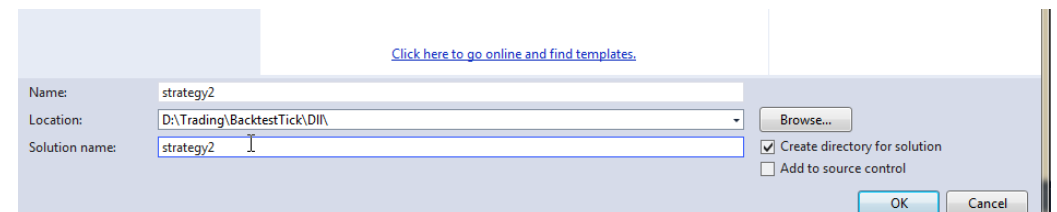
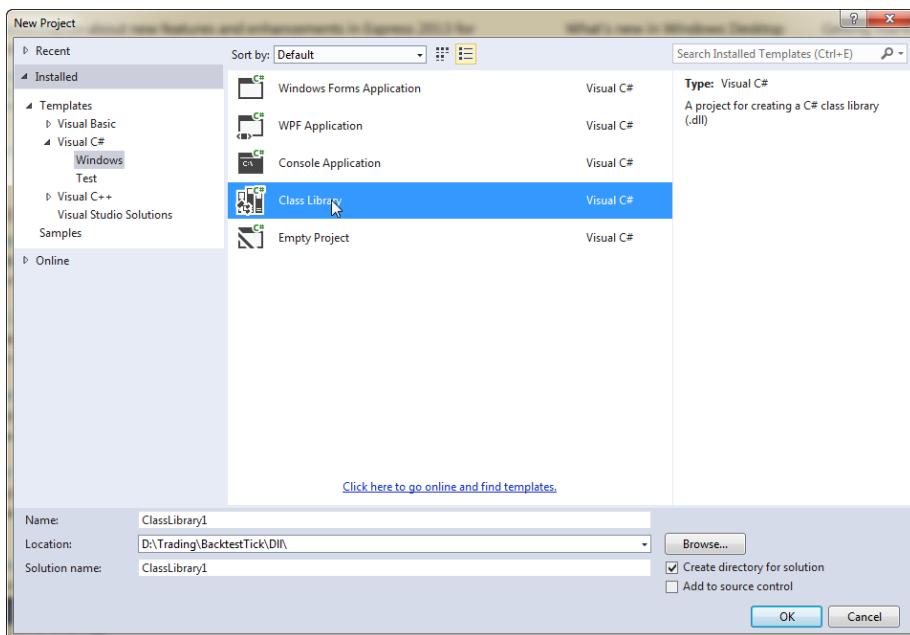
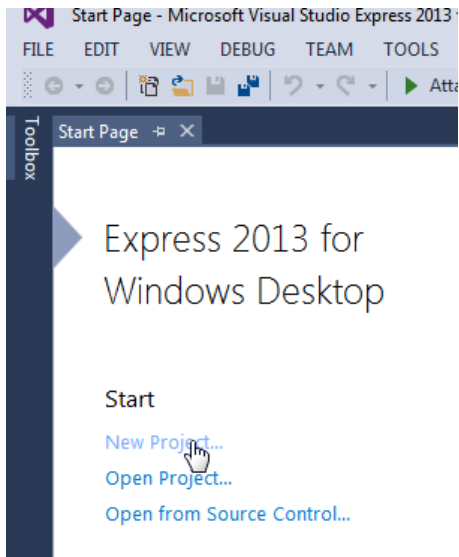
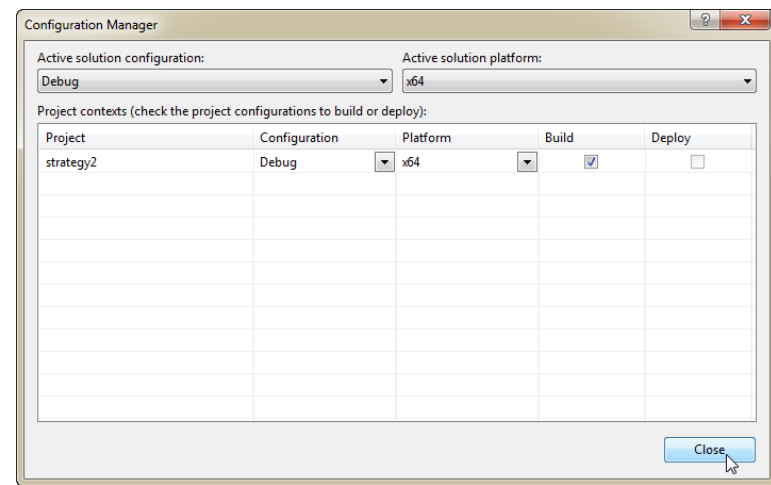
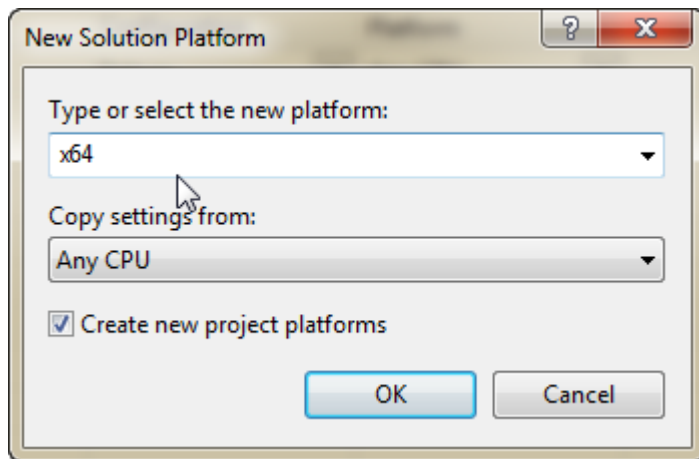
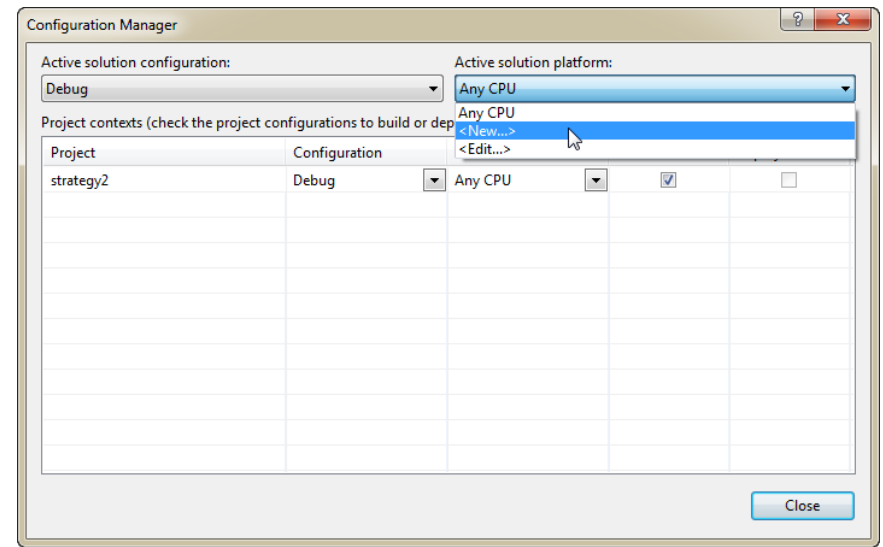
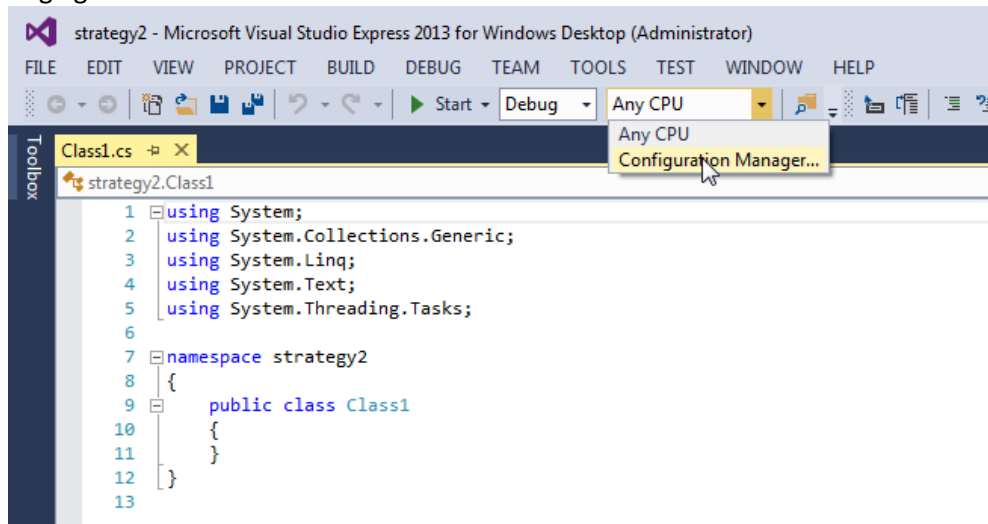


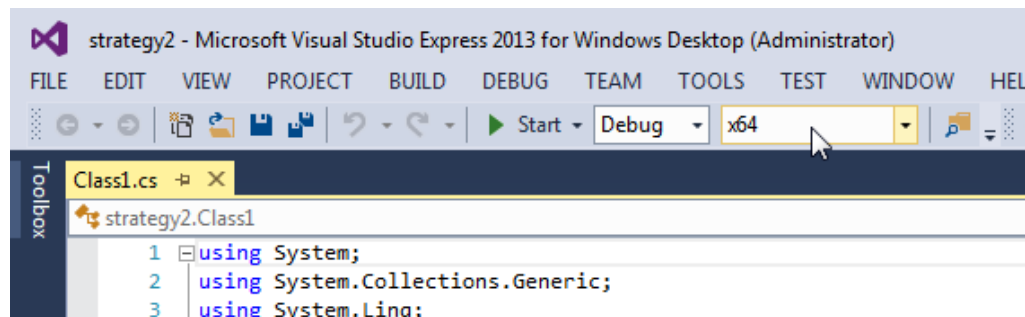
Tutorial pour créer une nouvelle dll de stratégie (sous Visual Studio Express 2013)



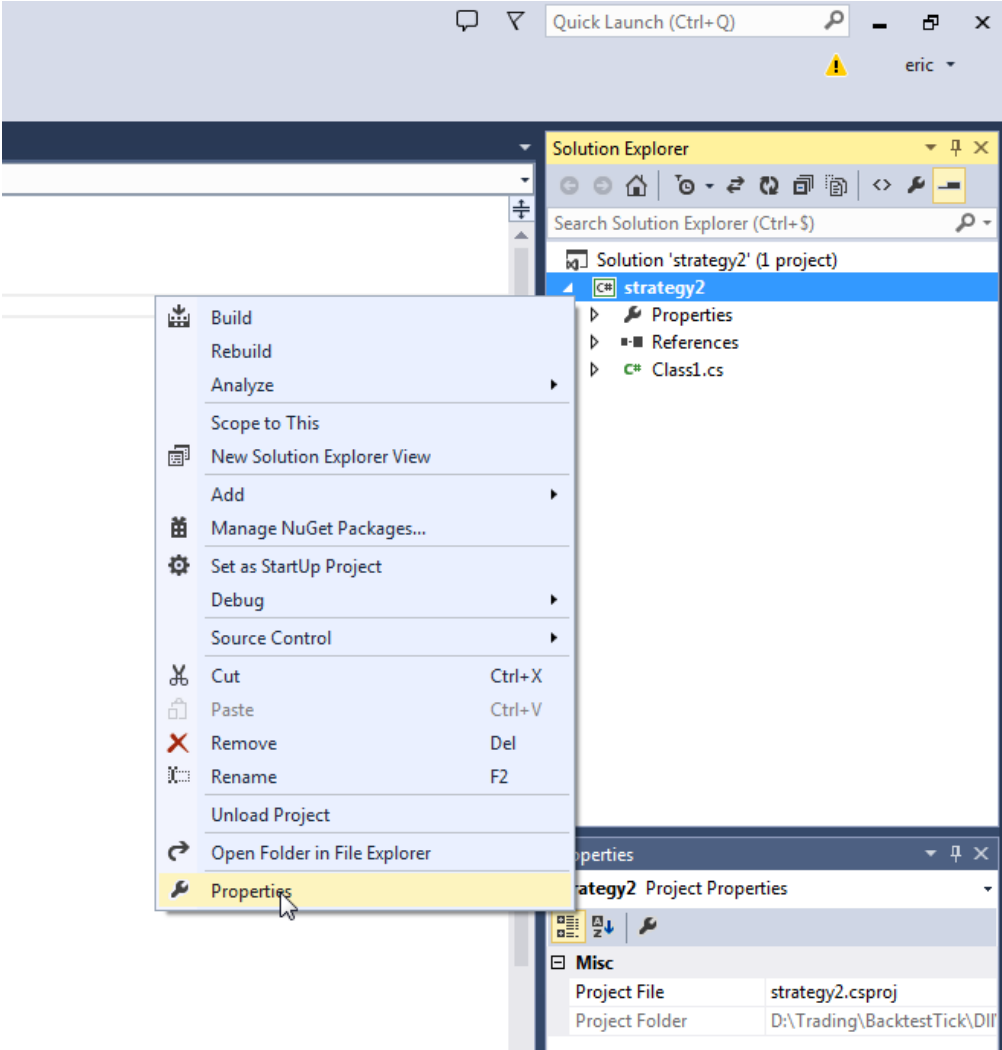
Réglages du CPU

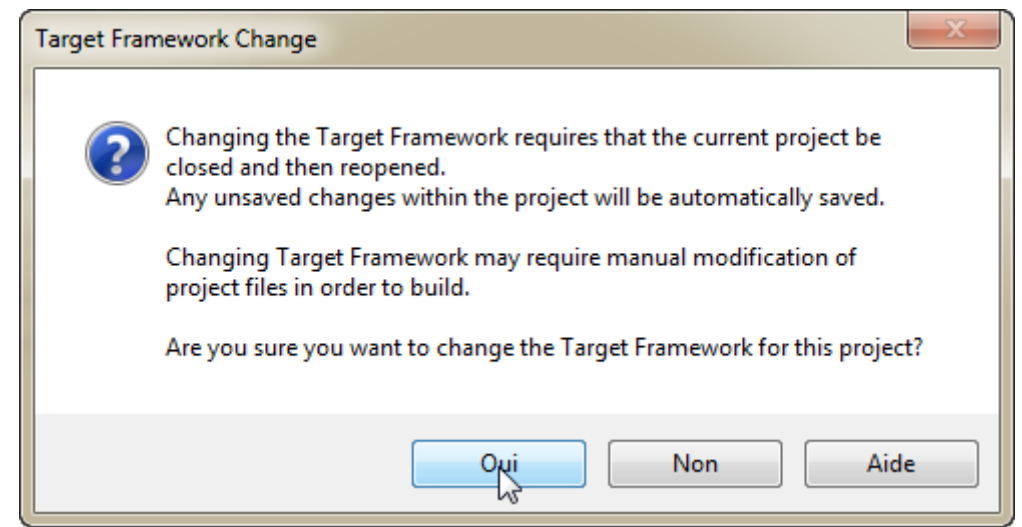
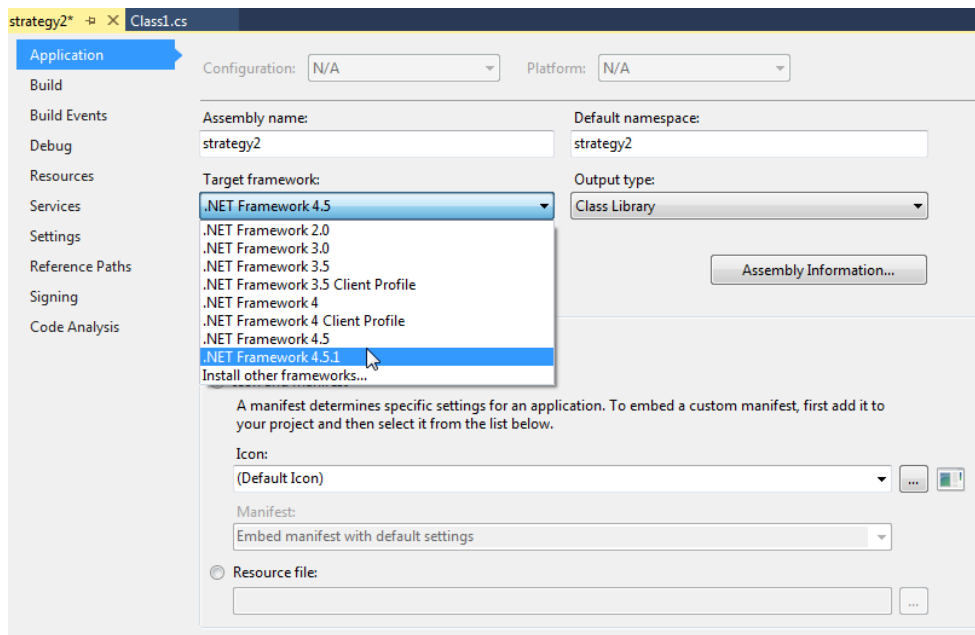


On arrive là



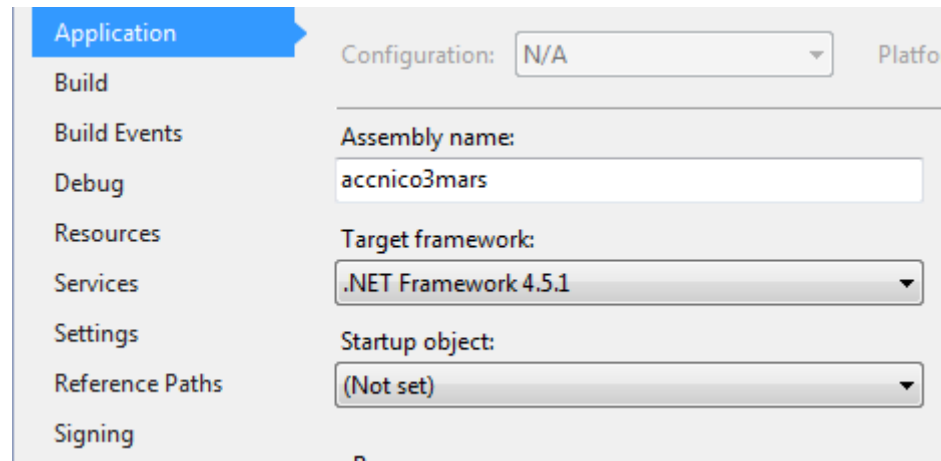
Réglage du framework (il faut qu'il soit le même que dans backtest.exe)



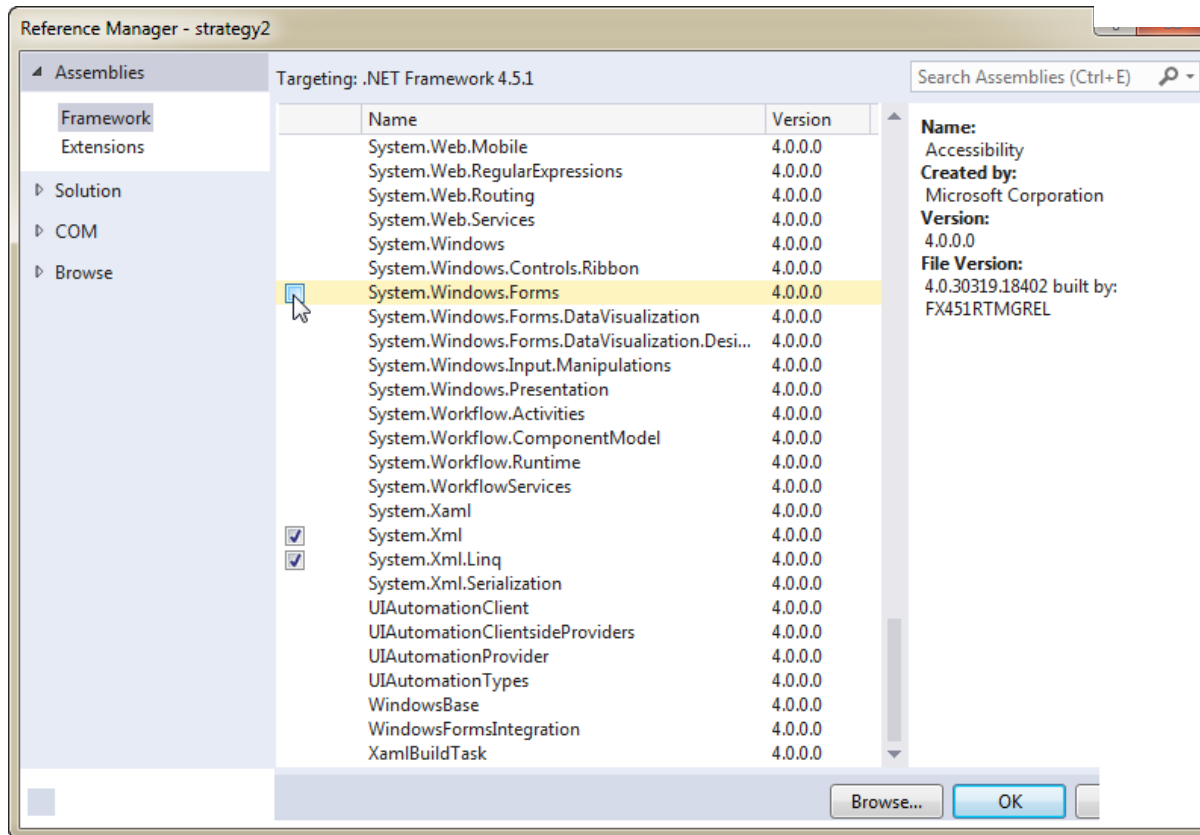
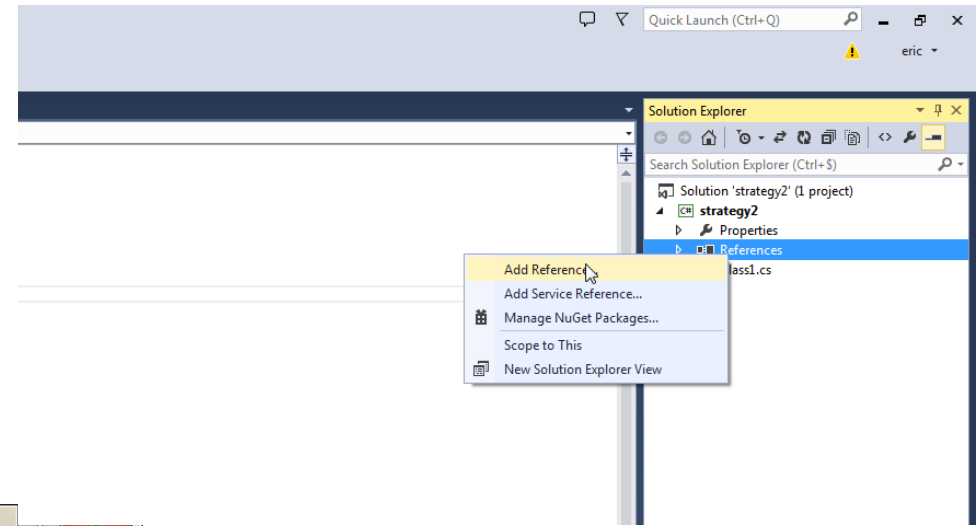
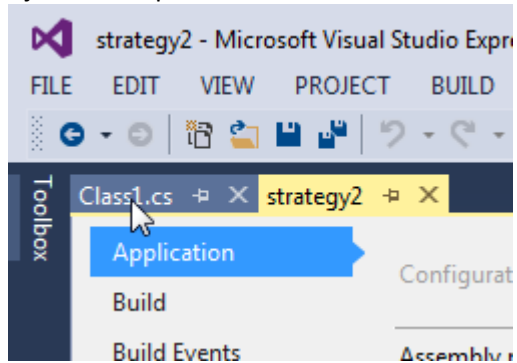


Sur cette même page renommer l'assembly name avec le nom de la stratégie

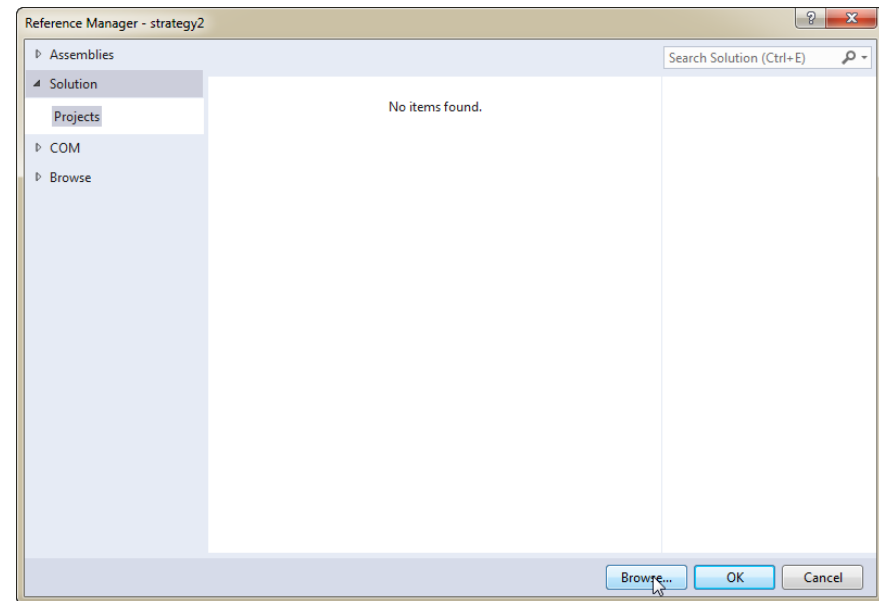
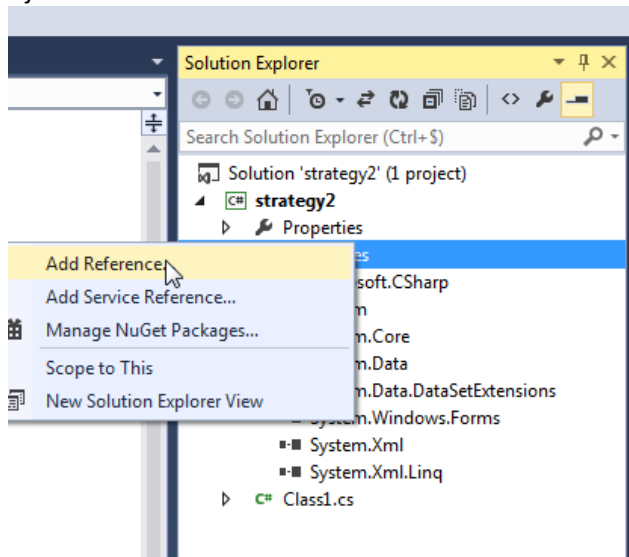
ça sera le nom de la dll générée à la compilation



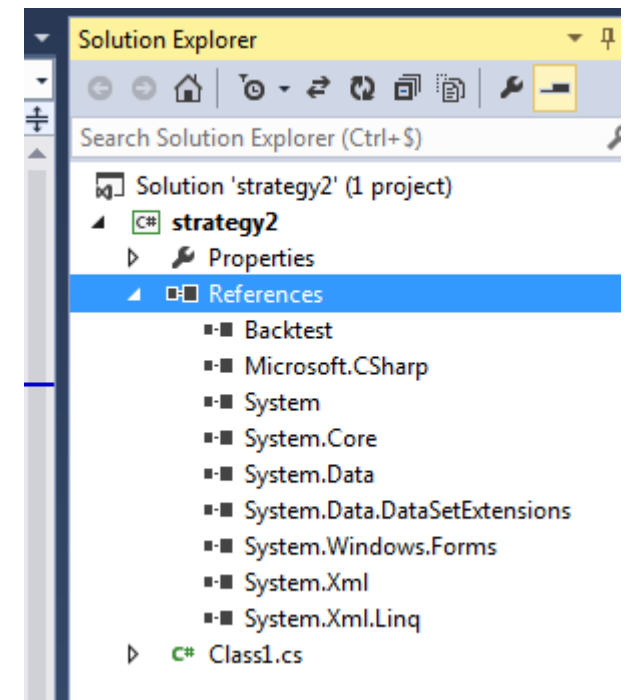
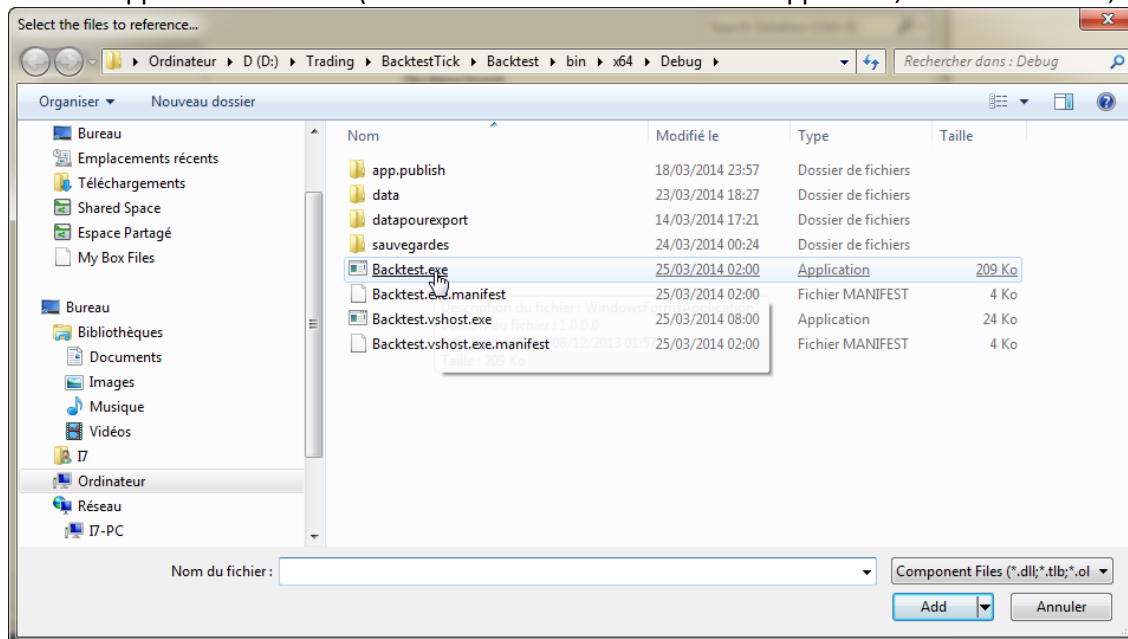
Ajout d'une première référence



Ajout d'une 2^{ème} référence



Choisir l'application backtest (dans son environnement de développement, ou celle installée, à vérifier si ça marche)



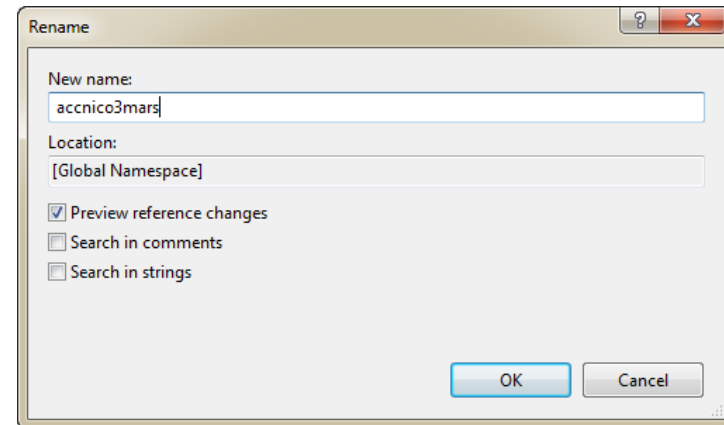
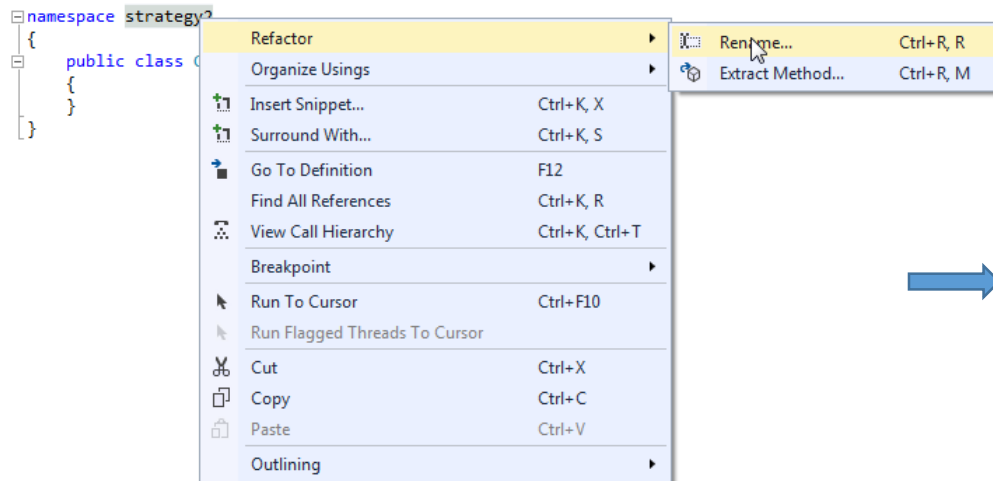
Modifier les « using »

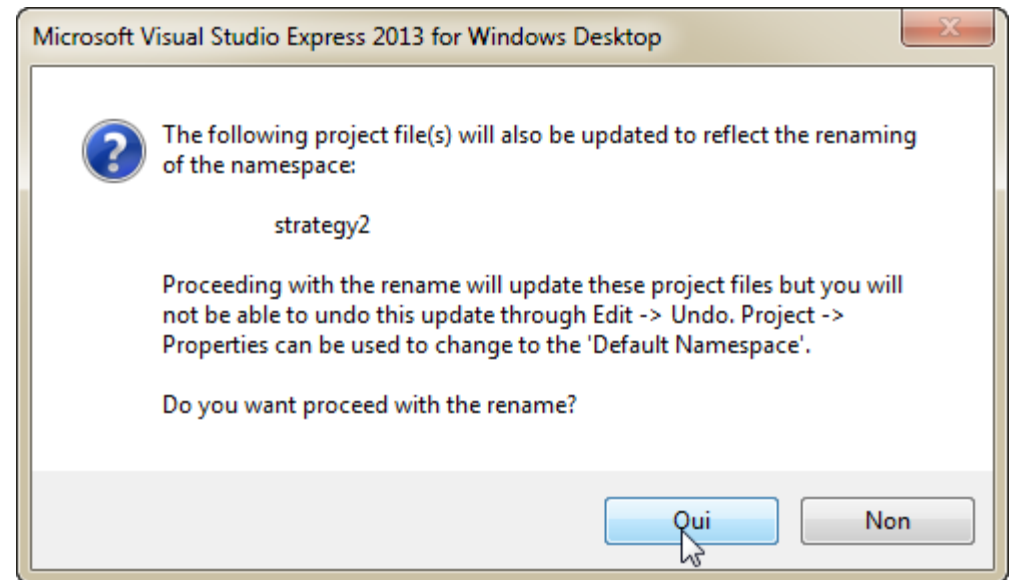
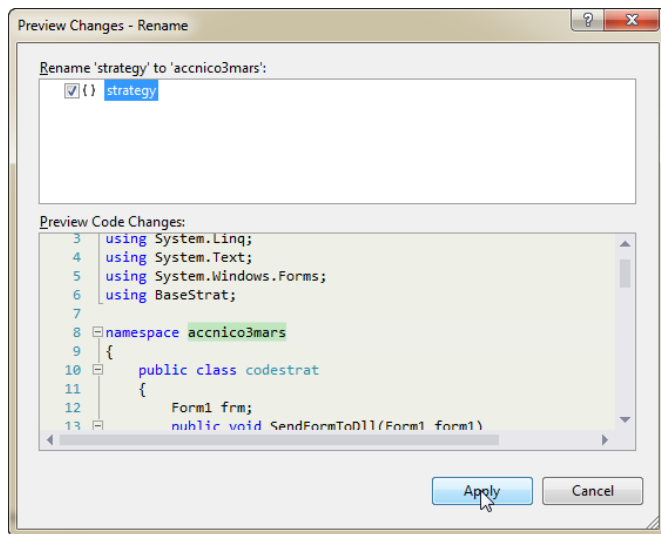
```
Class1.cs  strategy2*
strategy2.Class1
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace strategy2
8 {
9     public class Class1
10    {
11    }
12 }
13
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using BaseStrat;
```

Renommer le namespace -> nom de la stratégie





Renommer la classe -> nom de la stratégie

```
namespace accnico3mars
{
    public class codestrat
    {
    }
}
```

→

```
namespace accnico3mars
{
    public class Class1
    {
        Form1 frm;
    }
}
```

Ajouter ce code dans la classe codestrat

```
//-----début du code
    Form1 frm;
    public void SendFormToDll(Form1 form1)
    {
        frm=form1;
    }
    public void GetParamsFromDll()
    {
        frm.LabelNomStrat.Text = "accnico3mars";

        string[] param = new string[10];
        param[0] = "param;value;min;max;step";
        param[1] = "deb;0;0;0;100";
        param[2] = "fin;2400;2400;2400;100";
        param[3] = "Prfb;5;5;5;1";
        param[4] = "SL;15;15;15;1";
        param[5] = "BE;3;3;3;1";
        param[6] = "BEprf;1;1;1;1";
        param[7] = "accb;1;1;1;1";
        param[8] = "slip;0;0;0;1";
        param[9] = "acc;4;4;4;1";

        for (int numparam = 0; numparam <= 9; numparam++)
        {
            string[] split = param[numparam].Split(';');
            frm.dataGridView2.Rows.Add(split[0],split[1],split[2],split[3],split[4]);
        }
    }

    public void strategy(double p1, double p2, double p3, double p4, double p5, double p6, double p7, double p8, double p9)
    {
        //le code de la strat, voir plus bas
    }
}
//-----fin du code
```



```
namespace accnico3mars
{
    public class codestrat
    {
        Form1 frm;
        public void SendFormToDll(Form1 form1)
        {
            frm=form1;
        }
        public void GetParamsFromDll()
        {
            frm.LabelNomStrat.Text = "accnico3mars";

            string[] param = new string[10];
            param[0] = "param;value;min;max;step";
            param[1] = "deb;0;0;0;100";
            param[2] = "fin;2400;2400;2400;100";
            param[3] = "Prfb;5;5;5;1";
            param[4] = "SL;15;15;15;1";
            param[5] = "BE;3;3;3;1";
            param[6] = "BEprf;1;1;1;1";
            param[7] = "accb;1;1;1;1";
            param[8] = "slip;0;0;0;1";
            param[9] = "acc;4;4;4;1";

            for (int numparam = 0; numparam <= 9; numparam++)
            {
                string[] split = param[numparam].Split(';');
                for (int numitem = 0; numitem <= 4; numitem++)
                {
                    frm.dataGridView2.Rows[numparam].Cells[numitem].Value = split[numitem];
                }
            }
        }

        public void strategy(double p1, double p2, double p3, double p4, double p5, double p6, double p7, double p8, double p9)
        {
            //le code
        }
    }
}
```

Nom de la stratégie

Les paramètres
param[0] donne le format et ne doit pas être changé
param[1] et param[2] sont réservés aux heures de début et de fin du trading
slip est réservé au slippage
les autres paramètres sont libres

Fonction contenant le code de la stratégie, son nom
doit toujours être « strategy »

Exemple du code acc3mars

Note : L'objet frm représente la form1 de l'application backtest, qui permet à la dll de faire des entrées et sorties sur l'application backtest, et de connaître tous ses objets déclarés public (variables, fonctions, composants...)

Utilisation :

int abc = frm.variable écrit le contenu de variable de frm dans abc de la dll

frm.variable = abc écrit le contenu de abc de la dll dans la variable de frm

Les objets principaux de frm (remplis lorsque qu'on charge un fichier de tickdata dans backtest.exe)

- datetime est une liste d'objets DateTime (0 based index liste) contenant les temps des tickdata : Year,Month,Day,Hour,Minute,Secondes,Millisecondes(000 pour fxcms)
- bid ou ask est une liste d'objet double (0 based index liste) contenant les bid (ask) des tickdata

il y a quelques autres objets ou fonctions dont l'utilisation est détaillée dans le code

pour le code complet, voir la feuille Class1.cs (sur le ftp)