

Cours Client Riche Numéro 1



Objectifs : Connaître les éléments de base de l'éditeur C#.
Création d'un convertisseur Francs / Euro.
Gestion des exceptions.

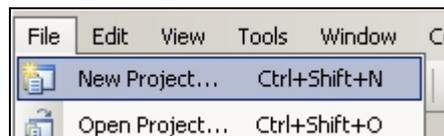
Pré requis : Avoir fait les cours consoles du C#.

Pour ce premier cours, nous allons créer un convertisseur Franc / Euro. Comme éditeur de code j'ai choisi Visual C# Express Edition. Vous pouvez le télécharger chez Microsoft www.microsoft.fr.

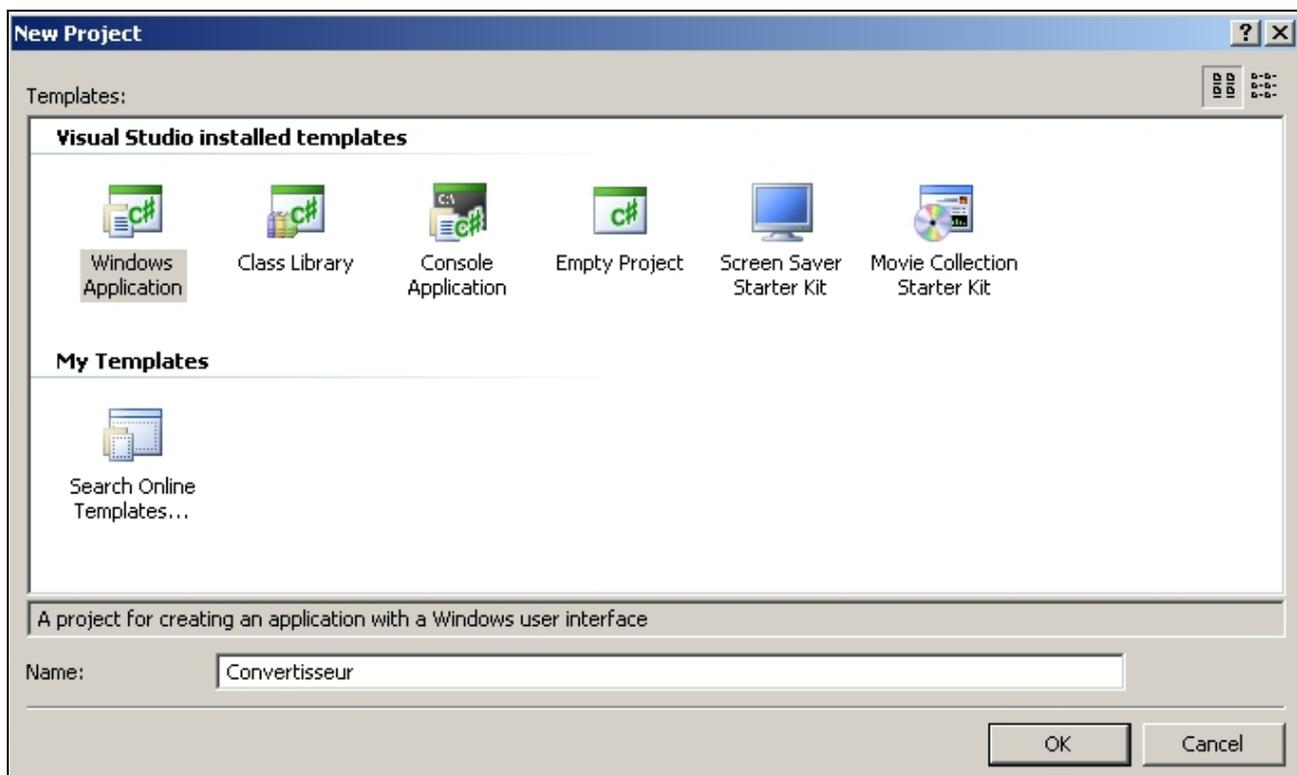


Lancez Visual C# Express Edition en double cliquant sur son icône

La fenêtre d'accueil apparaît, cliquez sur **Files puis New Project.**

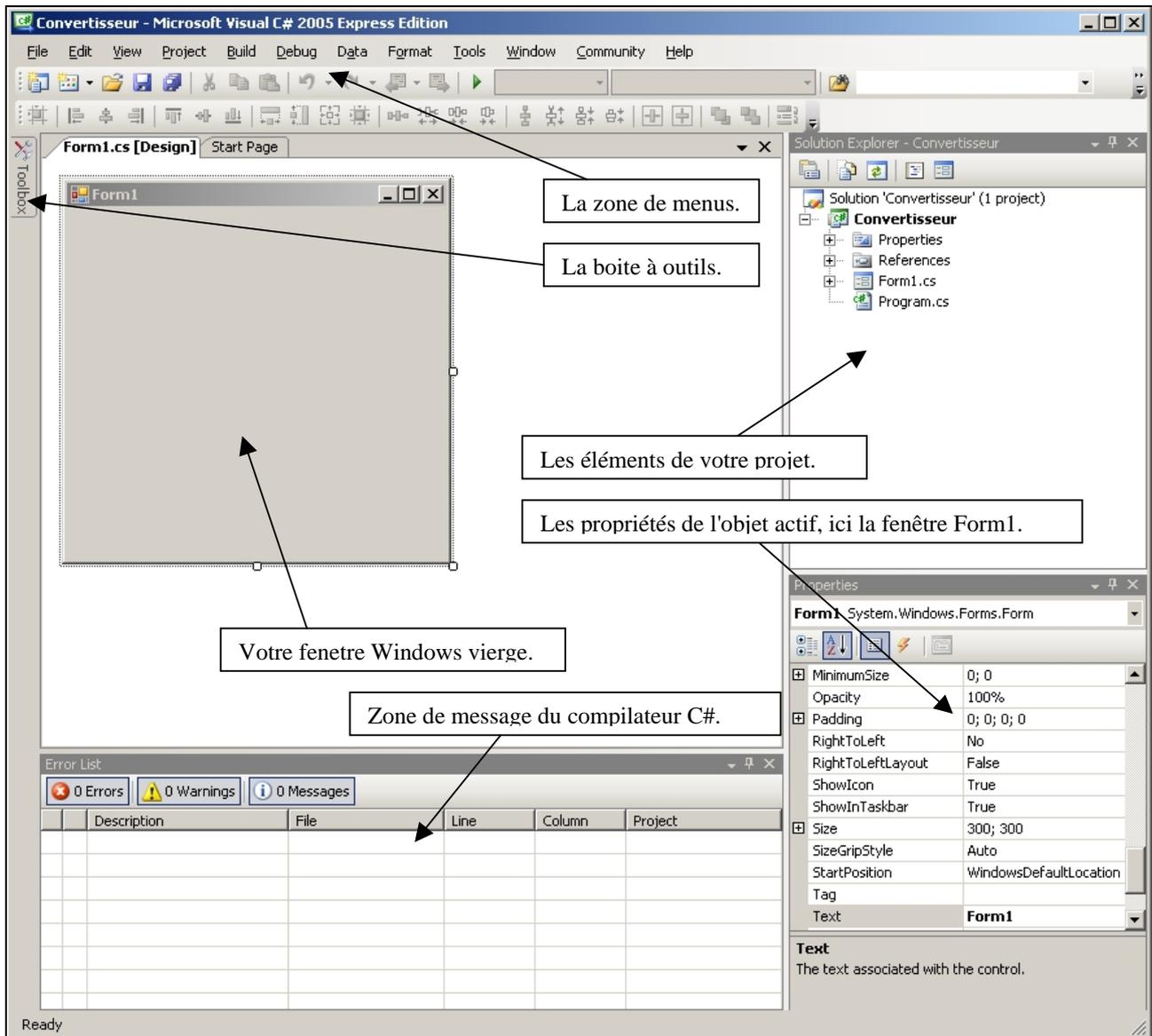


(A l'heure actuelle la version française n'est pas sortie, votre anglais va s'améliorer !)



Cette fenetre détermine les choix de conception de l'exécutable. Comme nous allons travailler dans un environnement fenêtré choisissez Windows Application. Inscrivez « convertisseur » dans la zone Name. Validez en cliquant sur **OK**.

Votre environnement de travail se configure de la sorte :

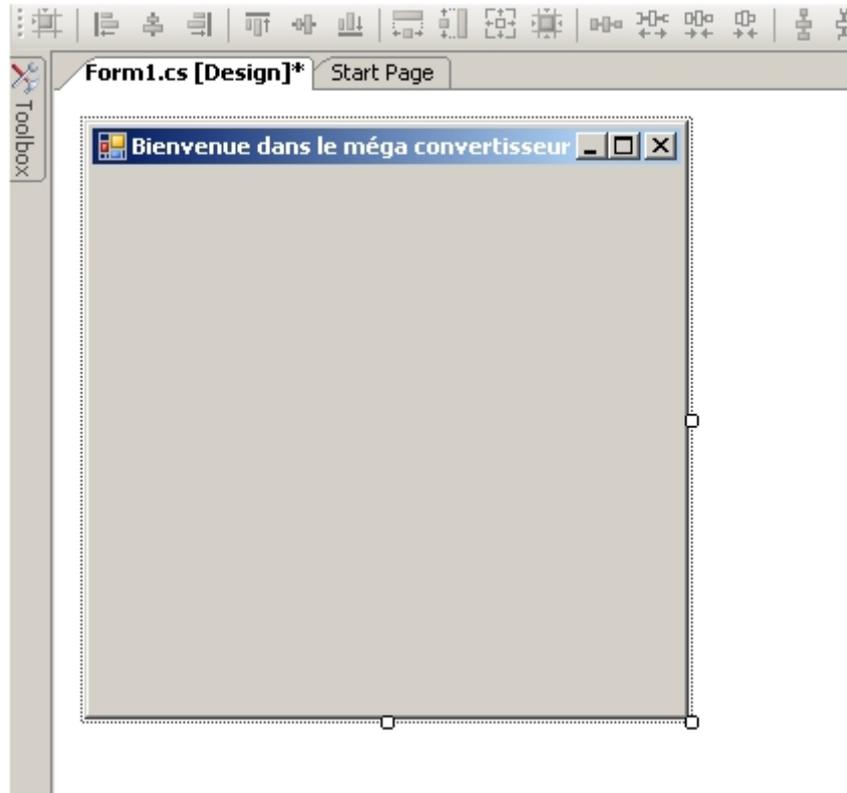


A ce stade nous avons une fenetre vide dans l'éditeur. Il nous reste à lui donner un nom, une taille et définir quelques comportements.

Dans la zone Properties (Propriétés) de la fenetre rechercher l'item **Name**, inscrivez **Depart** à la place de **Form1**.

A l'item **Text** inscrivez "Bienvenue dans le méga convertisseur".

Vous devriez avoir une fenetre vierge ressemblant à ceci :



Intéressons nous à l' **IHM** (**I**nterface **H**omme-**M**achine).

Nous allons faire quelques modification dans les propriétés de la fenêtre :

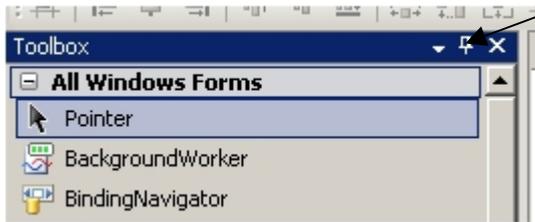
Nous allons faire en sorte que la fenêtre se positionne au centre de l'écran, pour cela recherchez **StartPosition** dans les items et mettez le sur **CenterScreen**.

Vous allez aussi faire en sorte qu'elle ne soit pas redimensionnable, recherchez **FormBorderStyle** et placez le sur **FixedToolsWindow**.

On va fixer la taille, recherchez **Size** cliquez sur la croix à gauche, l'item va ainsi se dérouler laissant voir 2 options : **Width** (Largeur) et **Height** (Hauteur). Fixez la largeur a **310** et la hauteur a **300**.

Remarquez les différences, vous avez maintenant une fenêtre avec un nom, un titre une taille et un comportement défini.

A l'intérieur de cette fenêtre, placez 4 objets : 3 "**champs de saisie**", 3 **Labels** et un bouton (faites un glisser/déposer ou drag & drop des champs dans la fenêtre « Bienvenue dans le méga convertisseur »).
Pour cela ouvrez la Toolbox et punaisez-la ici.



Les champs de saisie sont des **TextBox**, les labels sont des **Labels**, pour le bouton : cherchez (feignants !)

Le premier champ se nommera **franc** et aura comme Texte de label "Zone francs :". Cliquez 2 fois dessus pour en modifier les caractéristiques :

Le second se nommera **taux** et aura comme texte de label "Taux de change :".

Le troisième se nommera **conversion** et aura comme Texte de label "Résultat de la conversion :".

Enfin, Le bouton se nommera **calcul** et aura comme libellé "Calcule".

Pour tous les objets (textbox,label,bouton) dans la propriété **Font** passez la taille à 10.

Voici à quoi doit ressembler votre fenêtre (avec en surimpression le nom des champs) :



Il nous reste à mettre le code correspondant dans le bouton "calcule". Pour cela, faites un clic droit dessus et choisissez "**View code**" ou faite un double clic dessus.

Voici ce qui va apparaître, vous voyez une zone vide "**Calcul_Click(...)**" C'est à l'intérieur de cette classe qui gère le comportement du click sur le bouton Calcul que l'on va écrire le code.

```

Form1.cs* Start Page Form1.cs [Design]*
Convertisseur.Depart
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Convertisseur
{
    public partial class Depart : Form
    {
        public Depart()
        {
            InitializeComponent();
        }

        private void Calcul_Click(object sender, EventArgs e)
        {

        }
    }
}

```

Entrez la séquence comme indiqué ci-dessus.

```

private void Calcul_Click(object sender, EventArgs e)
{
    double Somme = double.Parse(franc.Text);
    double tauxEuro = double.Parse(Taux.Text);
    conversion.Text = (Somme / tauxEuro).ToString();
}

```



N'oubliez pas que le C# est sensible à la casse (différenciation Majuscule/Minuscule). Vérifiez toujours la casse en cas d'erreur de syntaxe.

double Somme : Nous indiquons au compilateur que l'on souhaite disposer d'une variable de type double.

double.Parse(franc.Text) : La textbox **franc** contient une valeur de type string (chaîne de caractère). **Double.Parse ()** a comme rôle de convertir la chaîne passée en paramètre en numérique, ici de type double.

L'inverse (passer de double en string), est réalisé par la séquence "**.ToString()**".



N'oubliez pas que le C# est fortement typé, c'est pour cela que l'on est obligé de convertir les types de variable de façon explicite.

Il est temps d'enregistrer notre projet par le menu **File / Save all**.

Une fois l'enregistrement achevé, nous allons tester le projet, pour cela cliquez sur **Debug / Start debugging**. Vous pouvez aussi appuyer sur la touche F5. Un projet lancé de cette manière est supervisé par le débogueur, s'il s'arrête de façon anormale vous aurez un rapport du débogueur.

On va essayer de ne pas faire planter le programme pour cela dans la zone Francs saisissez 100, dans la zone taux saisissez 6,55957 (mettez bien la virgule) et ensuite cliquez sur le bouton. Normalement le résultat du calcul apparaît.

Maintenant, on va générer une erreur, à la place de 6,55957 saisissez 6.55957.

Cliquez sur le bouton ! Boummm !!!

Le débogueur se met en route et essaye de vous aider, je vous laisse le soin de regarder dans la zone des messages.

Fermez le débogueur sans paniquer sur la suite des messages !

La question que l'on peut se poser est la suivante :

Peut-on intercepter les erreurs avant qu'elles ne se produisent ?

La réponse est oui, de la façon suivante : on va attraper les erreurs (ou exceptions) !

Modifiez le code ainsi :

```
private void Calcul_Click(object sender, EventArgs e)
{
    try
    {
        double Somme = double.Parse(franc.Text);
        double tauxEuro = double.Parse(Taux.Text);
        conversion.Text = (Somme / tauxEuro).ToString();
    }
    catch (System.Exception)
    {
        conversion.Text = "Impossible de convertir";
    }
}
```

Globalement, dans le bloc try on place le code à exécuter, dans le bloc catch le code traitant d'erreur.

Réessayez de lancer le projet en mettant une valeur erronée, vous voyez que le comportement de votre programme est plus conforme, plus sécurisé, donc plus professionnel.

Bravo à vous pour être parvenu jusqu'ici. Pour vous reposer, vous pouvez faire le petit exo ci-dessous !

Exercice applicatif.

Créez une application « EXO1 » contenant :

- Une fenêtre.
- 3 champs de type Textbox.
- 3 Libellés.
- 1 bouton.

Lorsque l'utilisateur clique sur le bouton, les champs textes se remplissent avec votre nom dans le premier champ, votre prénom dans le deuxième champ texte et votre année de naissance dans le troisième champ.

Vous avez toute liberté en ce qui concerne l'ergonomie, du nommage des champs, de la fenêtre, du bouton. Je vous rappelle que tout le code doit être contenu dans le bouton.