#ifndef \_\_Code\_\_ // if x.h hasn't been included yet...

#define \_\_Code\_\_ // #define this so the compiler knows it has been included

#include <string>

// mettre ici les differentes structure de donnees

typedef struct T\_Graphe

{

int i;

}T\_Graphe;

void lire\_fichier(std::string nom, T\_Graphe &Un\_Graphe);

#endif

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include "code.h"

void lire\_fichier(std::string nom,

T\_Graphe &Un\_Graphe)

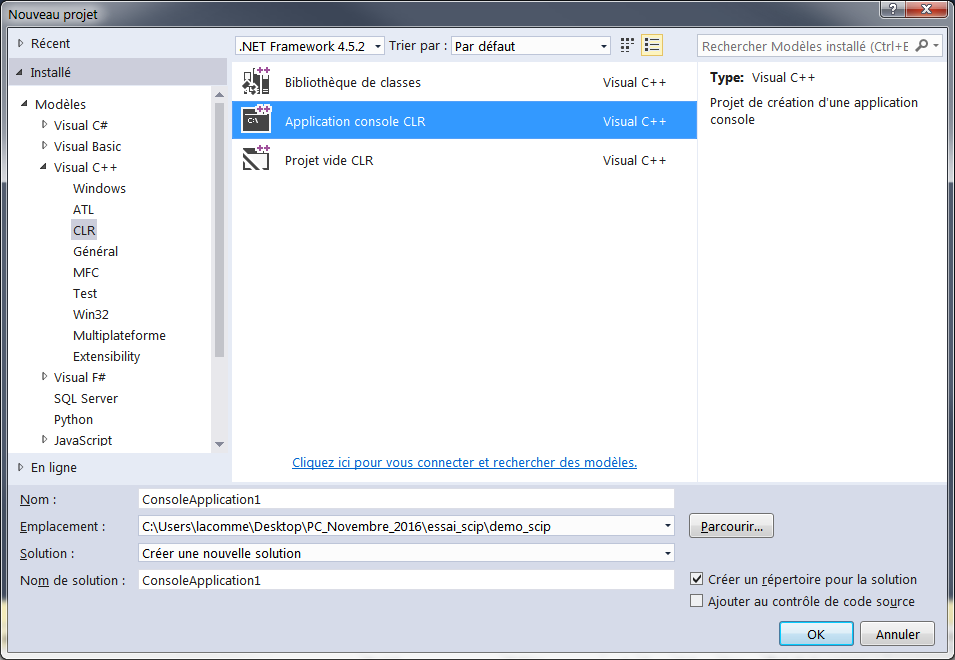
{

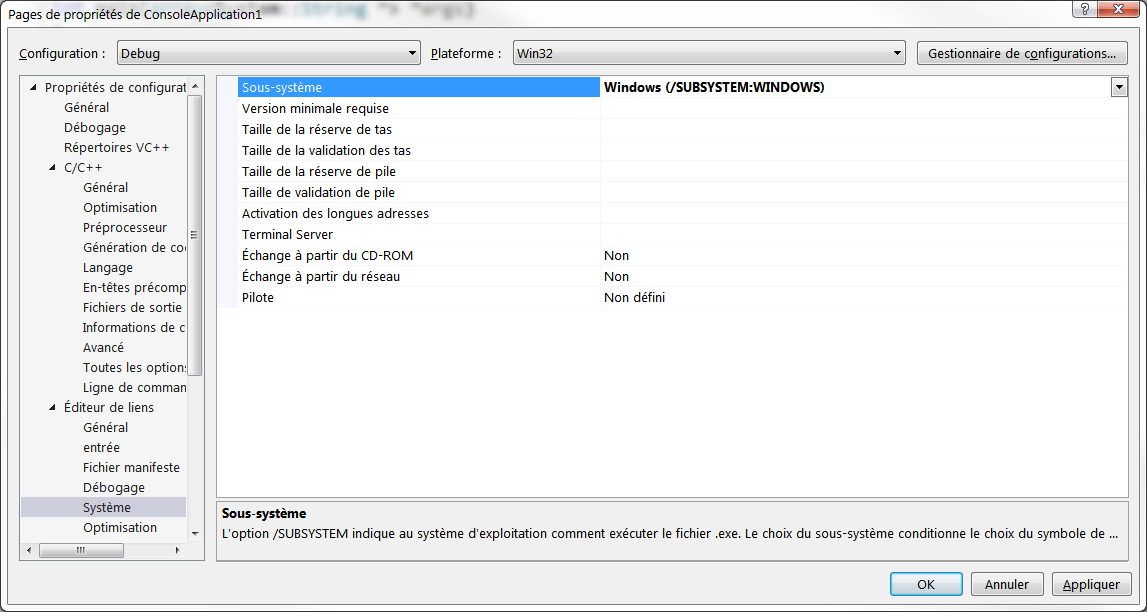
// on lit le fichier

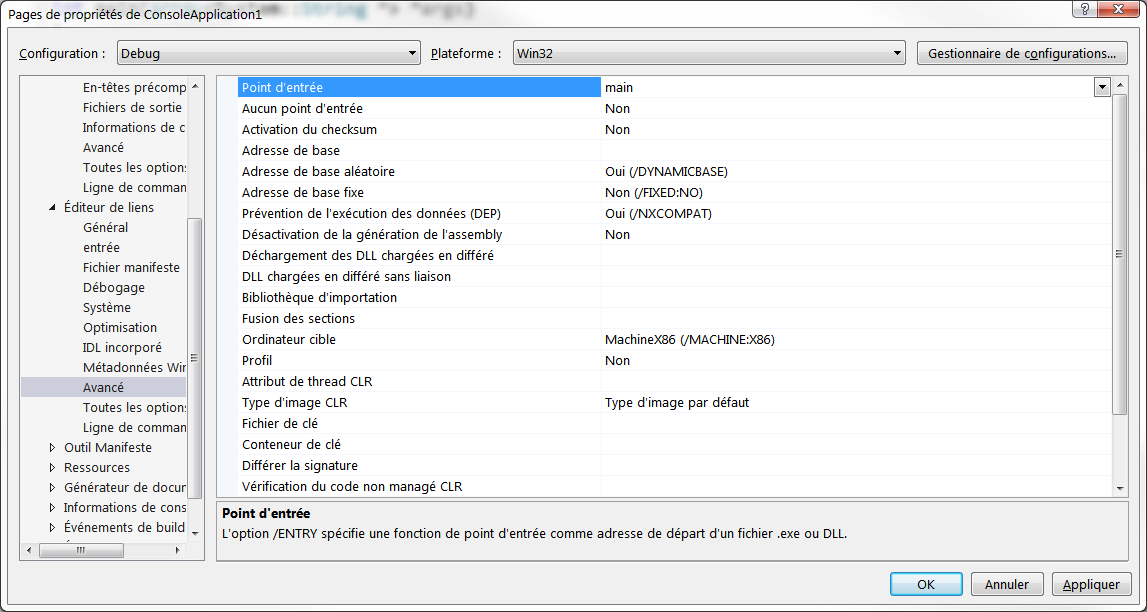
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

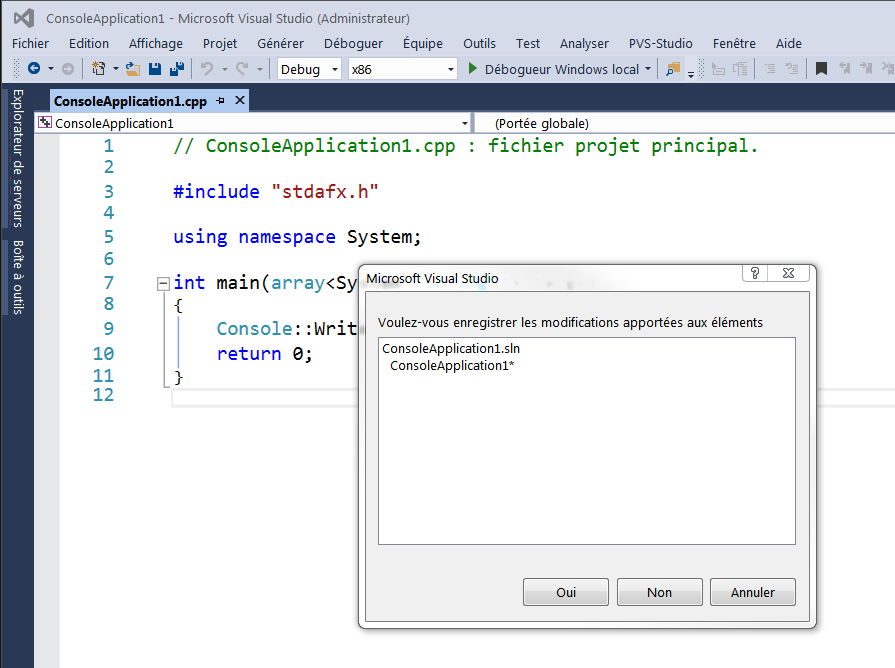
Faire une windows Form : Visual 2015







Fermer la solution et valider



Ouvrir le projet

#include "stdafx.h"

#include "form\_main.h"

using namespace System;

using namespace System::Windows::Forms;

int main(array<System::String ^> ^args)

{

    Console::WriteLine(L"Hello World");

Application::EnableVisualStyles();

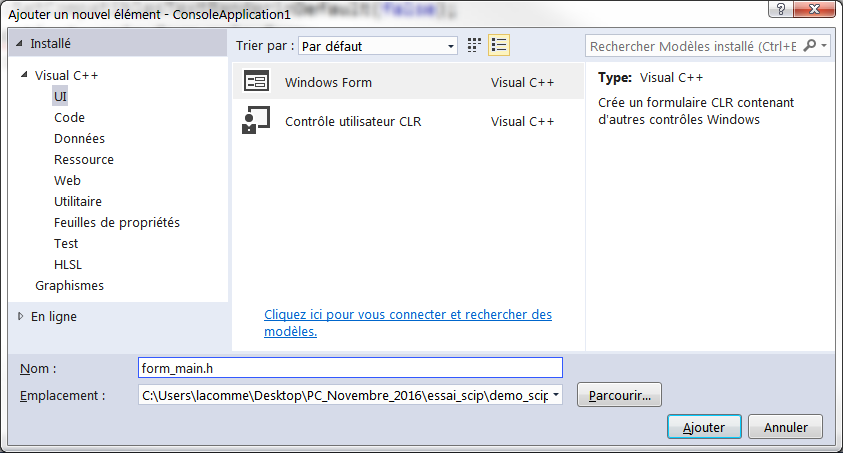
Application::SetCompatibleTextRenderingDefault(false);

ConsoleApplication1::form\_main main\_form;

Application::Run(%main\_form);

    return 0;

}



#include "stdafx.h"

#include "form\_main.h"

using namespace System;

using namespace System::Windows::Forms;

int main(array<System::String ^> ^args)

{

    Console::WriteLine(L"Hello World");

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

ConsoleApplication1::form\_main main\_form;

Application::Run(%main\_form);

    return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#pragma once

#include <msclr\marshal\_cppstd.h>

#include <sstream>

namespace ConsoleApplication1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace msclr::interop;

private: System::Void button2\_Click(System::Object^  sender, System::EventArgs^  e) {

System::String^ contenu = "essai";

msclr::interop::marshal\_context context;

std::string standardString = context.marshal\_as<std::string>(contenu);

System::String^ contenu\_zone = textBox1->Text;

richTextBox1->Text = contenu\_zone;

}

private: System::Void button3\_Click(System::Object^  sender, System::EventArgs^  e)

{

System::String^ s\_nb\_1 = textBox2->Text;

System::String^ s\_nb\_2 = textBox3->Text;

msclr::interop::marshal\_context context;

std::string chaine1 = context.marshal\_as<std::string>(s\_nb\_1);

std::string chaine2 = context.marshal\_as<std::string>(s\_nb\_2);

// convertir en integer

int nb1 = -1;

int nb2 = -1;

std::istringstream(chaine1) >> nb1;

std::istringstream(chaine2) >> nb2;

int nb\_res = nb1 + nb2;

// autre sol

std::stringstream  oss1;

oss1 << chaine1;

int nb11;

oss1 >> nb11;

// le res. en string

std::stringstream chaine\_res;

chaine\_res << nb\_res;

std::string chaine\_res\_string = chaine\_res.str();

System::String^ s\_nb\_res = marshal\_as<String^>(chaine\_res\_string);

textBox4->Text = s\_nb\_res;

}

System::String^ s\_nb\_1 = textBox2->Text;

System::String^ s\_nb\_2 = textBox3->Text;

msclr::interop::marshal\_context context;

std::string chaine1 = context.marshal\_as<std::string>(s\_nb\_1);

std::string chaine2 = context.marshal\_as<std::string>(s\_nb\_2);

// convertir en integer

int nb1 = -1;

int nb2 = -1;

std::istringstream(chaine1) >> nb1;

std::istringstream(chaine2) >> nb2;

int nb\_res = nb1 + nb2;

// autre sol

std::stringstream  oss1;

oss1 << chaine1;

int nb11;

oss1 >> nb11;

// le res. en string

std::stringstream chaine\_res;

chaine\_res << nb\_res;

std::string chaine\_res\_string = chaine\_res.str();

System::String^ s\_nb\_res = marshal\_as<String^>(chaine\_res\_string);

textBox4->Text = s\_nb\_res;