine par défaut. L'espace de noms racine sert à qualifier le nom des composants de l'assembly. Par exemple, si deux assemblies contiennent des composants nommés *CtlDeuxListe*, on peut spécifier le composant créé *CtlDeuxListe* à l'aide de *CtlDeuxListeLib.CtlDeuxListe*.

Dans l'Explorateur de solutions, cliquer avec le bouton droit sur *UserControl1.cs*, puis cliquer sur *Renommer*. Remplacer le nom de fichier par *CtlDeuxListe.cs*. Cliquer sur le bouton *Oui* à la question de savoir si vous souhaitez renommer toutes les références à l'élément de code "UserControl1".

Remarque :

Par défaut, un contrôle composite hérite de la classe *UserControl* fournie par le système. La classe *UserControl* fournit les fonctionnalités requises par les contrôles composites et implémente des méthodes et des propriétés standard.

Dans le menu *Fichier*, cliquez sur *Enregistrer tout* pour enregistrer le projet.

2. Ajout des contrôles windows et de composants

L'interface visuelle est une partie essentielle du contrôle composite. Elle est implémentée par l'ajout d'un ou de plusieurs contrôles Windows à l'aire du concepteur.

On pourra utiliser tous les contrôles existants pour construire son propre contrôle.

Par exemple :



On rajoutera le code qui va bien.....

3. Ajout de propriétés

Les propriétés des contrôles composant notre contrôle composite ne seront pas accessibles par l'utilisateur.

On pourra toutefois se créer ses propres propriétés avec les accesseurs mutateurs qui vont avec et qui agiront sur les propriétés des composants du contrôle.

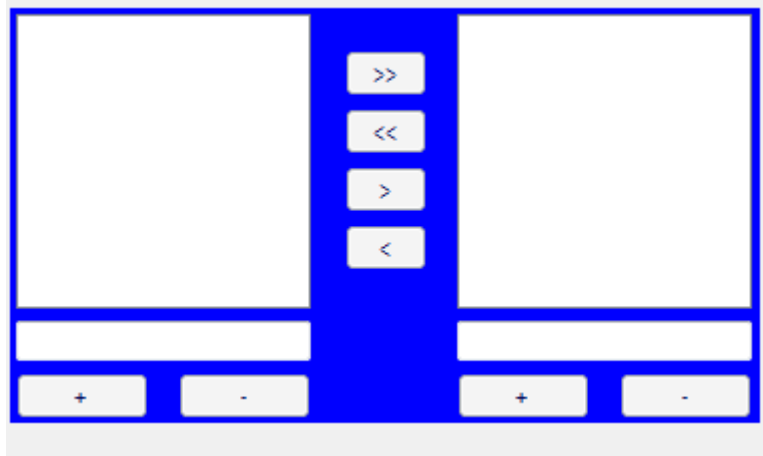
```
private Color _CouleurDeFond;
public Color CouleurDeFond
{
    get
    {
        return _CouleurDeFond;
    }
    set
    {
        _CouleurDeFond = value;
        this.BackColor = _CouleurDeFond;
    }
}
```

Et dans l'utilisation :

```
private void BtnCharger_Click(object sender, EventArgs e)
{
    colDialogFond.ShowDialog();
    CdlEmploye.CouleurDeFond = colDialogFond.Color;
}
```

On vient ici de créer une propriété `CouleurDeFond` qui permet de personnaliser la couleur de fond de notre contrôle.

Voilà ce que ça peut donner Bn, ce n'est pas top, mais c'est un exemple !!!



Dans le menu *Fichier*, cliquer sur *Enregistrer tout* pour enregistrer le projet.

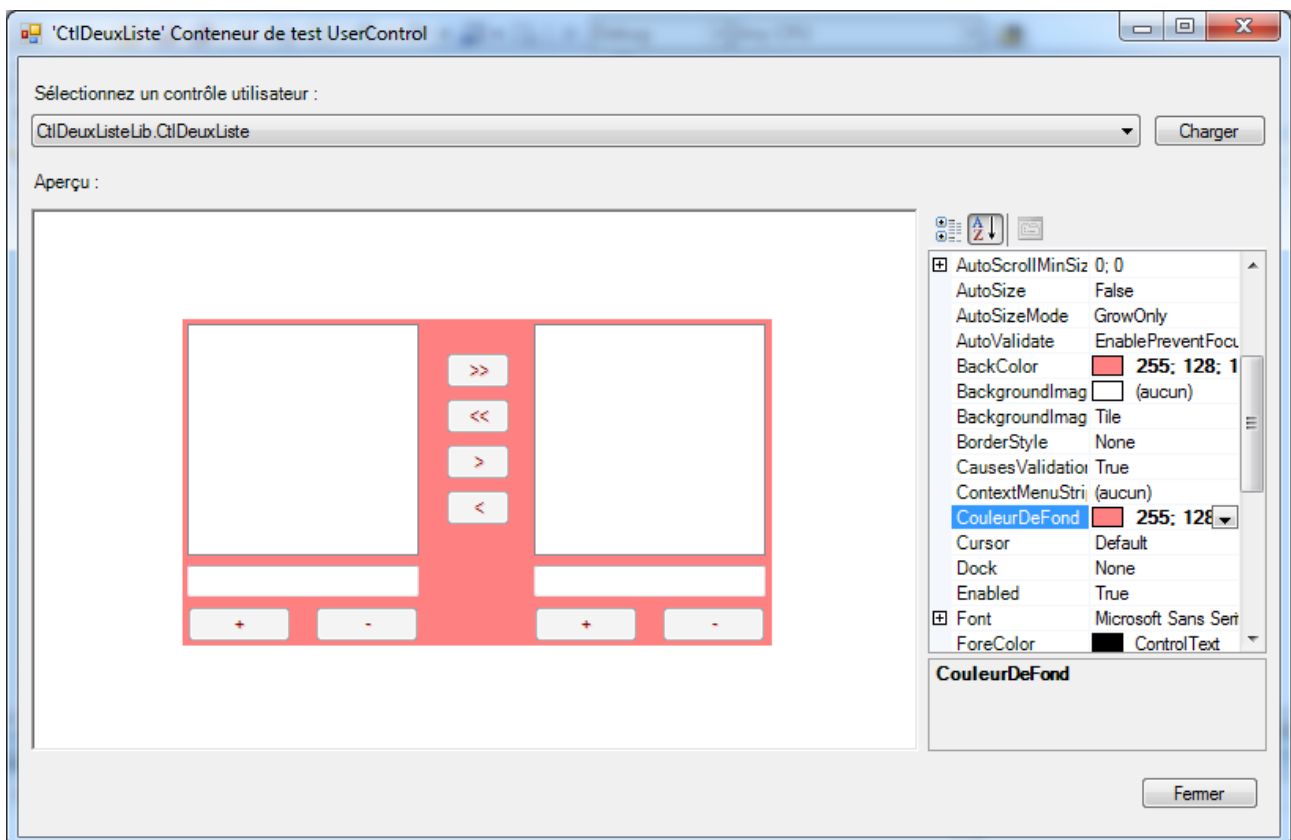
III- Test du composant

Appuyer sur F5 pour générer le projet et exécuter votre contrôle dans le *conteneur de test UserControl*.

Dans la grille des propriétés du conteneur de test, localisez la propriété *CouleurDeFond*, puis sélectionner la propriété pour afficher la palette de couleurs.

Choisissez une couleur en cliquant dessus.

La couleur d'arrière-plan du contrôle est remplacée par celle que sélectionnée.



4. Création d'un projet de test

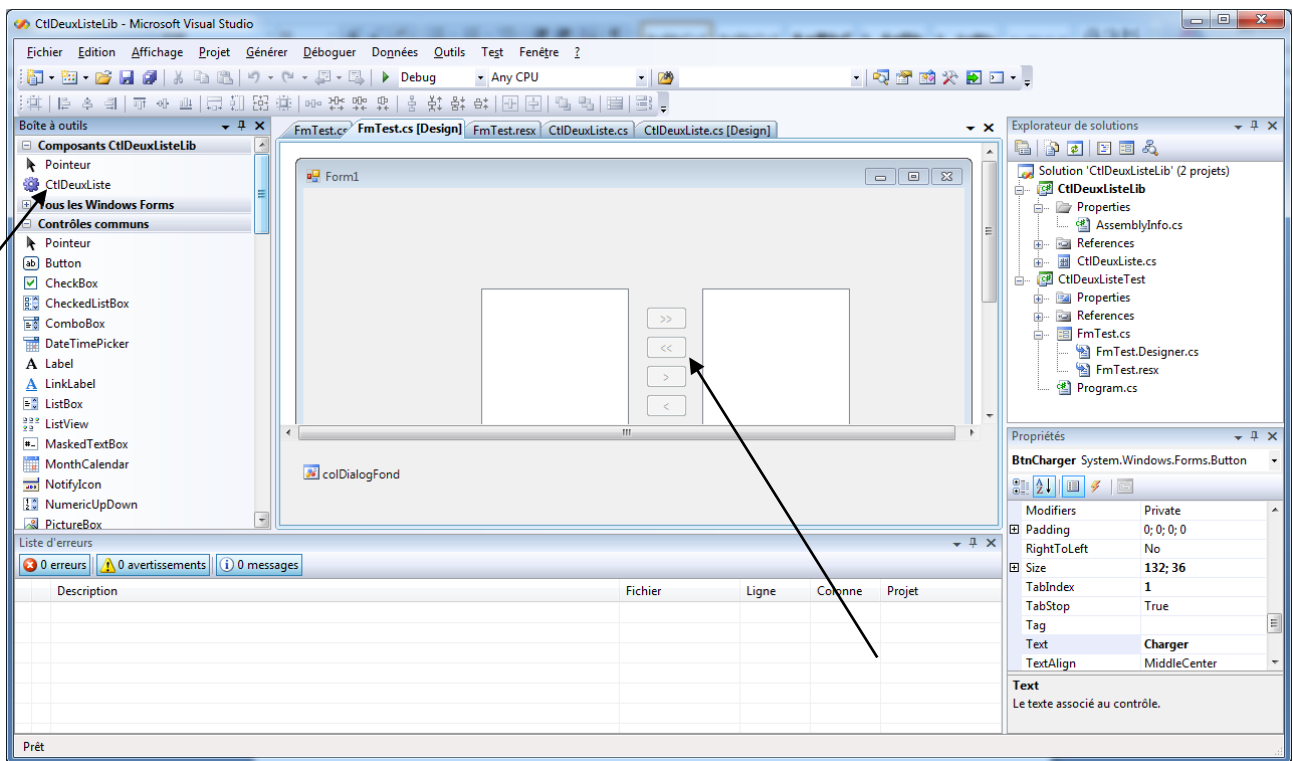
On peut aussi créer son propre projet de test, qui n'est rien d'autre qu'un projet winForm que l'on va intégrer à la solution

Méthodologie :

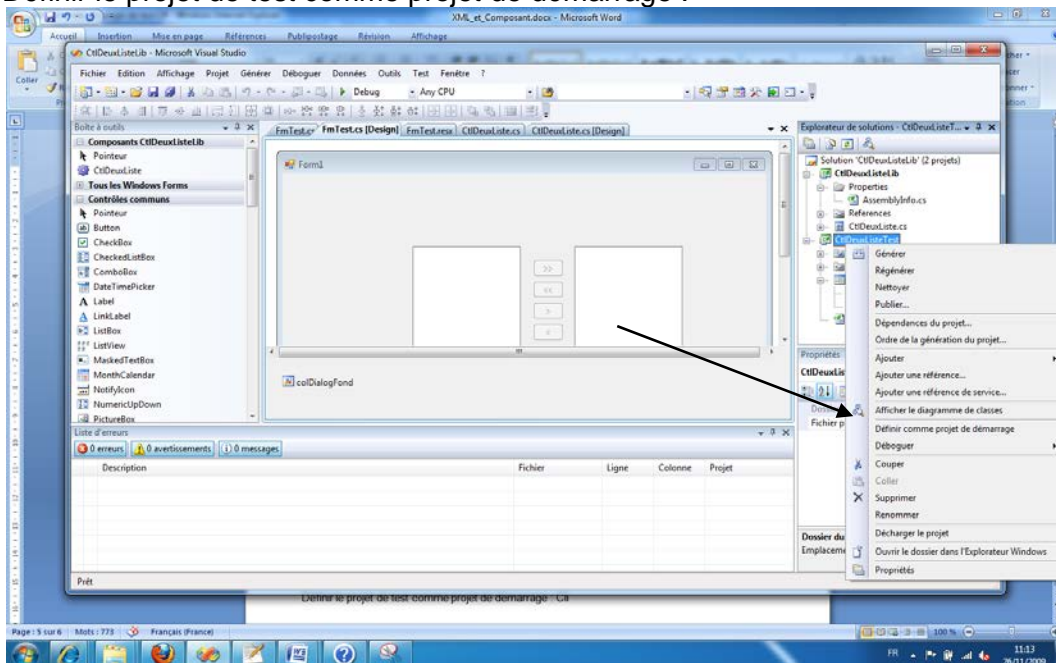
Click Bouton droit sur Solution.... Dans l'explorateur de solution/Ajouter/Nouveau projet.

Attribuer les noms avec la même rigueur que pour un projet applicatif de production.

Une fois que le projet est créé, il faut se positionner en mode design de ce nouveau projet et chercher le composant composite dans la boîte à outils :



Définir le projet de test comme projet de démarrage :

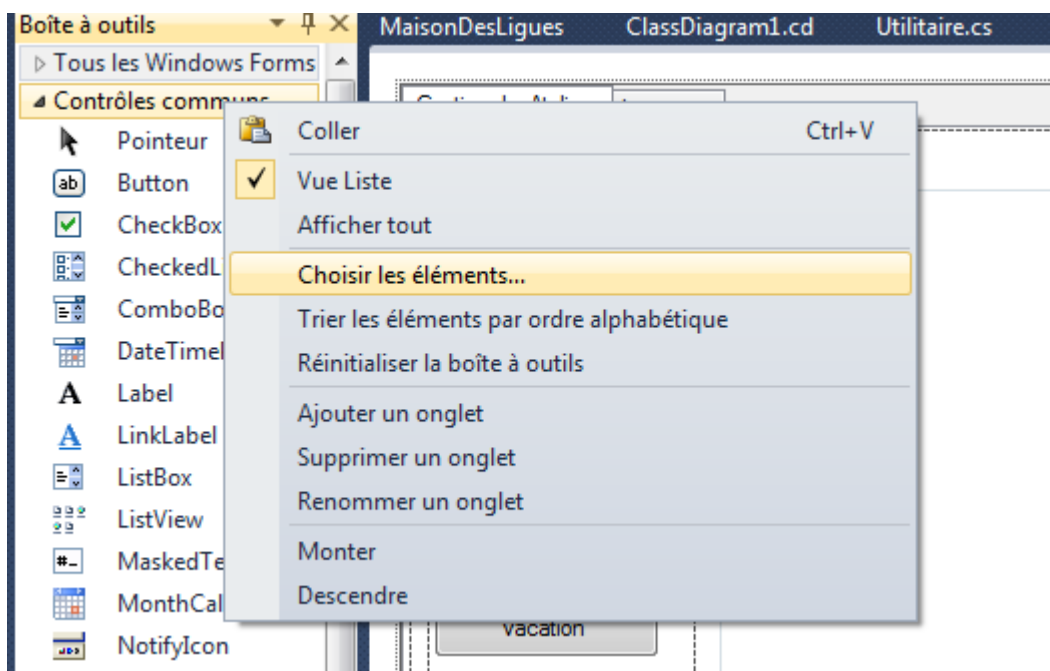


Il ne reste plus qu'à tester.....

5. Mise en service du composant

Lorsque le composant est testé et validé :
compiler votre projet en mode release,
aller chercher la dll générée dans le dossier racineProjet\bin\Release
La copier dans un dossier de votre choix,

Dans la partie design de votre projet, click droit/choisir les éléments
Aller chercher la dll dans le dossier qui l'héberge.



Votre composant apparait alors dans la boite à outils.

On peut aussi supprimer le projet de test de la solution.