

Enoncé complémentaire 1

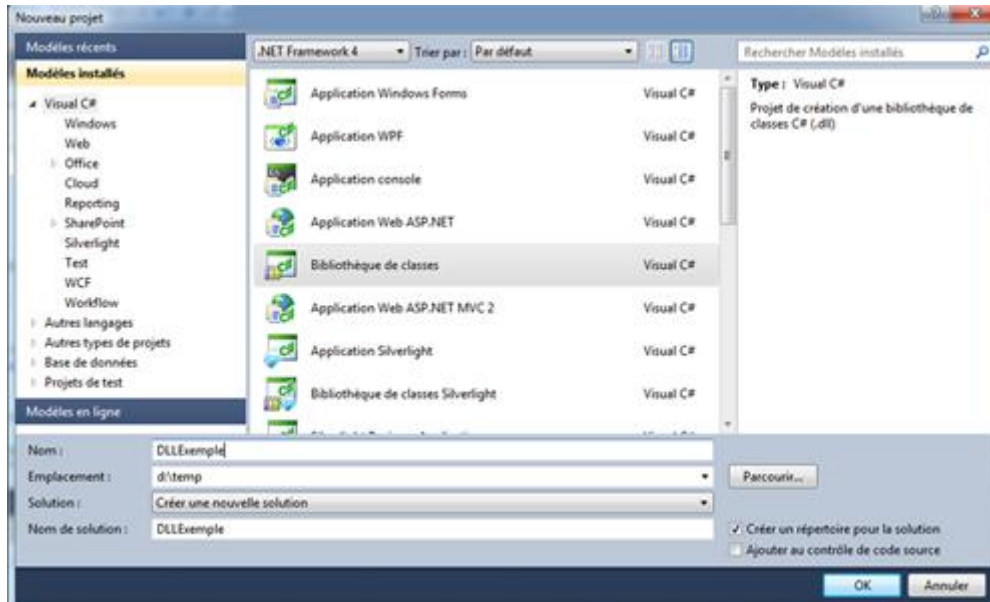
Tips :

1.	Générer et utiliser une DLL.....	2
1.1.	Créer la DLL.....	2
1.2.	Créer un projet qui utilise la DLL	2
2.	Stocker une image dans une table	5
2.1.	Créer la table	5
2.2.	Stocker et lire l'image sous forme de blob.....	6

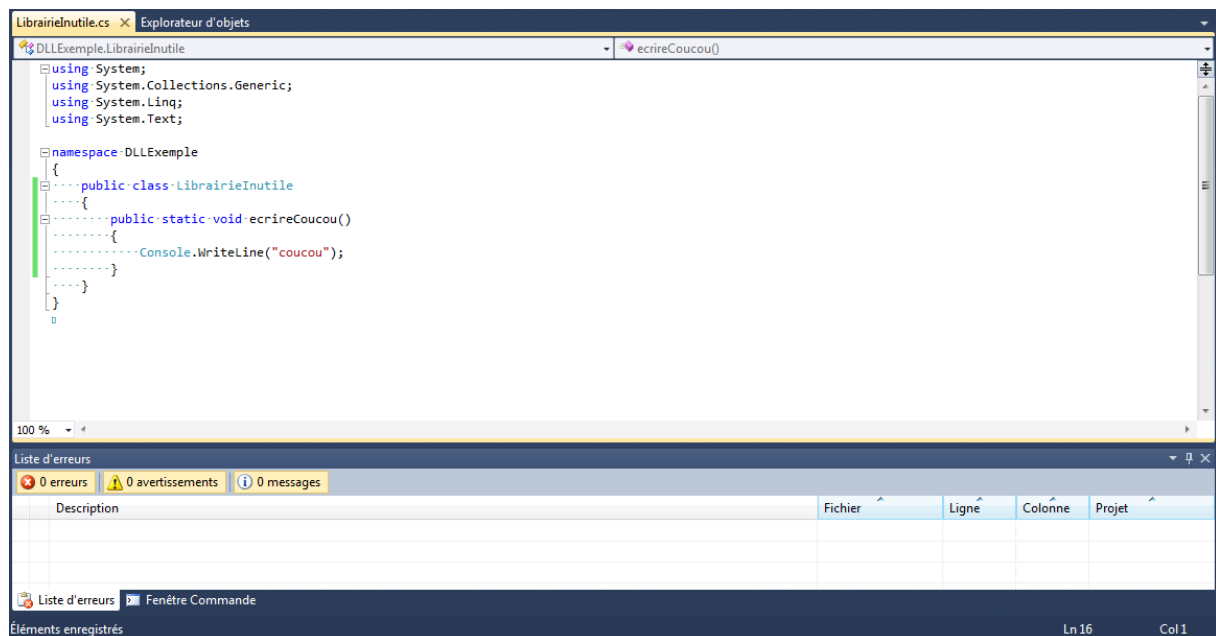
1. Générer et utiliser une DLL

1.1. Créer la DLL

Créer un nouveau projet de type Bibliothèque de classes.



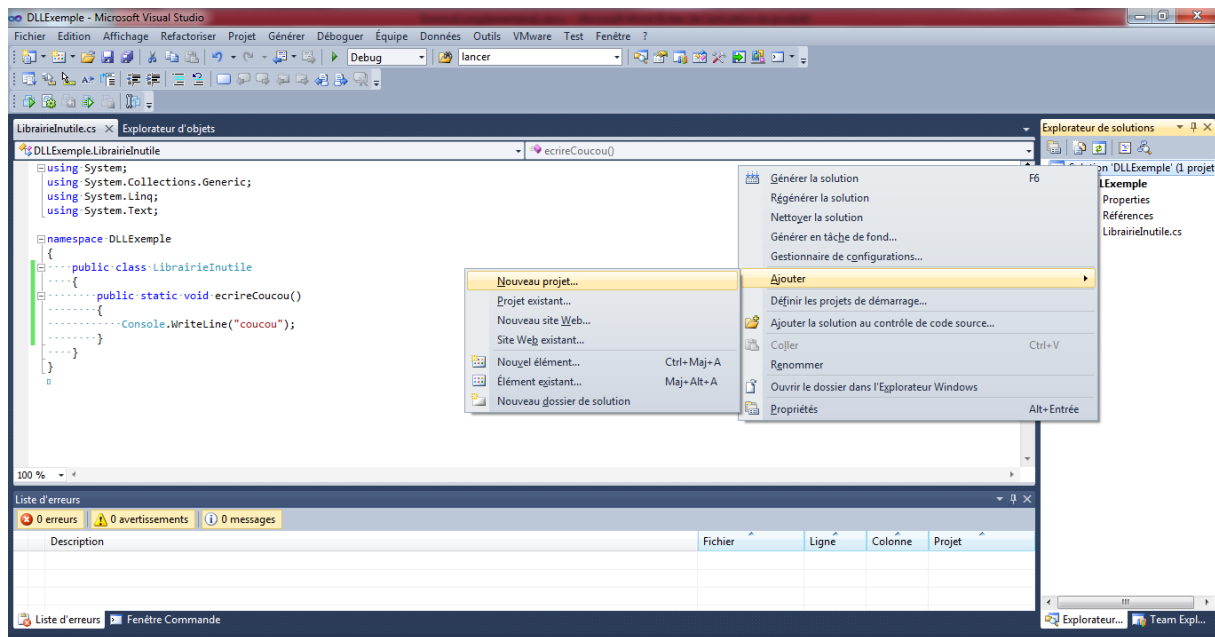
Créer une méthode quelconque.



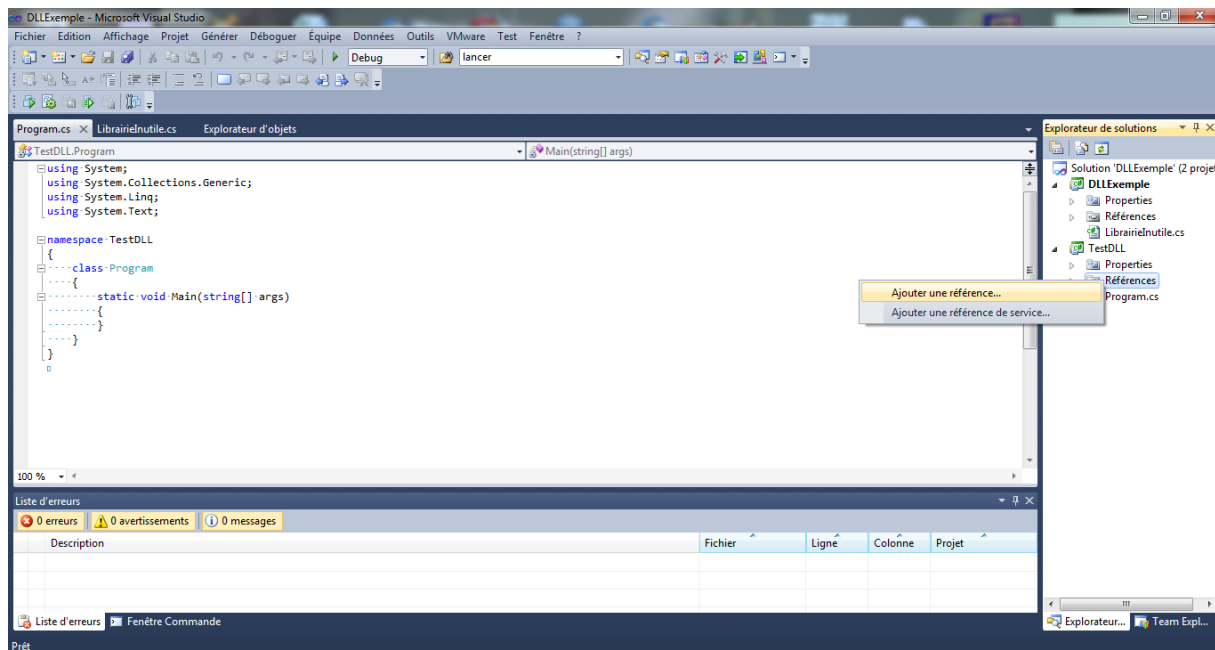
Générer la solution (F6).

1.2. Créer un projet qui utilise la DLL

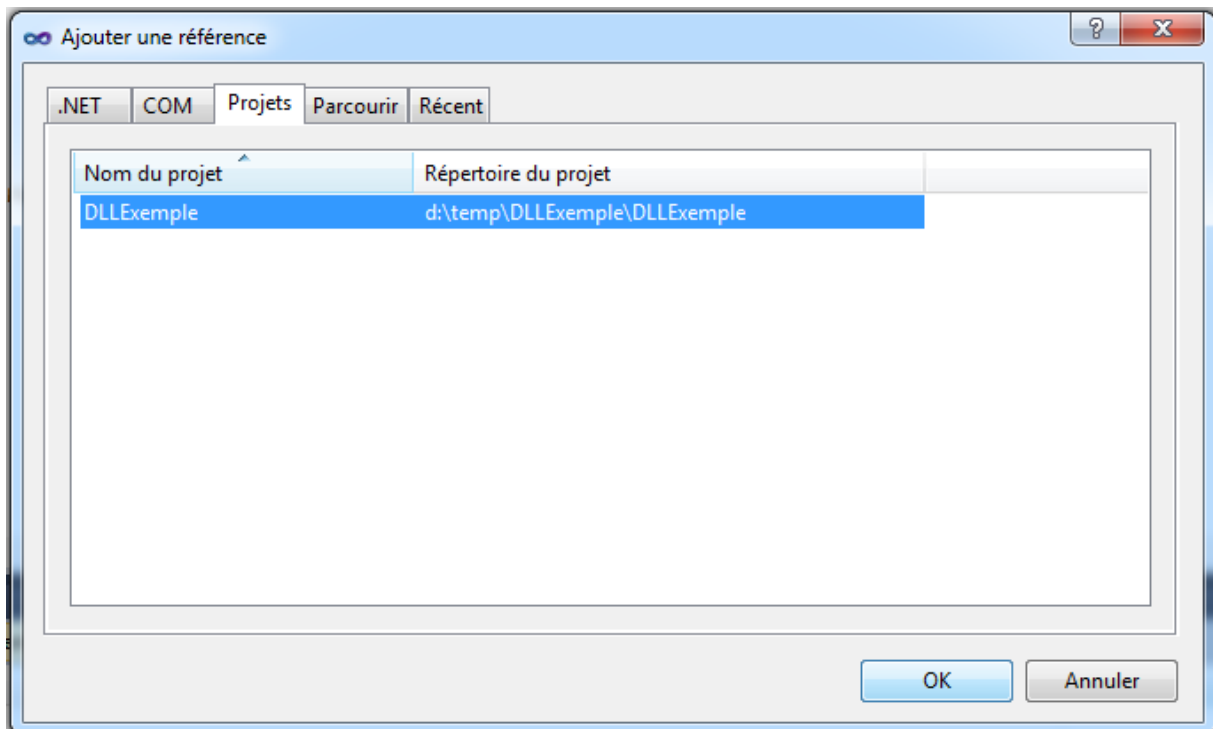
Dans la même solution, on ajoute un nouveau projet de type Console pour tester notre DLL (clique-droit sur la Solution->Ajouter->Nouveau Projet):



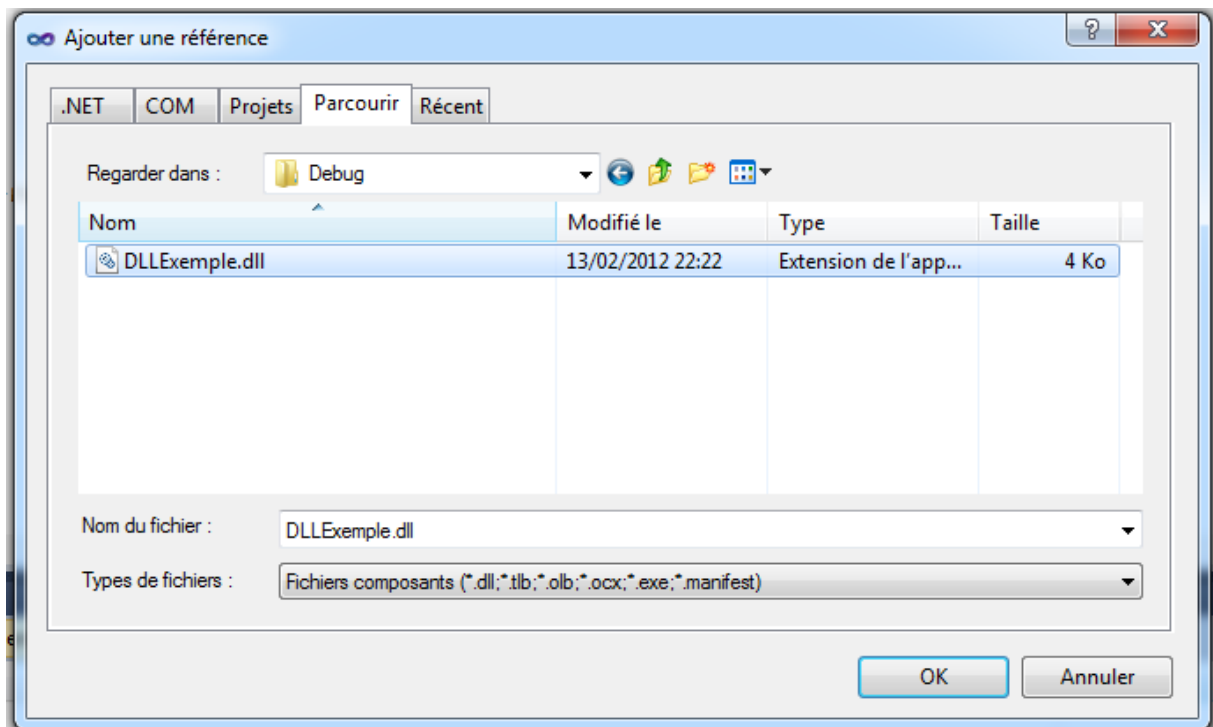
On ajoute une référence sur notre DLL dans notre nouveau projet console (clique-droit sur Références->Ajouter une référence) :



Il est possible d'ajouter directement notre projet de DLL:



Ou de cliquer sur parcourir pour récupérer la DLL générée par notre projet (attention dans ce cas à régénérer la DLL à chaque modification):



On peut donc utiliser les classes de notre DLL en rajoutant le using approprié (ici « using DLLExemple ») :

```
Program.cs X LibrairieInutile.cs Explorateur d'objets
TestDLL.Program Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using DLLExemple;

namespace TestDLL
{
    class Program
    {
        static void Main(string[] args)
        {
            LibrairieInutile.ecrireCoucou();
        }
    }
}
```

2. Stocker une image dans une table

2.1. Créer la table

On crée d'abord une table qui contiendra nos images. L'image en elle-même sera stockée dans la table sous la forme de blob, c'est-à-dire de tableau de byte. Notre table contiendra donc un id, la taille de notre fichier image ainsi que le blob :

	Column Name	Data Type	Allow Nulls
PK	id	nchar(50)	<input type="checkbox"/>
	size	int	<input type="checkbox"/>
	blob	image	<input type="checkbox"/>
			<input type="checkbox"/>

2.3. Stocker et lire l'image sous forme de blob

Voici la fonction permettant de sauver une image dans la base à l'aide d'ADO.NET :

```
public void addImage(String imageID, byte[]image){
    try
    {
        // connexion au serveur
        connexion.Open();

        // construit la requête
        SqlCommand ajoutImage = new SqlCommand(
            "INSERT INTO Image (id, blob, size) " +
            "VALUES(@id, @Blob, @size)", connexion);
        ajoutImage.Parameters.Add("@id", SqlDbType.VarChar, imageID.Length).Value
        = imageID;
        ajoutImage.Parameters.Add("@Blob", SqlDbType.Image, image.Length).Value
        = image;
        ajoutImage.Parameters.Add("@size", SqlDbType.Int).Value = image.Length;

        // execution de la requête
        ajoutImage.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        Console.WriteLine("Erreur :" + e.Message);
    }
    finally
    {
        // dans tous les cas on ferme la connexion
        connexion.Close();
    }
}
```

Et voici la méthode permettant de la récupérer avec un DataReader :

```
// récupération d'une image de la base à l'aide d'un DataReader
public byte[] getImage(String imageID)
{
    byte[] blob = null;
    try
    {
        // connexion au serveur
        connexion.Open();

        // construit la requête
        SqlCommand getImage = new SqlCommand(
            "SELECT id,size, blob " +
            "FROM Image " +
            "WHERE id = @id", connexion);
        getImage.Parameters.Add("@id", SqlDbType.VarChar, imageID.Length).Value =
            imageID;

        // exécution de la requête et création du reader
        SqlDataReader myReader =
            getImage.ExecuteReader(CommandBehavior.SequentialAccess);

        if (myReader.Read())
        {
            // lit la taille du blob
            int size = myReader.GetInt32(1);
            blob = new byte[size];

            // récupère le blob de la BDD et le copie dans la variable blob
            myReader.GetBytes(2, 0, blob, 0, size);
        }
    }
    catch (Exception e)
    {
        Console.WriteLine("Erreur :" + e.Message);
    } finally {
        // dans tous les cas on ferme la connexion
        connexion.Close();
    }
    return blob;
}
```