

Tutoriel de création et d'utilisation d'une bibliothèque dynamique (dll)

Cette procédure démontre comment on crée une bibliothèque dynamique (*Dynamic Link Library* ou *DLL*) contenant des classes utiles pour d'autres applications ou programmes.

L'utilisation de DLL s'avère importante pour la réutilisation du code, puisqu'elle permet la mise en commun efficace de bibliothèques de fonctions dans plusieurs applications.

En utilisant Visual C++ (Microsoft Visual Studio .Net 2005, version française), ce tutoriel couvre les sujets suivants:

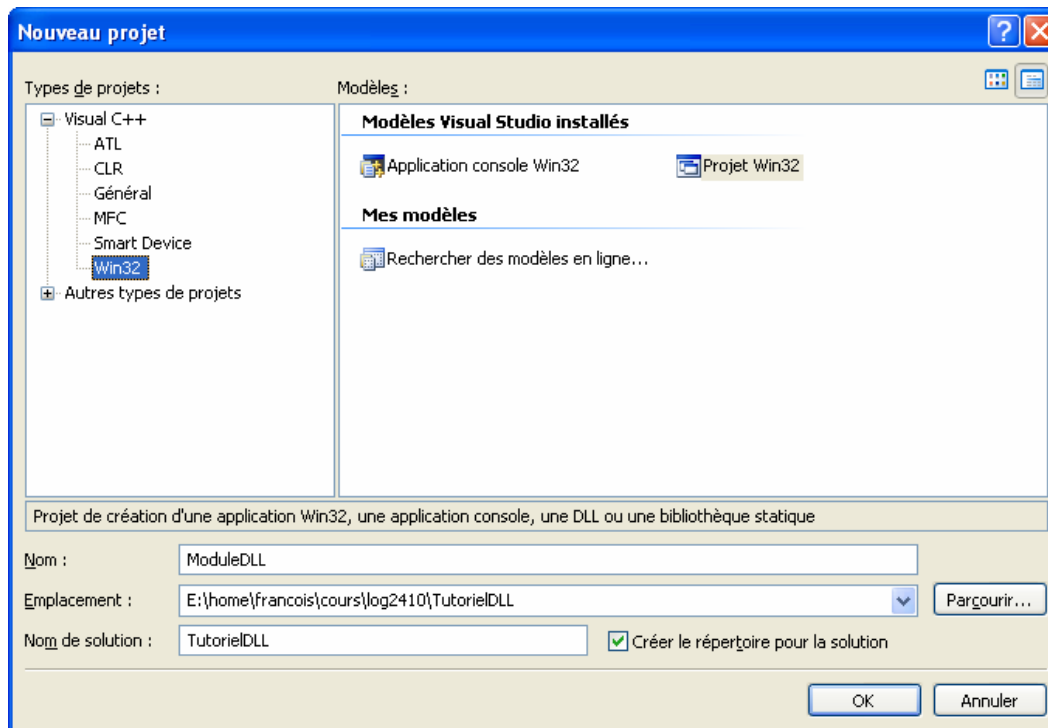
- Création d'une solution et d'un nouveau projet contenant une bibliothèque dynamique
- Ajout d'une classe (fichiers .h et .cpp) dans la bibliothèque.
- Création d'une application console (fichier .cpp) faisant référence à la bibliothèque dynamique.
- Utilisation des classes et fonctions de la bibliothèque par l'application console.
- Exécution de l'application console comme projet de démarrage.

Requis de l'application :

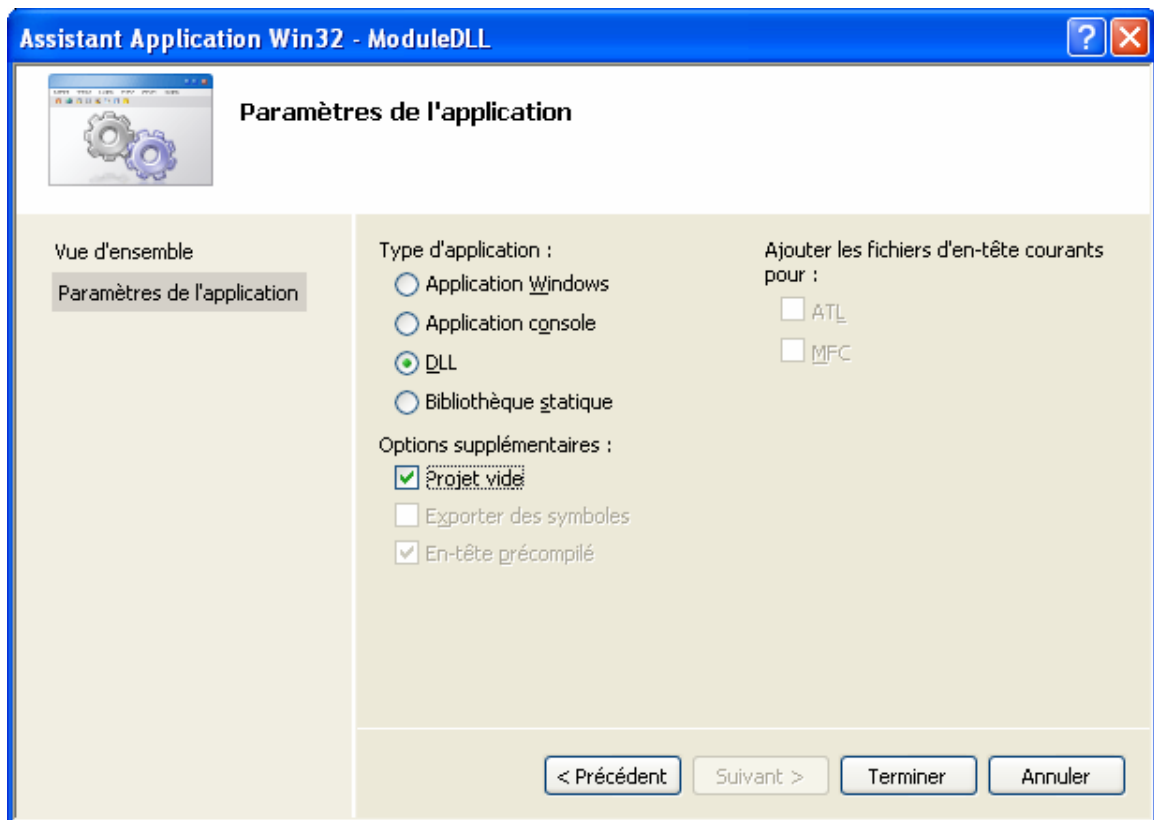
Notions de base du langage C++.

Création d'une solution et d'un nouveau projet de bibliothèque dynamique

1. Dans le menu **Fichier**, cliquez sur **Nouveau**, puis sur **Projet...**
2. A partir du volet **Types de projets**, sous le nœud **Visual C++**, sélectionner **Win32**.
3. Dans le volet **Modèles**, sélectionner **Projet Win32**.
4. Choisissez un nom de projet (**ModuleDLL**) et entrez-le dans le champ **Nom**. Choisissez le répertoire de votre projet dans le champ **Emplacement** en utilisant **Parcourir...**. Choisissez un nom pour la solution en remplissant le champ **Nom de solution** (**TutorielDLL**).



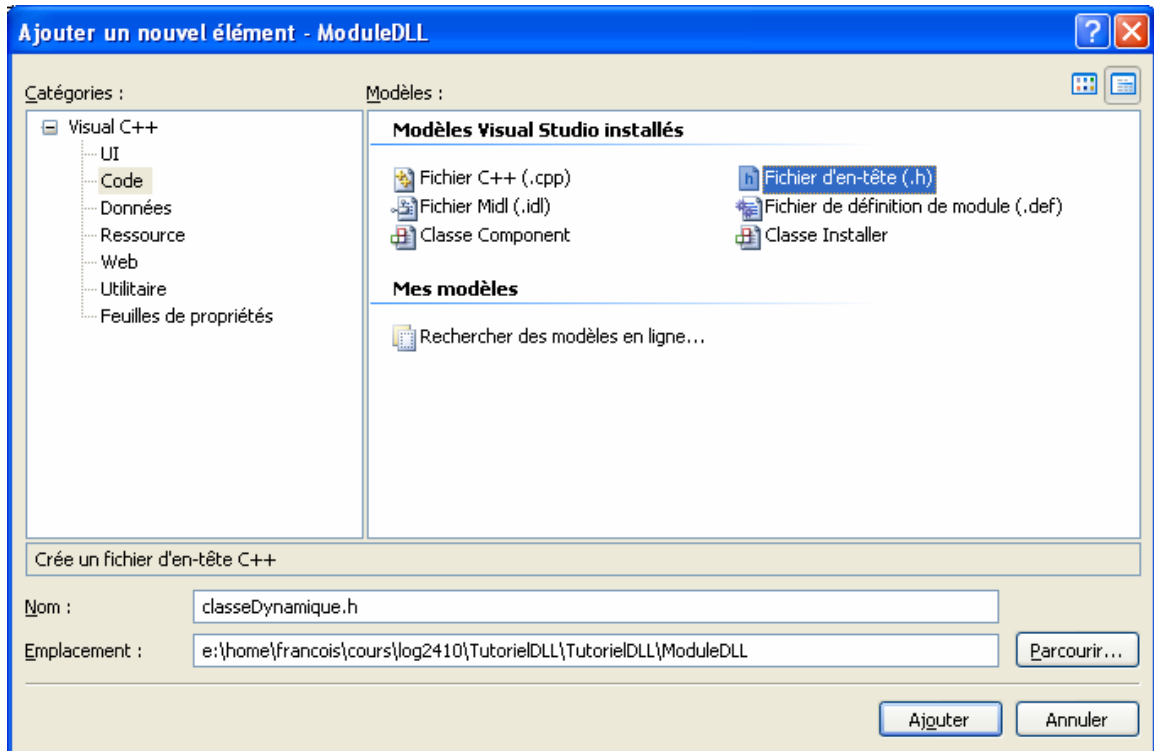
5. Cliquez sur **OK** afin de démarrer l'**Assistant Application Win32**. Cliquez sur **Suivant** pour aller dans **Paramètres de l'application**. Sous **Type d'application**, sélectionner **DLL**. Sous le volet **Options supplémentaires**, sélectionnez **Projet vide**.



6. Cliquez sur **Terminer** pour créer le projet.

Ajout d'une classe dans la bibliothèque dynamique :

1. Afin de créer un fichier d'en-tête pour la nouvelle classe, dans le menu **Projet**, sélectionner **Ajouter un nouvel élément...**. Une boîte de dialogue s'affiche. Sous le volet **Catégories**, sélectionner **Code** sous le nœud **Visual C++**. Dans le volet **Modèles Visual Studio installés** Cliquez sur **Fichier d'en-tête (.h)**. Choisissez un nom pour le fichier d'en-tête (**ClasseDynamique.h**) dans le champ **Nom** puis cliquez sur **Ajouter**. Un fichier vide sera affiché



2. Ajouter une classe simple **ClasseDynamique.h** :

```

// ClasseDynamique.h

#ifndef __HEADER__CLASSE_DYNAMIQUE__
#define __HEADER__CLASSE_DYNAMIQUE__

#include <string>
#include <iostream>

class ClasseDynamique
{
public:
    __declspec(dllexport) ClasseDynamique(std::string);
    __declspec(dllexport) ~ClasseDynamique( void );

    __declspec(dllexport) size_t getLength( void ) const;
    friend __declspec(dllexport) std::ostream& operator <<(
std::ostream&, const ClasseDynamique& );

private:
    std::string chaine_;
};

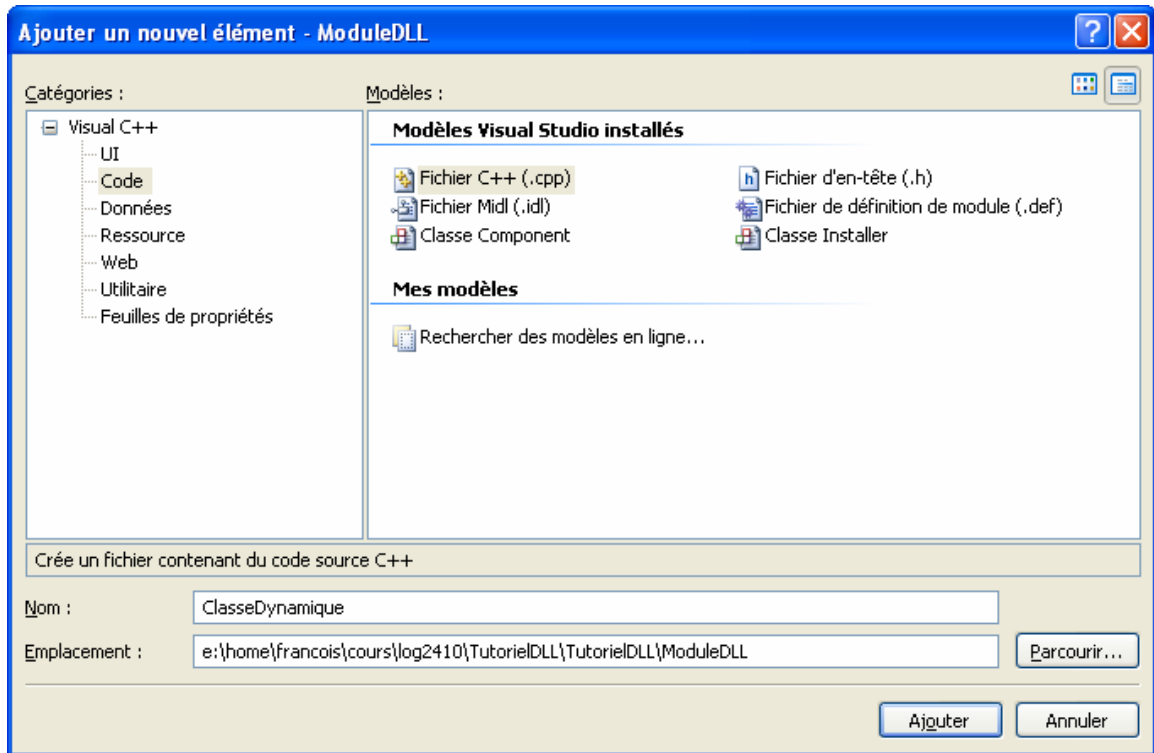
#endif

```

3. Dans les déclarations des méthodes, constructeur et destructeur de la classe **ClasseDynamique**, on remarque l'ajout obligatoire du modificateur **__declspec(dllexport)** pour permettre à la DLL

d'exporter ces méthodes afin qu'elles puissent être utilisées par d'autres programmes ou applications.

4. Afin de créer un fichier source pour la nouvelle classe, dans le menu **Projet**, sélectionner **Ajouter un nouvel élément...**. Une boîte de dialogue s'affiche. Sous le volet **Catégories**, sélectionner **Code** sous le nœud **Visual C++**. Dans le volet **Modèles Visual Studio installés** Cliquez sur **Fichier C++ (.cpp)**. Choisissez un nom pour le fichier d'en-tête (**ClasseDynamique.cpp**) dans le champ **Nom** puis cliquez sur **Ajouter**. Un fichier vide sera affiché



5. Implémenter les fonctionnalités de la classe **ClasseDynamique** dans le fichier source. Le code doit être comme suit :

```

#include "ClasseDynamique.h"

ClasseDynamique::ClasseDynamique( std::string chaine )
    : chaine_( chaine )
{
}

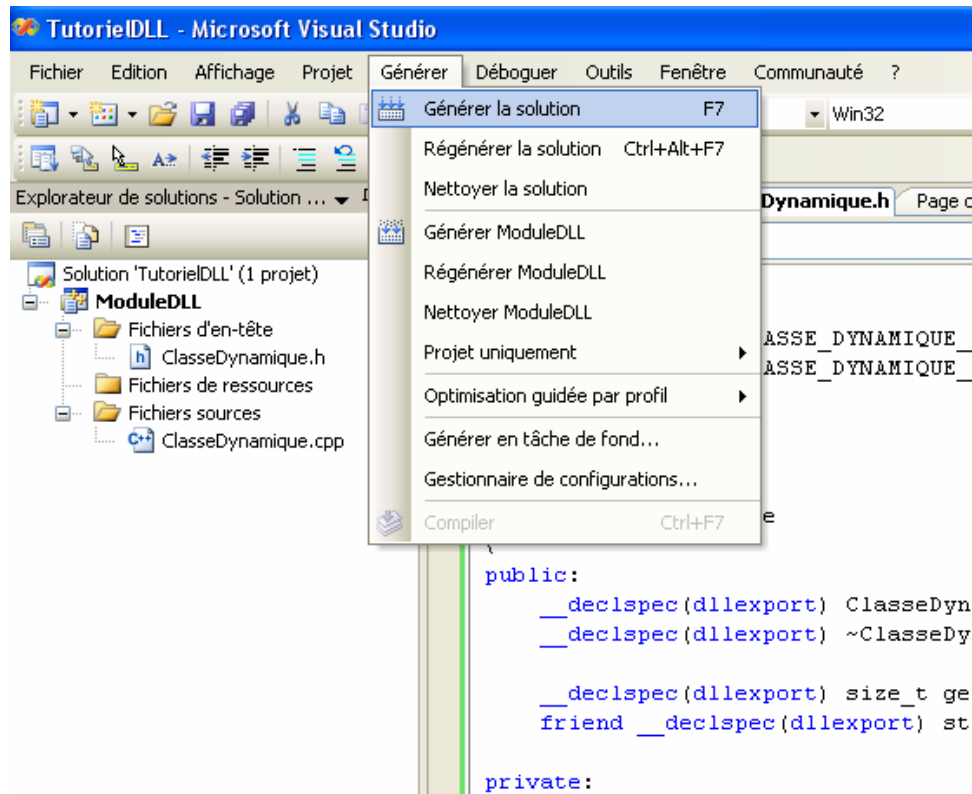
ClasseDynamique::~ClasseDynamique( void )
{
}

size_t ClasseDynamique::getLength( void ) const
{
    return chaine_.size();
}

std::ostream& operator<<( std::ostream& o, const ClasseDynamique&
classeDynamique )
{
    return o << classeDynamique.chaine_ << "( "
        << classeDynamique.getLength() << " )" << std::endl;
}

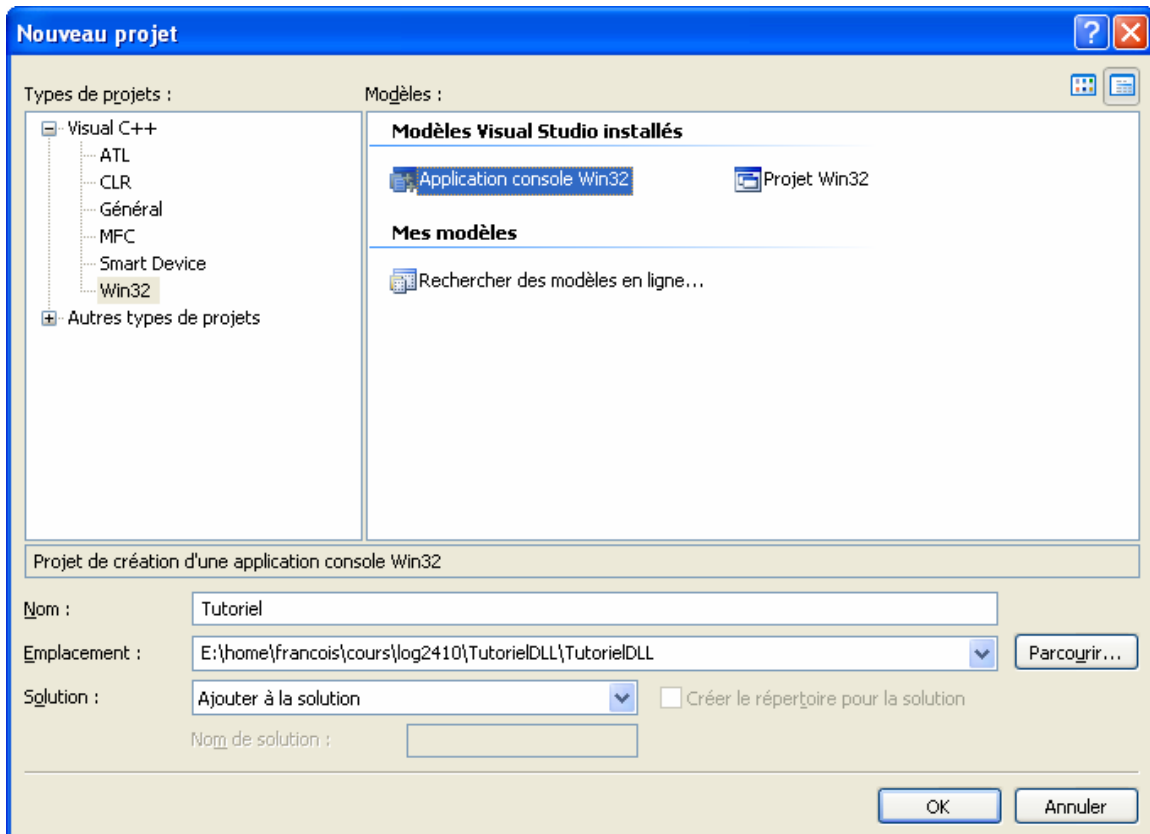
```

6. Dans le menu **Générer** cliquez sur **Générer la solution** afin de compiler la bibliothèque de liens dynamiques. Cela va créer une DLL qui peut être utilisé par d'autres applications ou programmes.

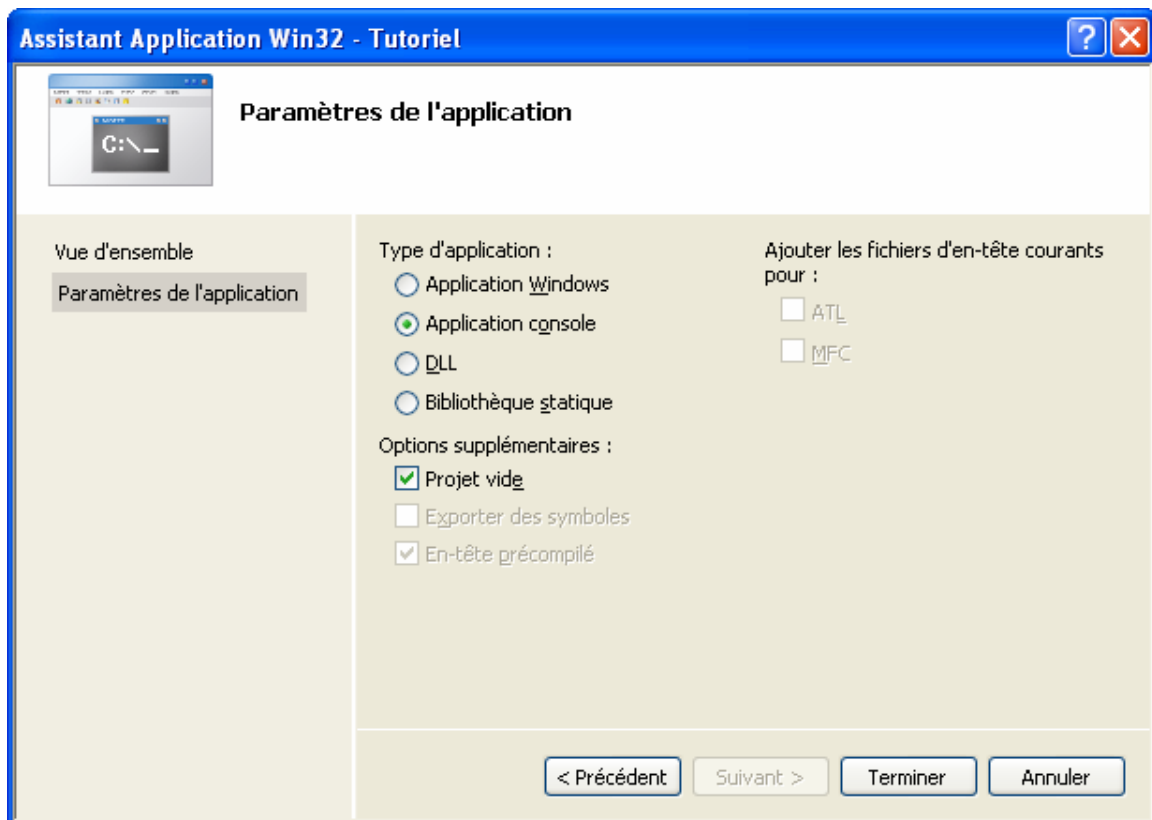


Création d'une application console en faisant la référence à la bibliothèque dynamique

1. Afin de créer une application principale qui va utiliser les fonctionnalités de la bibliothèque, sélectionner **Nouveau** dans le menu **Fichier**, puis cliquez sur **Projet...**
2. A partir du volet **Types de projets**, sous le nœud **Visual C++**, sélectionner **Win32**.
3. Dans le volet **Modèles**, sélectionner **Application console Win32**.
4. Choisissez un nom de projet (**Tutoriel**) et entrez-le dans le champ **Nom**. Choisissez le répertoire de votre projet dans le champ **Emplacement** (ce répertoire peut être le même que pour le projet de la bibliothèque) en utilisant **Parcourir...** Choisissez d'ajouter le nouveau projet dans la solution : **Ajouter à la solution**.



5. Cliquez sur **OK** afin de démarrer l'**Assistant Application Win32**. Cliquez sur **Suivant** pour aller dans **Paramètres de l'application**. Sous **Type d'application**, sélectionner **Application console**. Sous le volet **Options supplémentaires**, sélectionnez **Projet vide**.



6. Cliquez sur **Terminer** pour créer le projet.

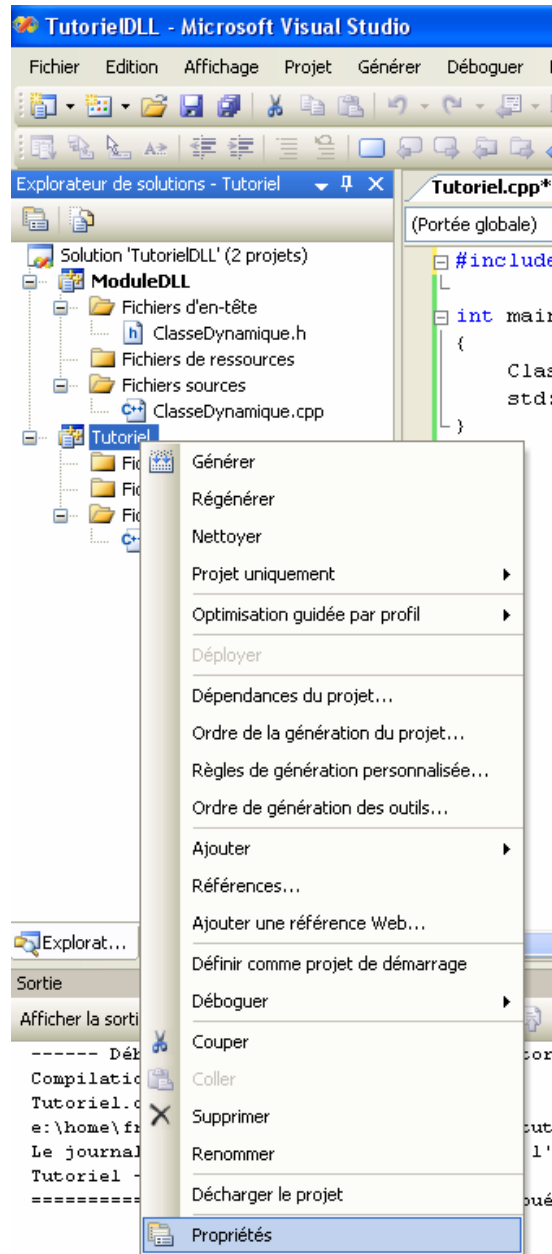
Utilisation des fonctionnalités de la bibliothèque par l'application console.

1. On vient de créer un projet vide **Tutoriel**. Créer un fichier source dont le nom sera identique au nom du projet qui vient d'être créé **Tutoriel.cpp**.
2. afin de créer un fichier source pour le nouveau projet, dans le menu **Projet**, sélectionner **Ajouter un nouvel élément...**. Une boîte de dialogue s'affiche. Sous le volet **Catégories**, sélectionner **Code** sous le nœud **Visual C++**. Dans le volet **Modèles Visual Studio installés** Cliquez sur **Fichier C++ (.cpp)**. Choisissez un nom pour le fichier (**Tutoriel.cpp**) dans le champ **Nom** puis cliquez sur **Ajouter**. Un fichier vide sera affiché
3. Implémenter les fonctionnalités du fichier source **Tutoriel.cpp**. Le code doit être comme suit :

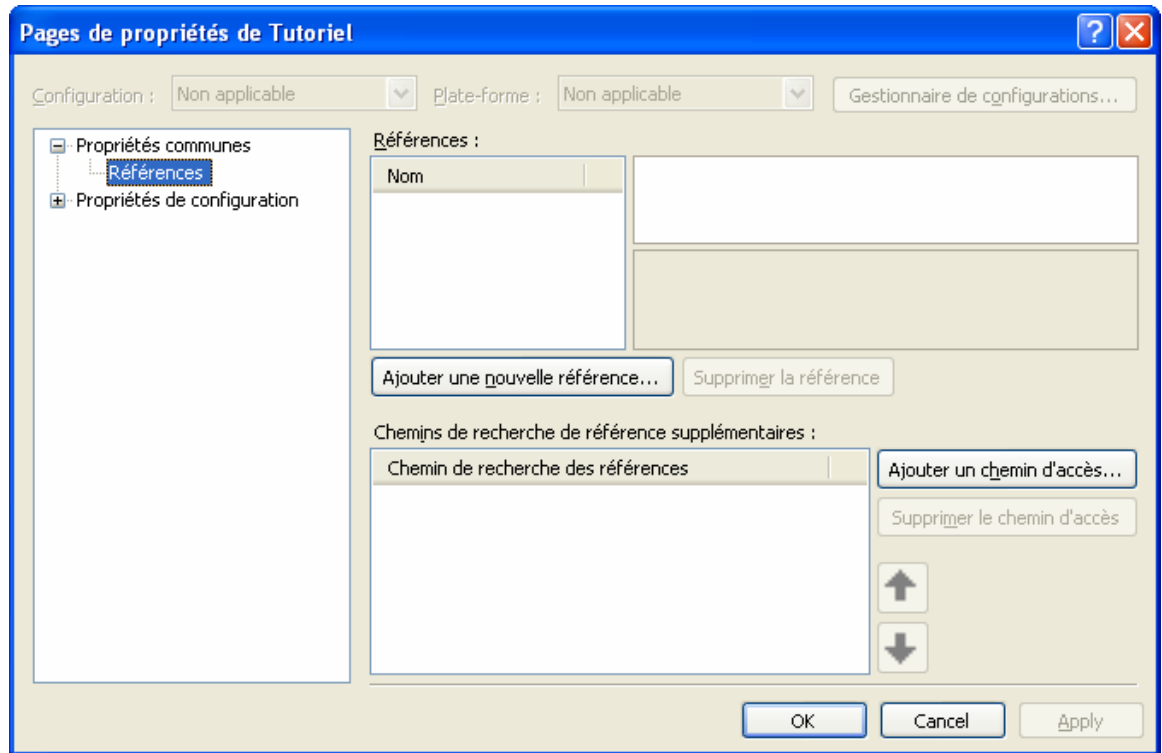
```
#include "ClasseDynamique.h"

int main( int argc, char** argv )
{
    ClasseDynamique classeDynamique( "Bonjour le monde !" );
    std::cout << classeDynamique << endl;
}
```

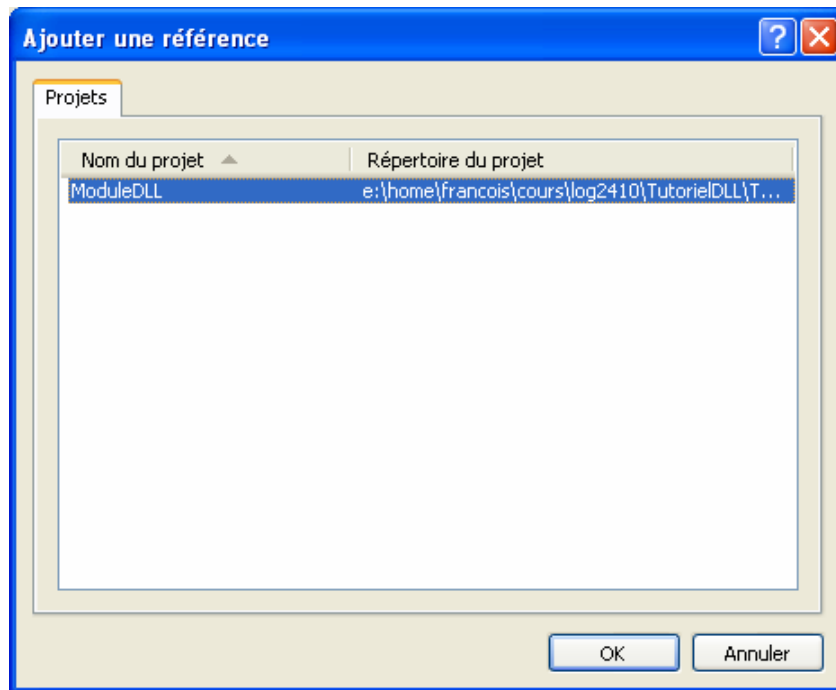
4. Afin d'utiliser les méthodes de la classe créée dans la bibliothèque dynamique, on devra référencer celle-ci. Sélectionner **Propriétés** dans le menu contextuel associé au projet **Tutoriel**.



Dans la boîte de dialogue **Page de propriétés de Tutoriel**, sélectionner **Références** sous le nœud **Propriétés communes**, par la suite cliquez sur le bouton **Ajouter une nouvelle référence**.

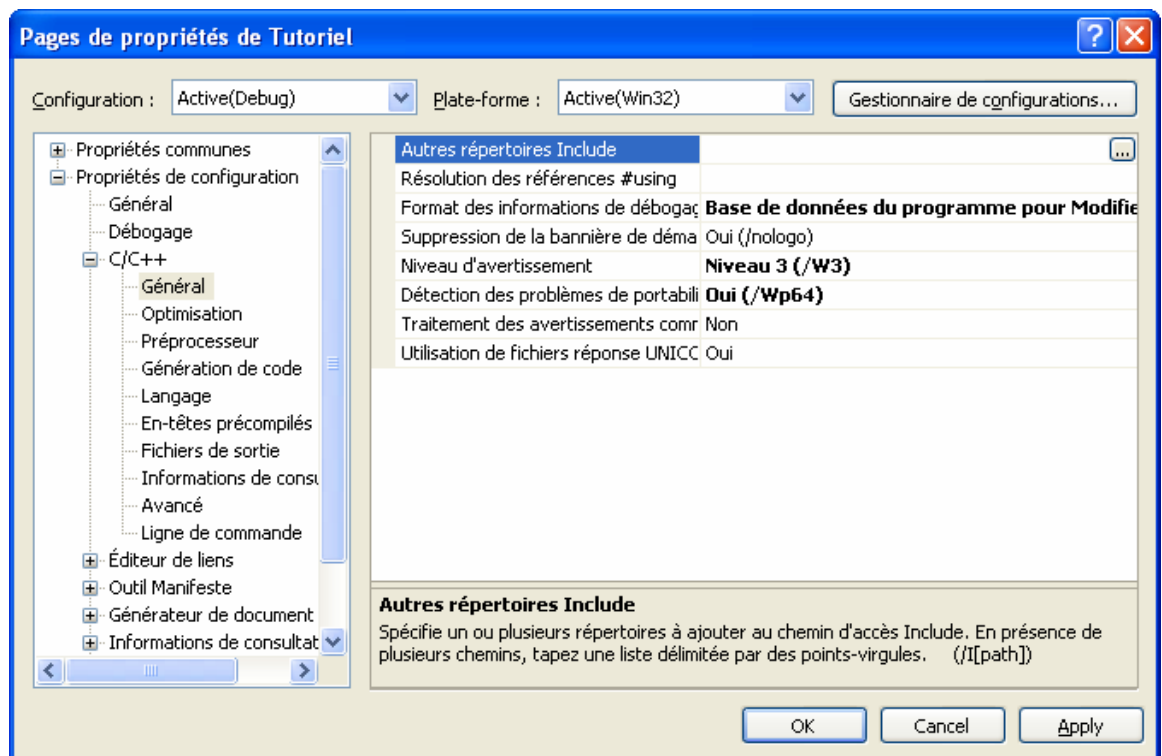


5. La boîte de dialogue **Ajouter une référence** s'affiche. Cette boîte contient toutes les bibliothèques qu'on peut référencer. Dans l'onglet **Projets** sélectionner **ModuleDLL** sous le champ **Nom du projet** puis cliquez sur **OK**.



6. Pour référencer les fichiers d'en-tête (.h) de la bibliothèque de liens dynamiques, on devra modifier le chemin d'accès des répertoires inclus. Sous le nœud **Propriétés de configuration** de la boîte de dialogue **Pages de propriétés de Tutoriel**, développez le nœud **C/C++** puis

sélectionner **Général**. Dans le champ **Autres répertoires Include** tapez le chemin d'accès vers l'emplacement du fichier d'en-tête **ClasseDynamique.h** (**E:\...\cours\log2410\TutorielDLL\TutorielDLL\ModuleDLL**), puis cliquez sur **OK**.



7. Générer le fichier exécutable en sélectionnant **Générer la solution** dans le menu **Générer**.

Exécution de l'application console comme projet de démarrage

1. Sélectionner le projet **Tutoriel** puis sous le menu **Projet** sélectionner **Définir comme projet de démarrage**.
2. Afin d'exécuter le projet, sélectionner **Exécuter sans débogage** sous le menu **Déboguer**. Le résultat doit être comme suit :

