



Documentation pour le test de projets C++ avec  
Visual Studio 2010 ultimate  
Quatrième année - Informatique

2010-2011



---

# Table des matières

---

1	Introduction . . . . .	4
2	Création d'un nouveau projet de test . . . . .	5
3	Création d'un nouveau test . . . . .	6
4	Exécution d'un test avec une instruction de la bibliothèque assert.h . . . . .	8
5	Écriture d'un test pour un projet C++ . . . . .	11
5.1	Propriétés du projet Visual C++ (Project1) . . . . .	11
5.2	Propriétés du projet de test (TestProject1) . . . . .	14
5.3	Importations de headers dans le projet de test (TestProject1) . . . . .	17

# 1 Introduction

Visual Studio vous offre la possibilité de créer un projet de test vous permettant de tester votre application. Cette fonctionnalité est facile à mettre en place pour un projet écrit en C#. Lorsque l'on veut tester un projet écrit en C++, il faut faire quelques adaptations.

Dans la suite nous suposerons que le projet C++ que vous souhaitez tester s'appelle `Project1`, vous n'aurez qu'à adapter avec le nom de votre projet. Ci-dessous, se trouve un screenshot de la solution à l'état initial c'est-à-dire que la solution ne contient que le projet C++ (`Project1`).

Dans le répertoire `Test de Ressources Projet` du Moodle, se trouve l'exemple développé dans ce document (fichier `Project1.zip`).

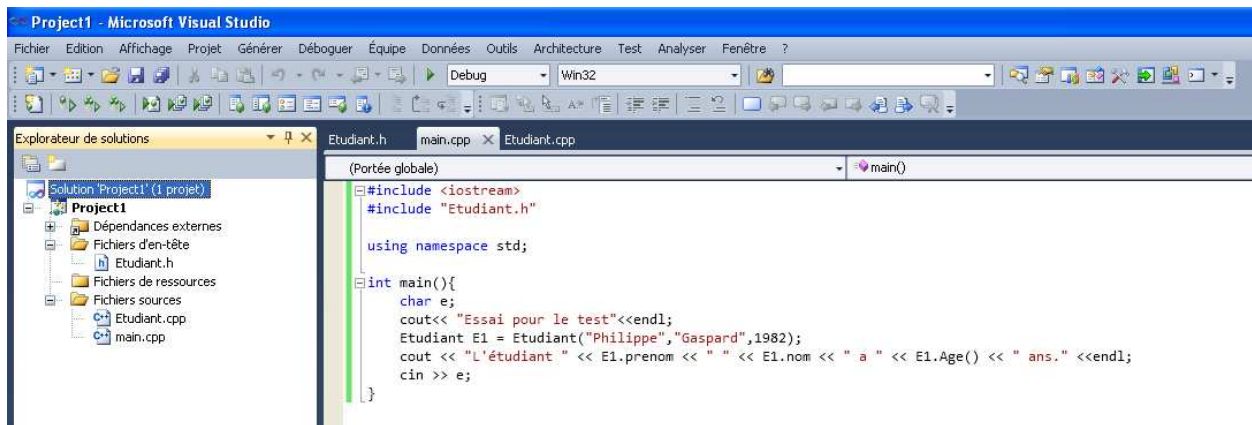


FIGURE 1 – Etat initial de votre projet sous Visual Studio.

## 2 Création d'un nouveau projet de test

Dans votre solution, créez un nouveau projet de test en faisant un clic droit sur la solution et en sélectionnant “Ajouter” et “Nouveau projet”.

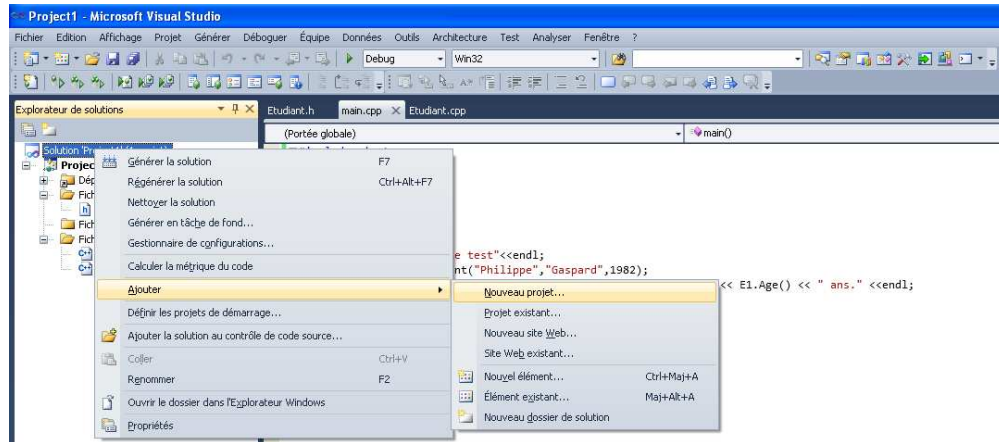


FIGURE 2 – Création d'un nouveau projet de test.

Ensuite choisir dans la boîte de dialogue un projet de test pour visual C++. Pour l'exemple, nous appellerons ce nouveau projet “TestProject1”.

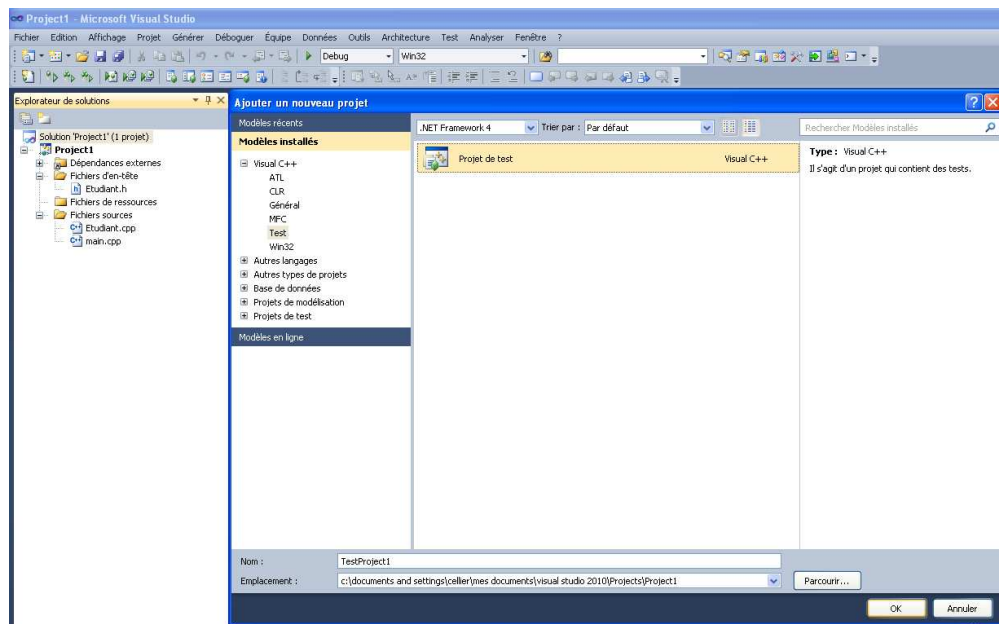


FIGURE 3 – Choix d'un projet de test.

À gauche dans l'explorateur de solutions le nouveau projet de test apparaît.

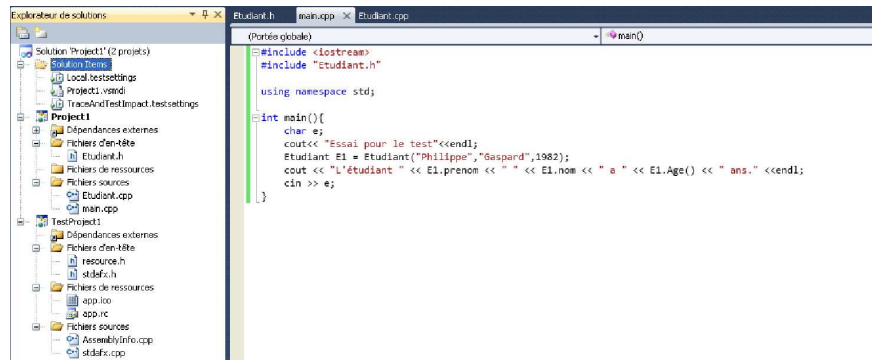


FIGURE 4 – Nouveau projet de test.

### 3 Création d'un nouveau test

Pour créer un nouveau test faire dans le menu contextuel du projet de test : Ajouter > Nouveau test.

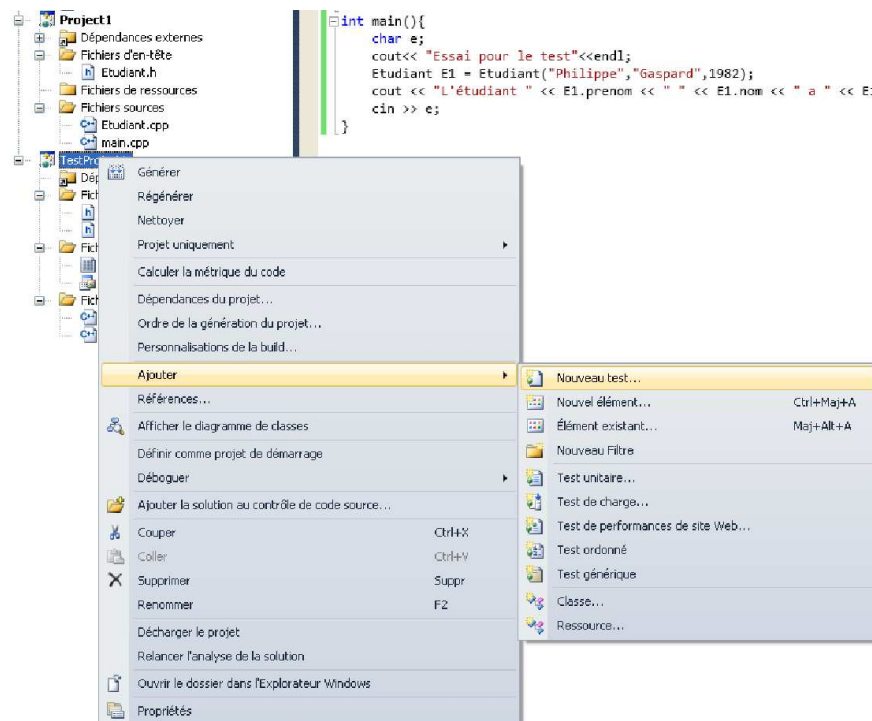


FIGURE 5 – Création d'un nouveau test.

Une fenêtre proposant différents types de tests apparaît. Vous pouvez choisir le type de test qui convient le mieux à ce que vous souhaitez tester. Pour l'exemple nous choisirons un test unitaire.



FIGURE 6 – Choix d'un test unitaire.

À gauche dans l'explorateur de solutions un nouveau fichier cpp (`UnitTest1.cpp`) apparaît, il s'agit du nouveau test que nous venons de créer.

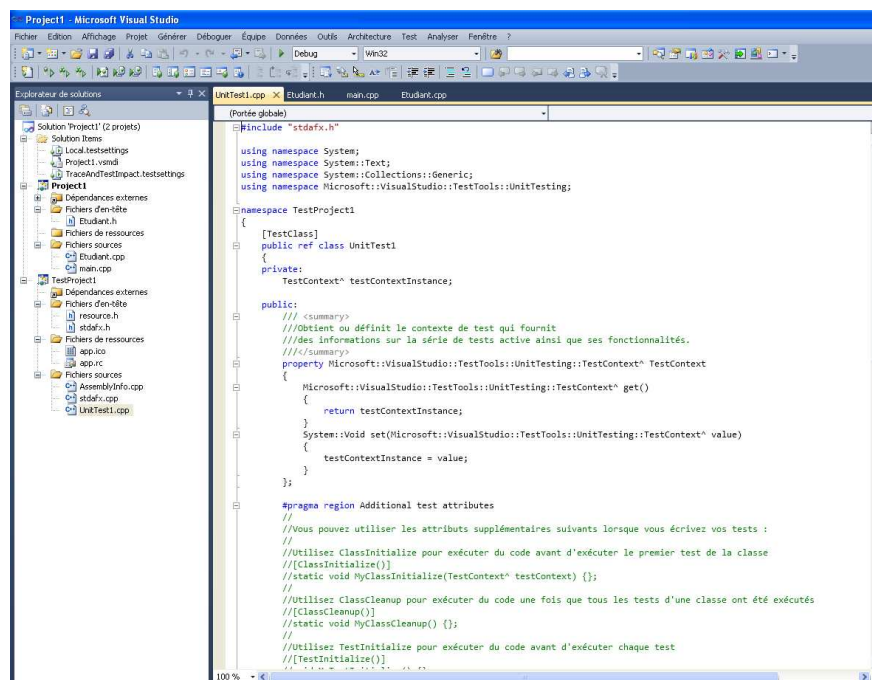


FIGURE 7 – Nouveau test unitaire.

## 4 Exécution d'un test avec une instruction de la bibliothèque assert.h

Il existe une bibliothèque `assert.h` permettant de vérifier si des prédicats sont vrais. Si tous les prédicats sont vérifiés, le test est en succès, sinon en échec. Pour pouvoir utiliser cette bibliothèque incluez là à votre fichier de test.

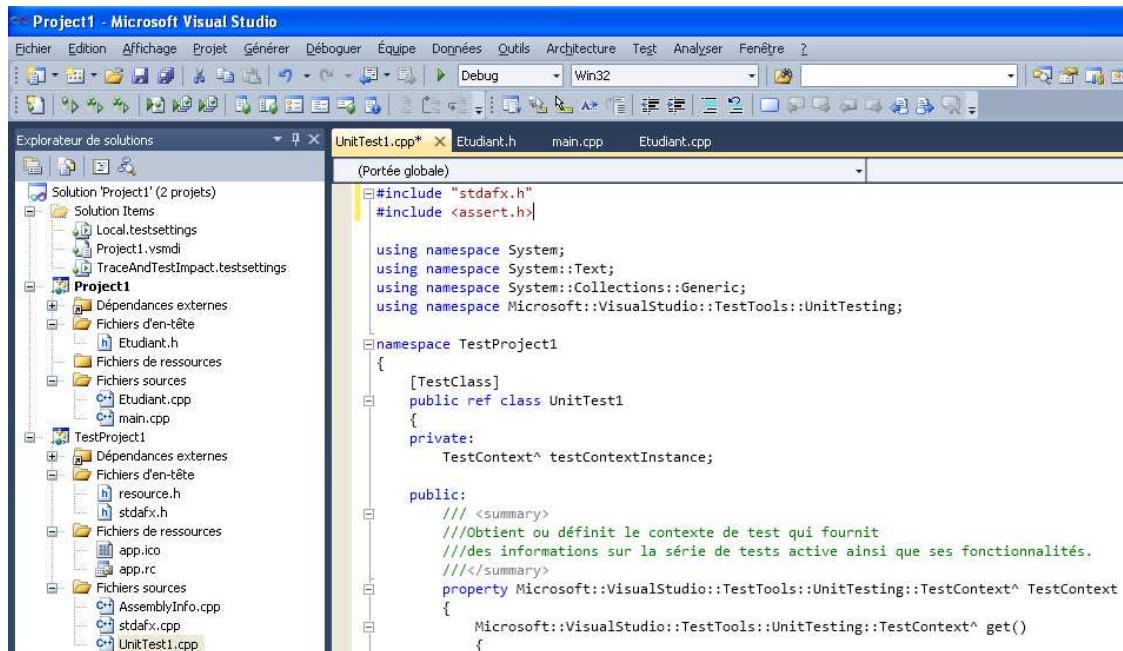


FIGURE 8 – Inclusion de `assert.h`.

Définissez ensuite les prédicats devant être vérifiés. Dans l'exemple on teste si 2 est égale à 3.

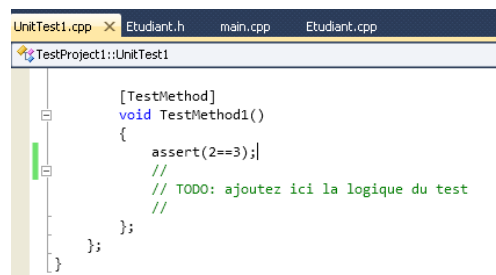


FIGURE 9 – Exemple d'utilisation de la bibliothèque `assert`.



Avant de pouvoir lancer le test, il faut paramétrer le projet. Ouvrez les propriétés du projet de test.

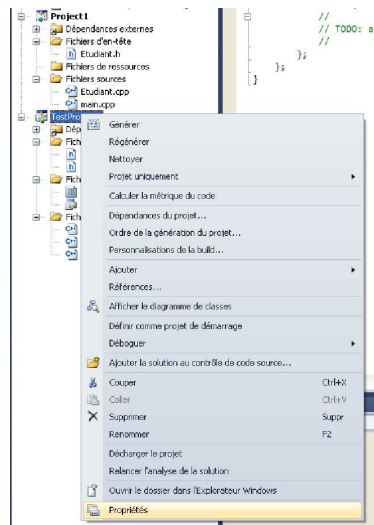


FIGURE 10 – Paramétrage du projet pour le C++ simple.

Puis dans Propriétés de configuration > C/C++ > Général, modifiez l'item Prise en charge du Common Language Runtime en l'affectant à Prise en charge du Common Language Runtime (/clr). Attention, il ne faut pas sélectionner /clr:pure ou les autres options. Cliquez sur OK.

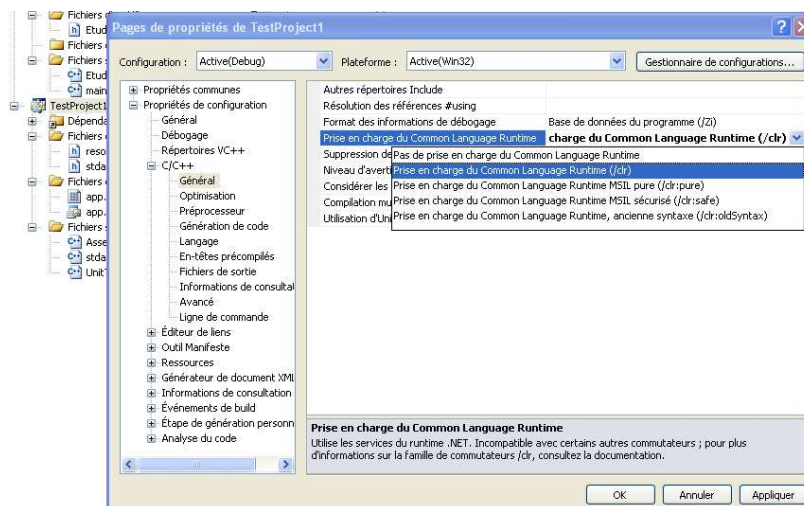


FIGURE 11 – Modification de l'option "Prise en charge du Common Language Runtime".

Vous pouvez maintenant lancer votre test. Cliquez droit sur le test à lancer et sélectionnez Exécuter les tests.

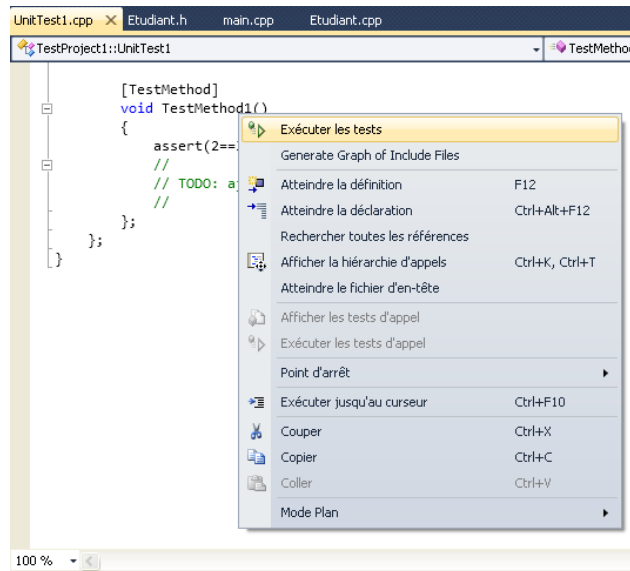


FIGURE 12 – Lancement d'un test.

Le test s'exécute et le résultat (à savoir si le test est réussi ou en échec) s'affiche dans la fenêtre des résultats. Dans l'exemple, le test a échoué.

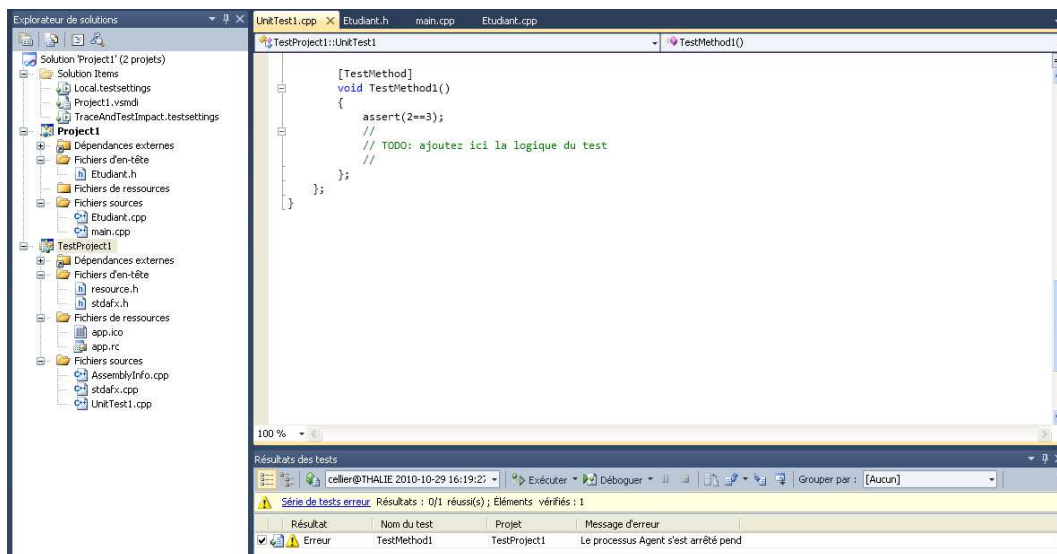


FIGURE 13 – Fenêtre de résultat des tests.

## 5 Écriture d'un test pour un projet C++

Pour pouvoir tester une méthode du projet C++ (Project1), il va falloir modifier plusieurs éléments de votre configuration aussi bien pour le projet C++ que pour le projet de test (TestProject1).

### 5.1 Propriétés du projet Visual C++ (Project1)

Tout d'abord, il faut transformer le projet C++ Project1 en bibliothèque statique. Ouvrez les propriétés de votre projet en C++ (Project1).

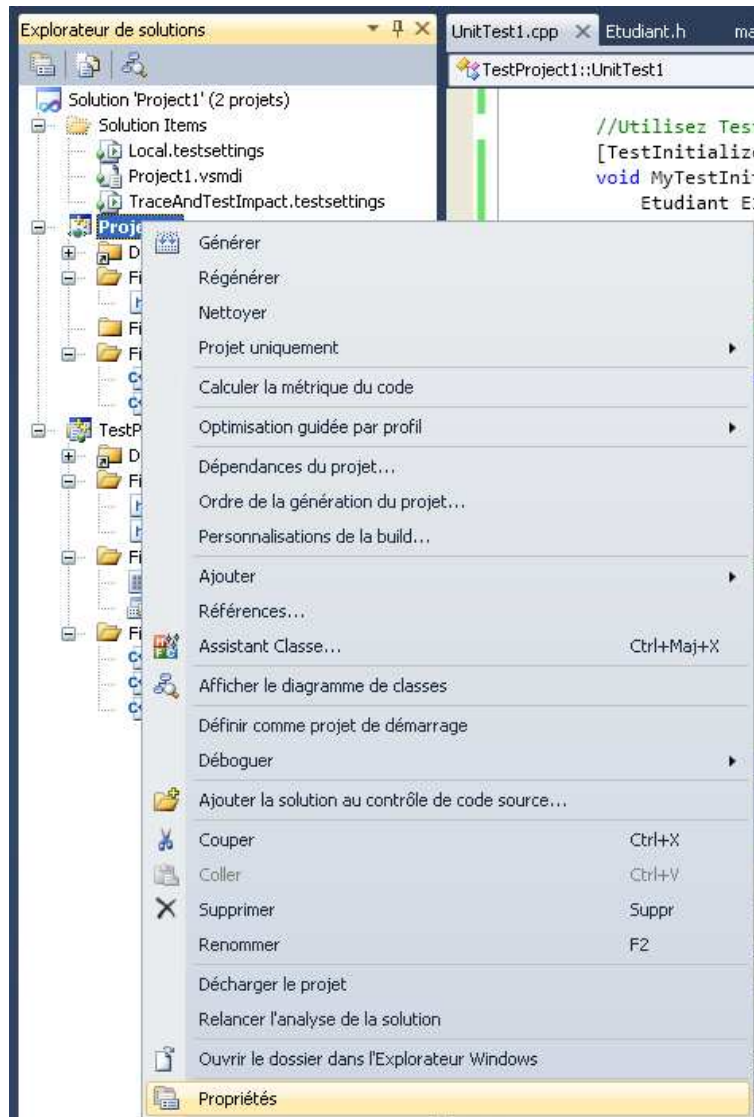


FIGURE 14 – Sélection des propriétés du projet Visual C++.

Modifiez dans Propriétés de configuration > Général > Paramètres par défaut, l'item Type de configuration et instanciez le à Bibliothèque statique (.lib).

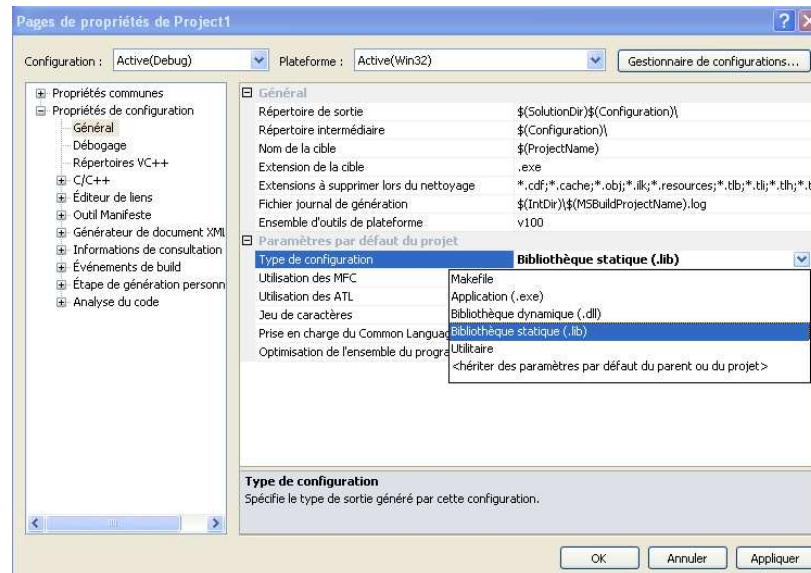


FIGURE 15 – Configuration pour une bibliothèque statique.

De plus, dans Propriétés de configuration > Général > Général, modifiez l'item Extension de la cible et instanciez le à .lib.

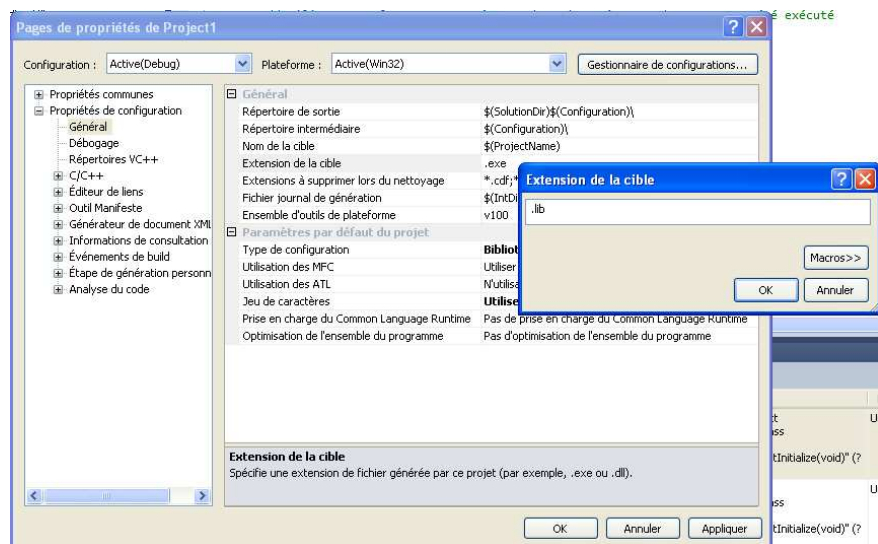


FIGURE 16 – Configuration de l'extension de la cible.

Ensuite il est important que vous régénériez votre projet visual C++ (Project1).

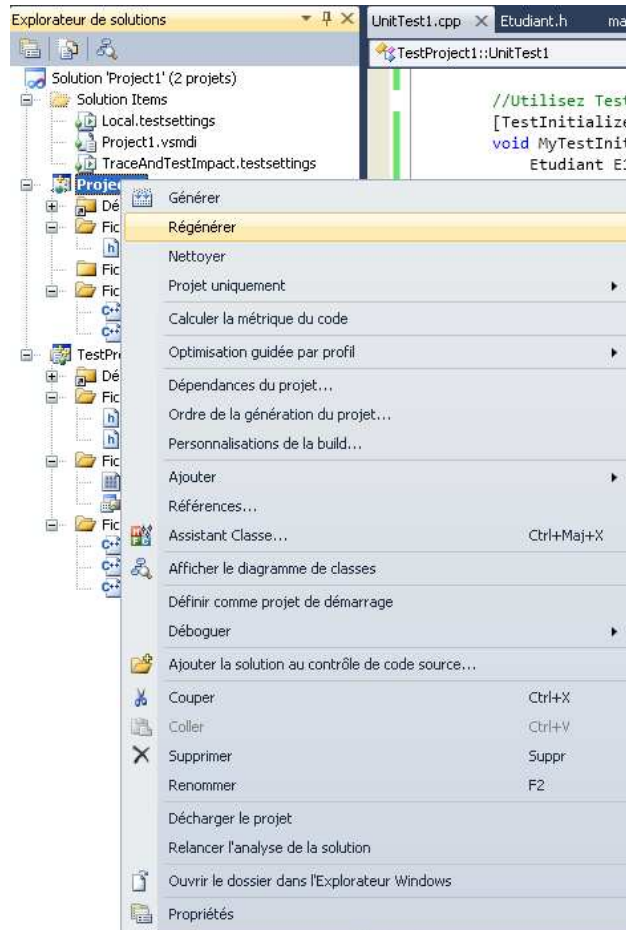


FIGURE 17 – Régénération du projet Visual C++.

Après cette régénération du projet, un nouveau fichier (Project1.lib) apparaît dans le répertoire Debug de la solution.

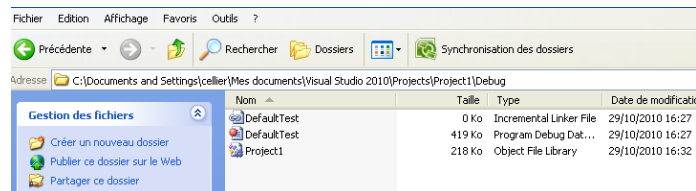


FIGURE 18 – Fichier .lib créé.

## 5.2 Propriétés du projet de test (TestProject1)

Maintenant que votre projet C++ génère bien un fichier `.lib`, il faut faire en sorte que votre projet de test puisse y accéder.

Tout d'abord créez une dépendance en le projet de test et le projet visual C++. Cliquez droit sur le projet de test et sélectionnez `Dépendances` du projet.

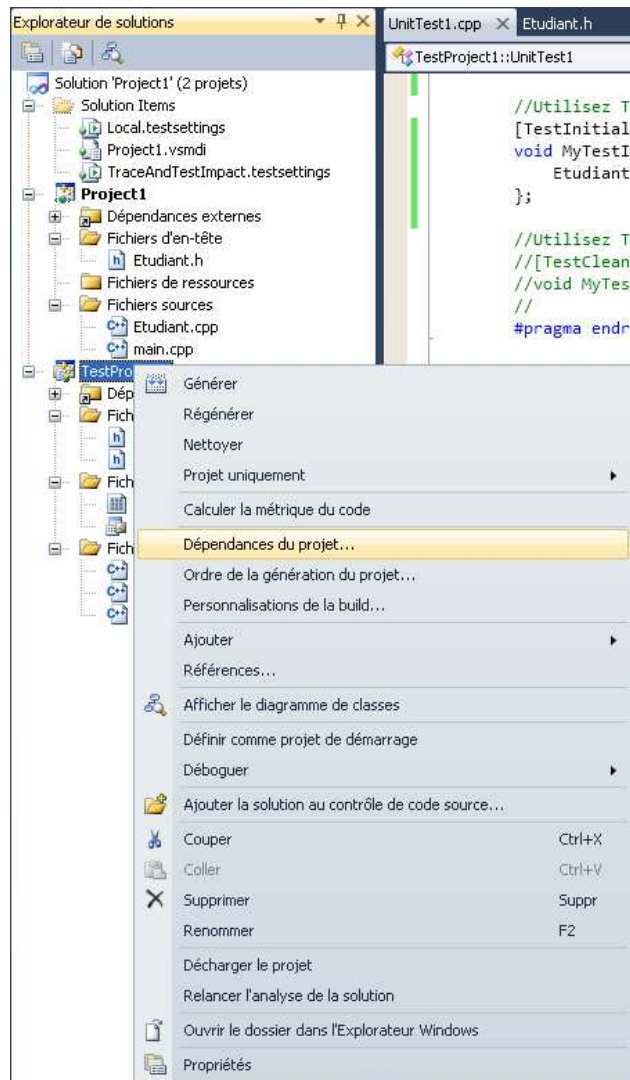


FIGURE 19 – Création d'une dépendance.

Ajoutez une dépendance vers le projet C++ (Project1), cliquez sur OK.

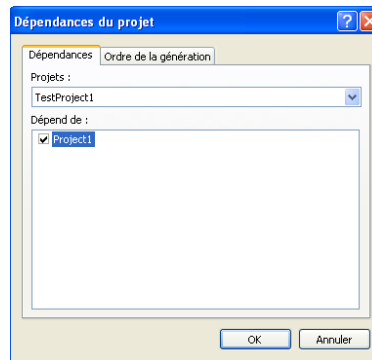


FIGURE 20 – Ajout d'une dépendance.

Cliquez droit sur le projet de test et sélectionnez Propriétés.

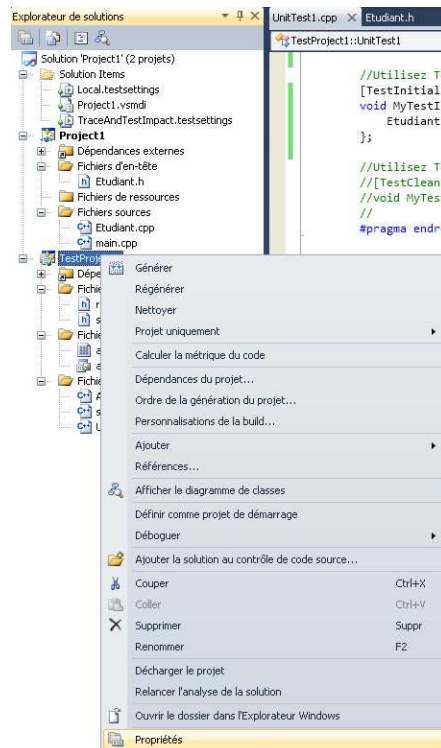


FIGURE 21 – Ouverture des propriétés du projet de test.



Dans Propriétés de configuration > Editeur de liens > Général, modifiez l'item Répertoires de bibliothèques supplémentaires en indiquant le répertoire où se trouve le .lib du projet Visual C++ généré précédemment.

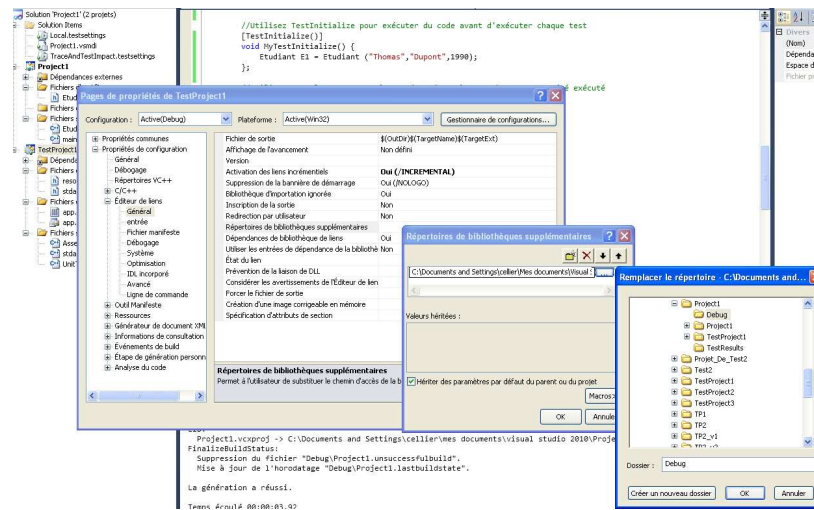


FIGURE 22 – Chemin de la bibliothèque.

Puis dans Propriétés de configuration > Editeur de liens > entrée, modifiez l'item Dépendances supplémentaires en précisant quelle bibliothèque statique vous souhaitez importer (Project1.lib).

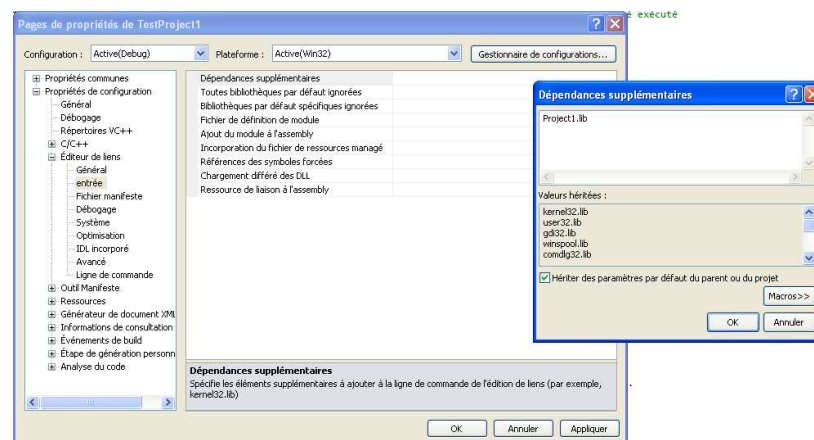
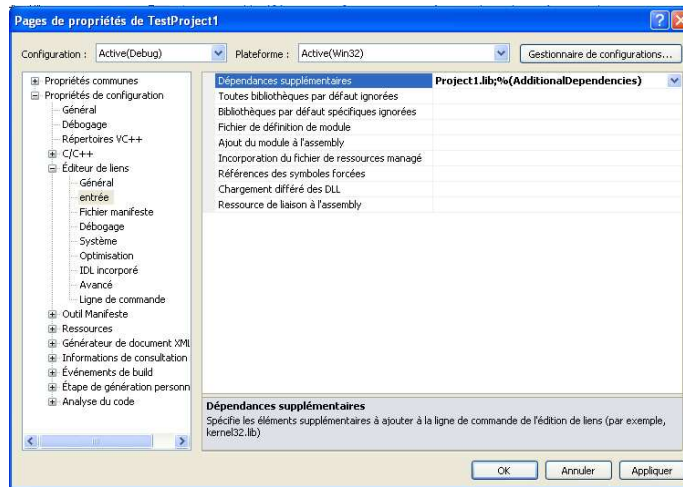


FIGURE 23 – Configuration de la bibliothèque choisie.



La bibliothèque apparait dans l'item Dépendances supplémentaires.



### 5.3 Importations de headers dans le projet de test (TestProject1)

Enfin, il faut indiquer où se trouve les fichiers que les tests du projet de test peuvent appeler.

Dans la fenêtre de Propriétés du projet de test, modifiez dans Propriétés de configuration > C/C++ > Général, l'item Autres répertoires Include et indiquez le chemin des header du projet Visual C++ (Project1).

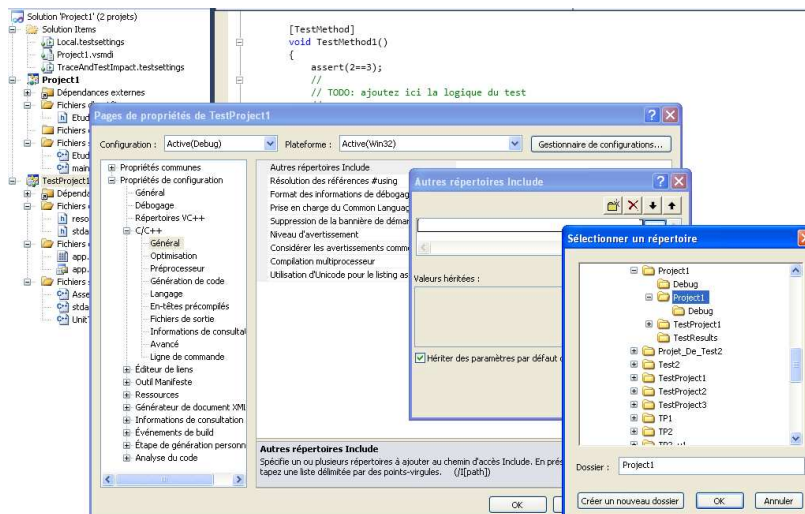
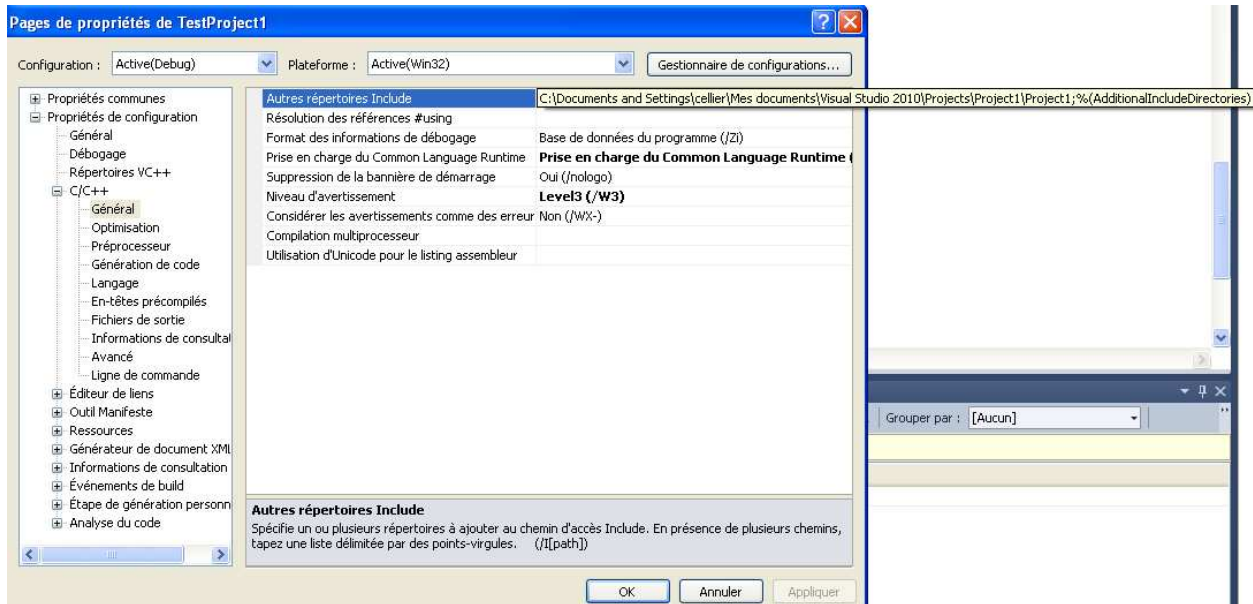


FIGURE 24 – Paramétrage du chemin d'accès aux includes.

Le bon chemin apparait dans la valeur de l'item Autres répertoires Include.



Ainsi on peut inclure dans le test de TestProject1, le header Etudiant.h du projet Visual C++ Project1.

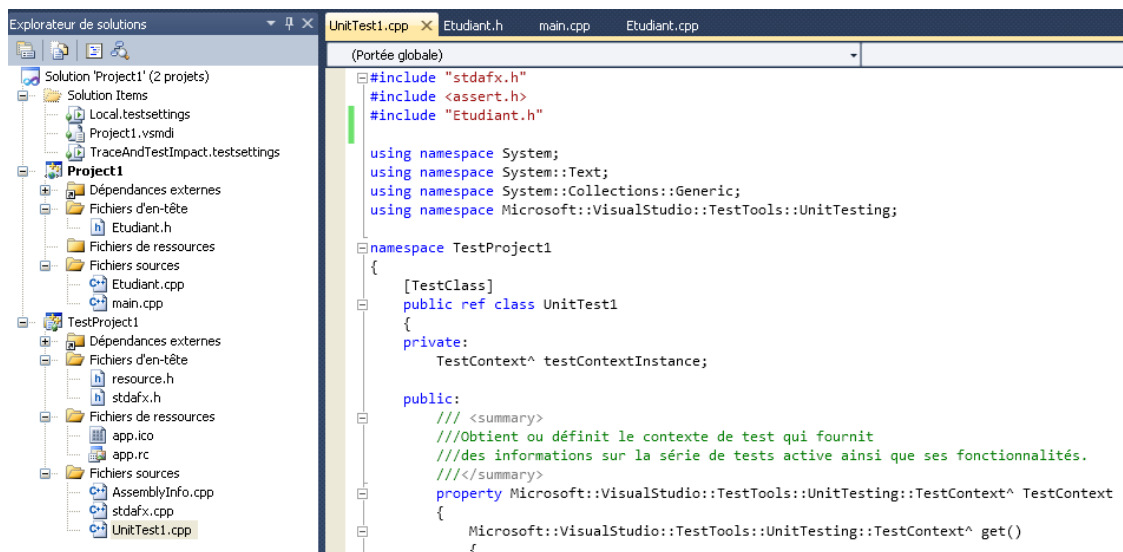


FIGURE 25 – Inclusion du header “Etudiant.h” dans le test.

On relance un nouveau test sur une méthode de la classe Etudiant du projet Project1 qui réussit.

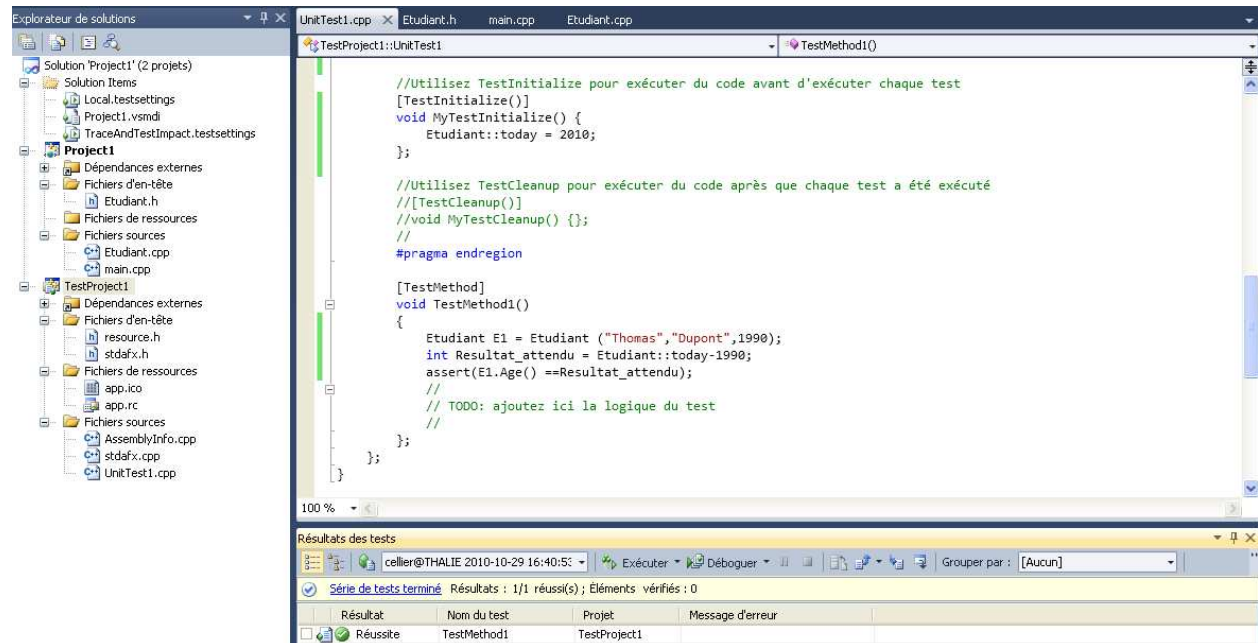


FIGURE 26 – Test qui réussit de la méthode Age de la classe Etudiant.