

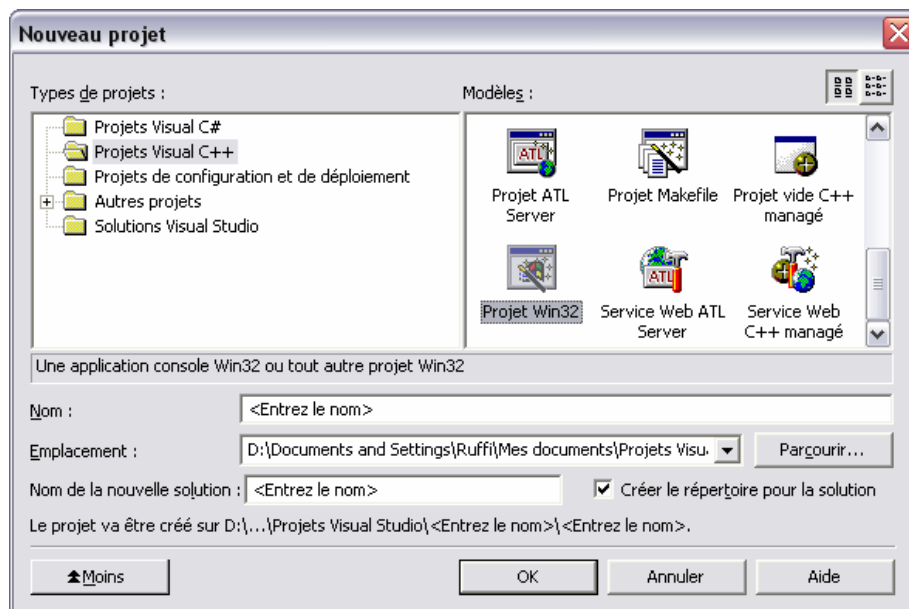
Création d'une dll avec Visual C++

Introduction

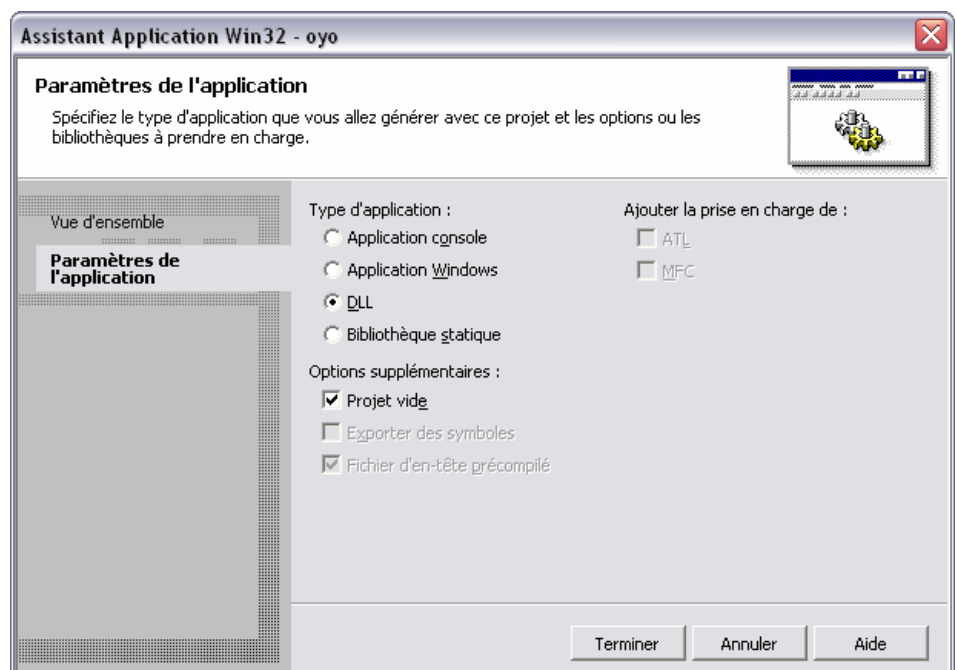
Pour mon premier tutorial, je vais vous montrer comment faire des dll, pour exporter une classe. Vous allez voir, c'est facile à faire☺. Pour faire le moteur du jeu SELRA, bob et moi avons conclu que l'utilisation de dll serait importante. D'où ce tutorial, histoire de partager nos connaissances.

Création de la DLL

Lancer le Visual, créez un nouveau projet, de type Projet Win32 :



Puis sélectionnez DLL et Projet vide



Voici le code du fichier .h :

```
#if defined( USE_DLL )
#define MYDLL __declspec( dllimport )
#elif defined( COMPILE_DLL )
#define MYDLL __declspec( dllexport )
#else
#define MYDLL /* rien */
#endif

class MYDLL cMyClass
{
public:
    cMyClass();
    void setA(int _a);
    void add(int _a);
    void sub(int _a);
    int getA();
private:
    int a;
};
```

La partie du début permet d'utiliser le même fichier lors de la création que lors de son utilisation. Il suffira de spécifier si on souhaite créer la dll en faisant #define COMPILE_DLL ou si on souhaite l'utiliser en faisant #define USE_DLL.

Ceci crée la macro nécessaire. Cette macro doit être placée devant toute déclaration de classe ou de fonction devant être exportée (ou importée).

Voici le code du fichier .cpp :

```
#define COMPILE_DLL

#include "testDLL.h"

MYDLL cMyClass::cMyClass()
: a(0)
{
}

MYDLL void cMyClass::setA(int _a)
{
    a = _a;
}

MYDLL void cMyClass::add(int _a)
{
    a += _a;
}
```

```
MYDLL void cMyClass::sub(int _a)
{
    a -= _a;
}

MYDLL int cMyClass::getA()
{
    return a;
}
```

C'est juste une classe de test, faisant rien de spécial. Il ne faut pas oublier de bien définir `COMPILE_DLL`, et de mettre la macro `MYDLL` devant chaque implémentation de la classe.

Lorsque vous compilerez, il vous créera un fichier `.dll` et un fichier `.lib`

Utilisation de la dll

C'est bien gentil d'avoir une dll, mais encore faut-il savoir l'utiliser.
C'est encore plus simple que la création.

Voici le code utilisé pour utiliser la dll.

Il a été fait dans un nouveau projet console application :

```
#include <iostream>

#define USE_DLL
#include "testDLL.h"

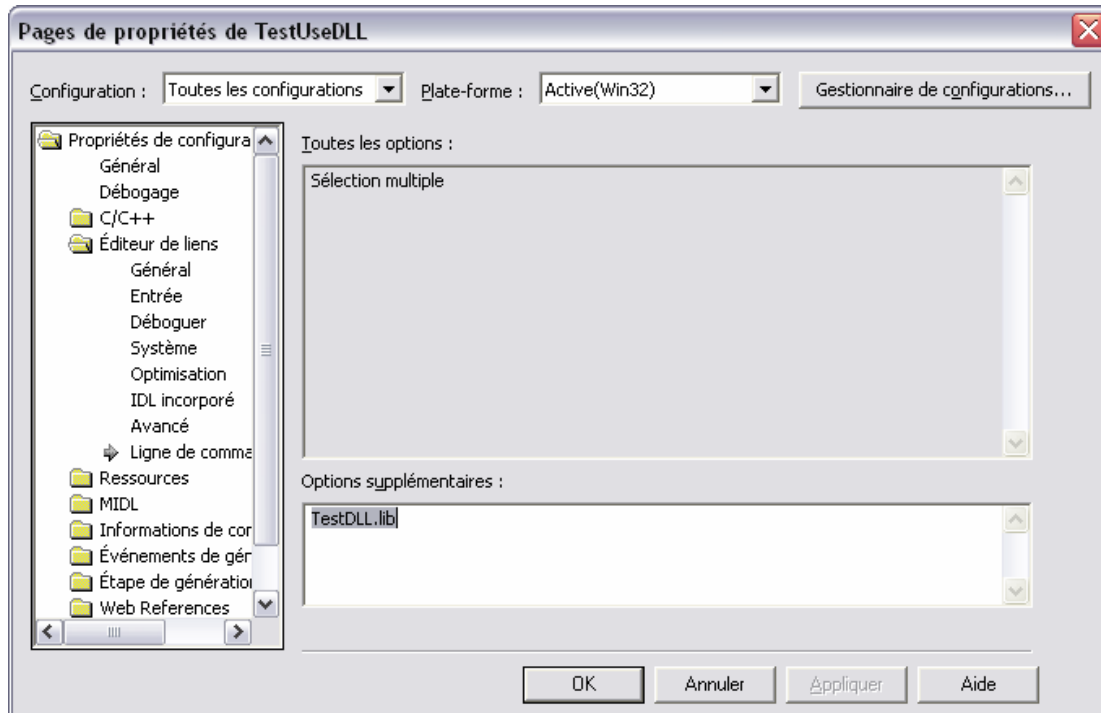
void main() {

    cMyClass objet;
    objet.setA(5);
    objet.add(10);
    std::cout << "a = " << objet.getA();

    int tmp;
    std::cin >> tmp;
}
```

Pour le faire fonctionner il vous faut :

- Copier le fichier .h (ici testDLL.h) de votre projet de dll dans celui de votre application
- Définir USE_DLL afin d'utiliser le .h en mode "utilisation"
- Copier le .dll et le .lib généré dans le projet de dll dans celui de votre application
- Linker la lib dans votre compilateur :



Faites le bien pour toutes les configurations ! (Sinon en Release sa va pas linker et vous vous demanderez pourquoi)

Conclusion

Facile, non ? Maintenant il ne vous reste plus qu'à l'utiliser dans vos projets !

Sources utilisées

MSDN, Flipcode et différents forums. (Merci google)