

Microsoft Visual C++ 2010 Express : Téléchargement, installation et configuration

par 3DArchi ([Ressources proposées par 3DArchi](#))

Date de publication : 19 mai 2010

Dernière mise à jour :

Ce tutoriel présente la procédure d'installation et de configuration de Visual C++ Express Edition 2010, l'environnement de développement C++ gratuit de Microsoft. Poursuivant la même politique de mise à disposition d'une version gratuite, la sortie de Visual Studio 2010 s'est accompagnée des versions Express : Visual C++ 2010, Visual Basic 2010, Visual C# 2010, Visual Web Developer 2010 et Visual Studio 2010 Express for Windows Phone CTP. Nous abordons ici l'installation de Visual C++ Express 2010. Visual C++ Express 2010 constitue un environnement de développement complet en C et C++.

Réagissez à cet article :

I - Introduction.....	3
I-A - Révisions.....	3
I-B - Remerciements.....	3
I-C - Avant-propos.....	3
II - Installer Visual C++ Express 2010.....	3
II-A - Téléchargement.....	3
II-B - Installation.....	3
III - Premier lancement et enregistrement.....	13
IV - Éléments de configuration.....	19
IV-A - Configuration basique/expert.....	19
IV-B - Sauvegarde des paramètres de configuration.....	20
IV-C - Page d'accueil.....	21
IV-D - Répertoire des projets.....	22
IV-E - Configuration des répertoires des bibliothèques.....	23
V - Créer un projet C++.....	24
V-A - Création du projet.....	24
V-B - La solution.....	29
V-C - Créer un nouveau fichier.....	29
V-D - Ajout d'une fonction principale.....	31
V-E - Ajouter une nouvelle classe.....	32
V-F - Compiler un projet.....	38
V-G - Exécuter ou déboguer un programme.....	40
V-H - Directive de compilation Visual C++ 2010.....	42
VI - Configurer un projet C++.....	42
VI-A - Configuration générale.....	42
VI-A-1 - Sortie de la compilation.....	43
VI-A-2 - Compiler en C++0x ou en C++03.....	43
VI-A-3 - Exécutable ou bibliothèque ?.....	44
VI-A-4 - Utilisation des MFC et de l'ATL.....	44
VI-A-5 - Unicode ou pas ?.....	44
VI-B - Configuration des options de Debug.....	44
VI-C - Configurer les répertoires du projet.....	45
VI-C-1 - Dans quels répertoires Visual C++ Express cherche-t-il les fichiers d'en-tête ?.....	48
VI-C-2 - Dans quels répertoires Visual C++ Express cherche-t-il les fichiers .lib pour l'édition de lien ?.....	49
VI-D - Configuration C++ générale.....	51
VI-E - Désactiver les extensions de langage.....	52
VII - C++0x.....	53

I - Introduction

I-A - Révisions

Date	Révision article
Mai 2010	Rédaction

I-B - Remerciements

Je remercie **Arzar** pour sa relecture et ses conseils, ainsi que **eusebe19** et **ClaudeLELOUP** pour leur méticuleuse relecture.

D'une façon plus globale, je remercie les membres des forums de developpez.com qui par la qualité de leurs interventions m'ouvrent constamment de nouvelles pistes de réflexion sur ma pratique de développement.

I-C - Avant-propos


Apparues en 2005, les versions Express de Visual Studio étendent la gamme des environnements de développement proposés par Microsoft. Gratuites et disponibles pour différents langages, elles offrent un environnement de développement suffisamment mature pour des projets personnels ou semi-professionnels.

Ainsi, la sortie de Visual Studio 2010 s'est tout naturellement accompagnée de celle de Visual C++ Express 2010. Bien sûr, le studio est moins complet que les versions payantes, mais le compilateur est réputé être le même et, de toute façon, reste un bon outil de développement sur PC Windows.

Vous trouverez un  **tableau des différentes versions** sur le site de Microsoft.

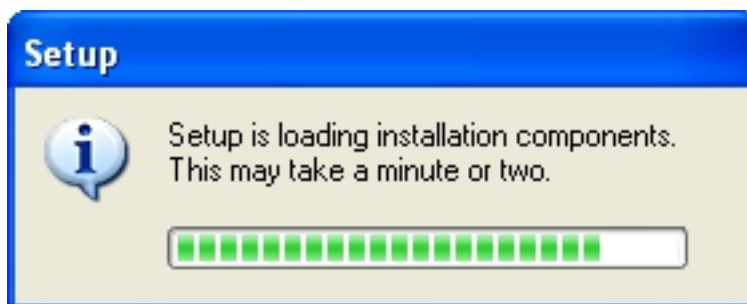
II - Installer Visual C++ Express 2010

II-A - Téléchargement

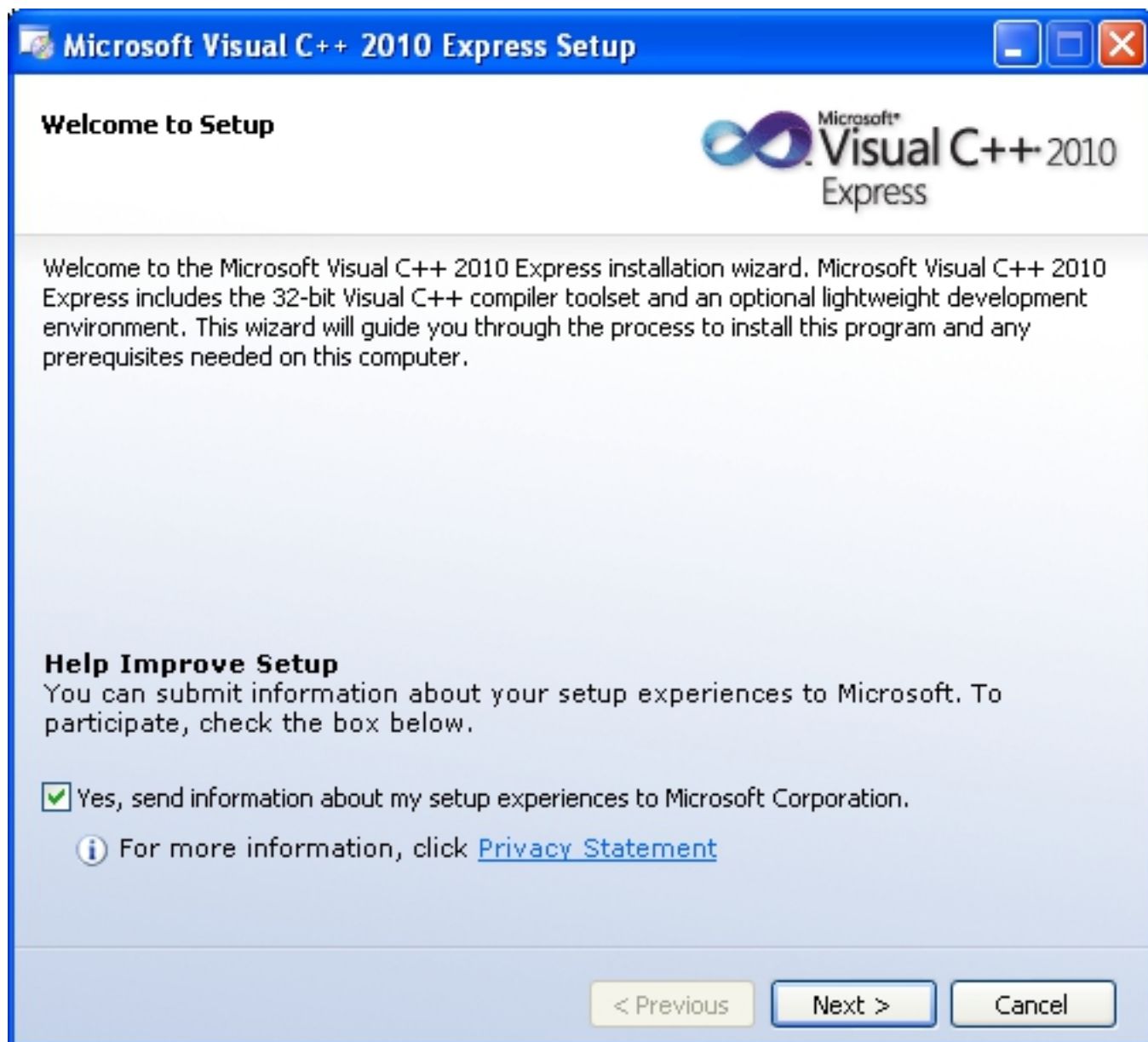
Visual C++ Express est disponible au téléchargement sur le  **site de Microsoft**. Pour réaliser cette présentation, nous avons téléchargé la version anglaise de Visual C++ Express 2010. Notre point de départ est par conséquent : vc_web.exe !

II-B - Installation

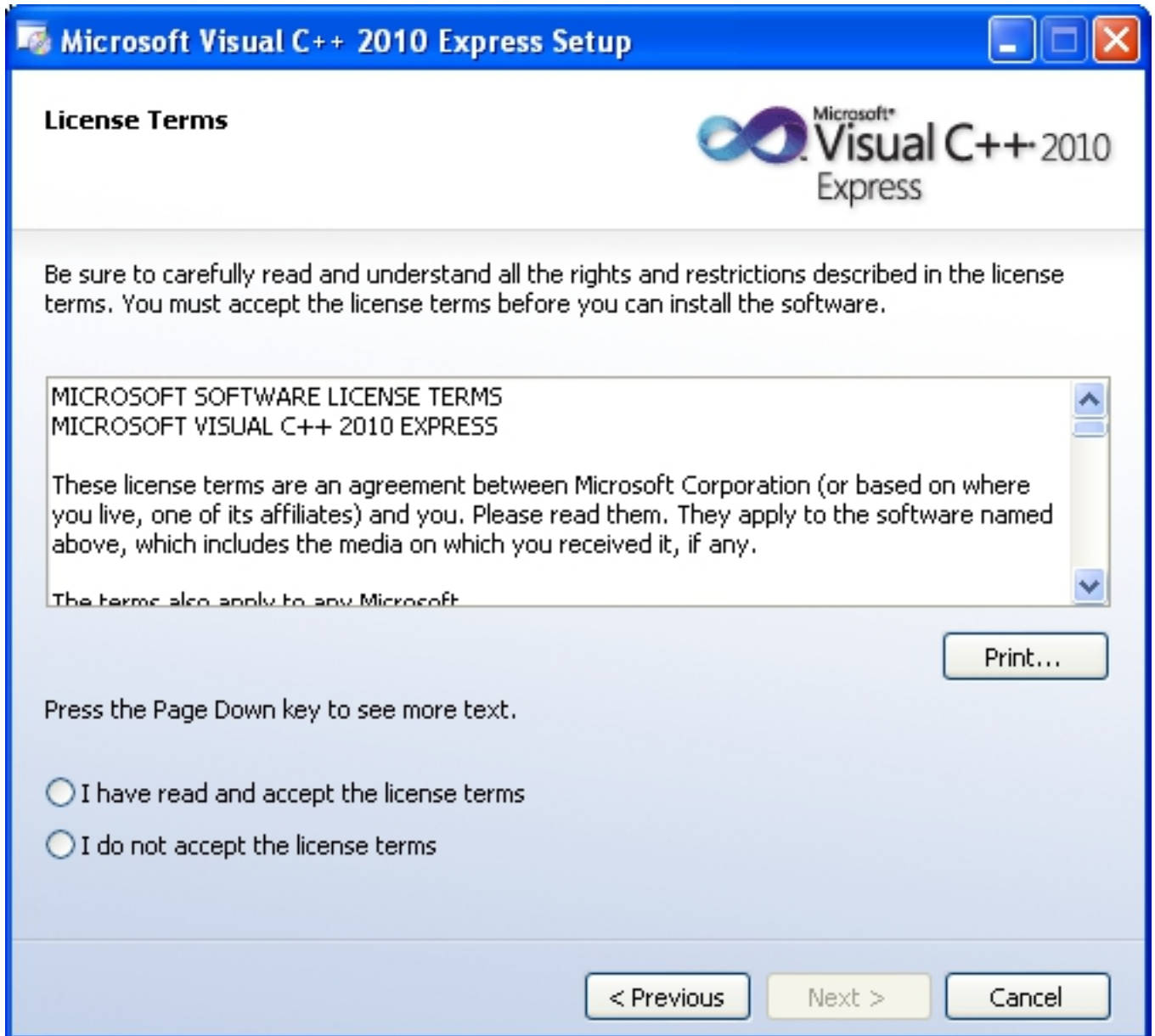
L'installation peut démarrer en exécutant le fichier que nous venons de télécharger : double-cliquez sur vc_web.exe. Ce petit fichier de 3 Mo n'est pas encore l'installateur à proprement parler. Il déclenche le téléchargement des composants nécessaires à l'installation du produit :



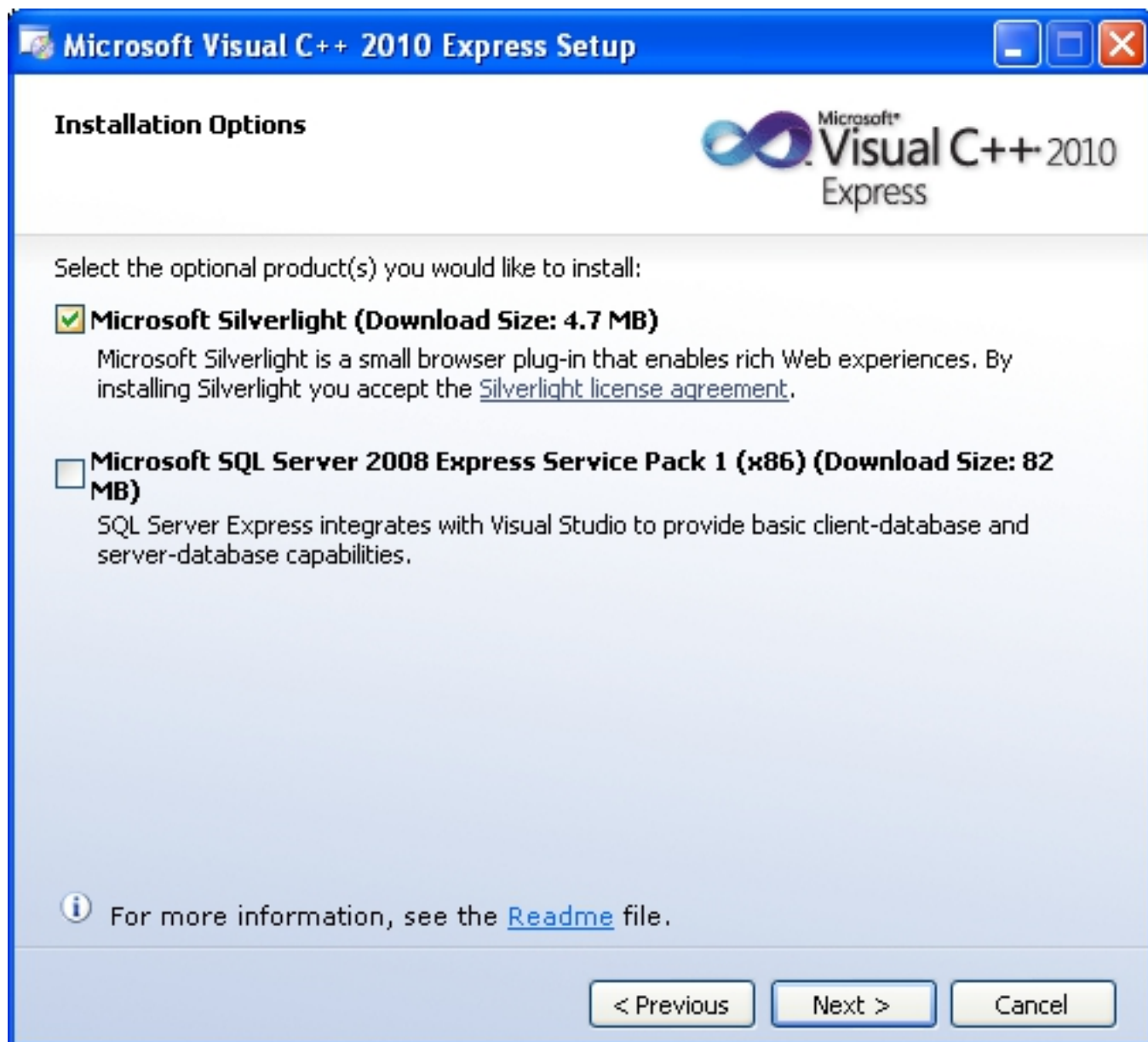
Une fois ces premiers téléchargements terminés, un assistant va nous guider tout le long de l'installation :



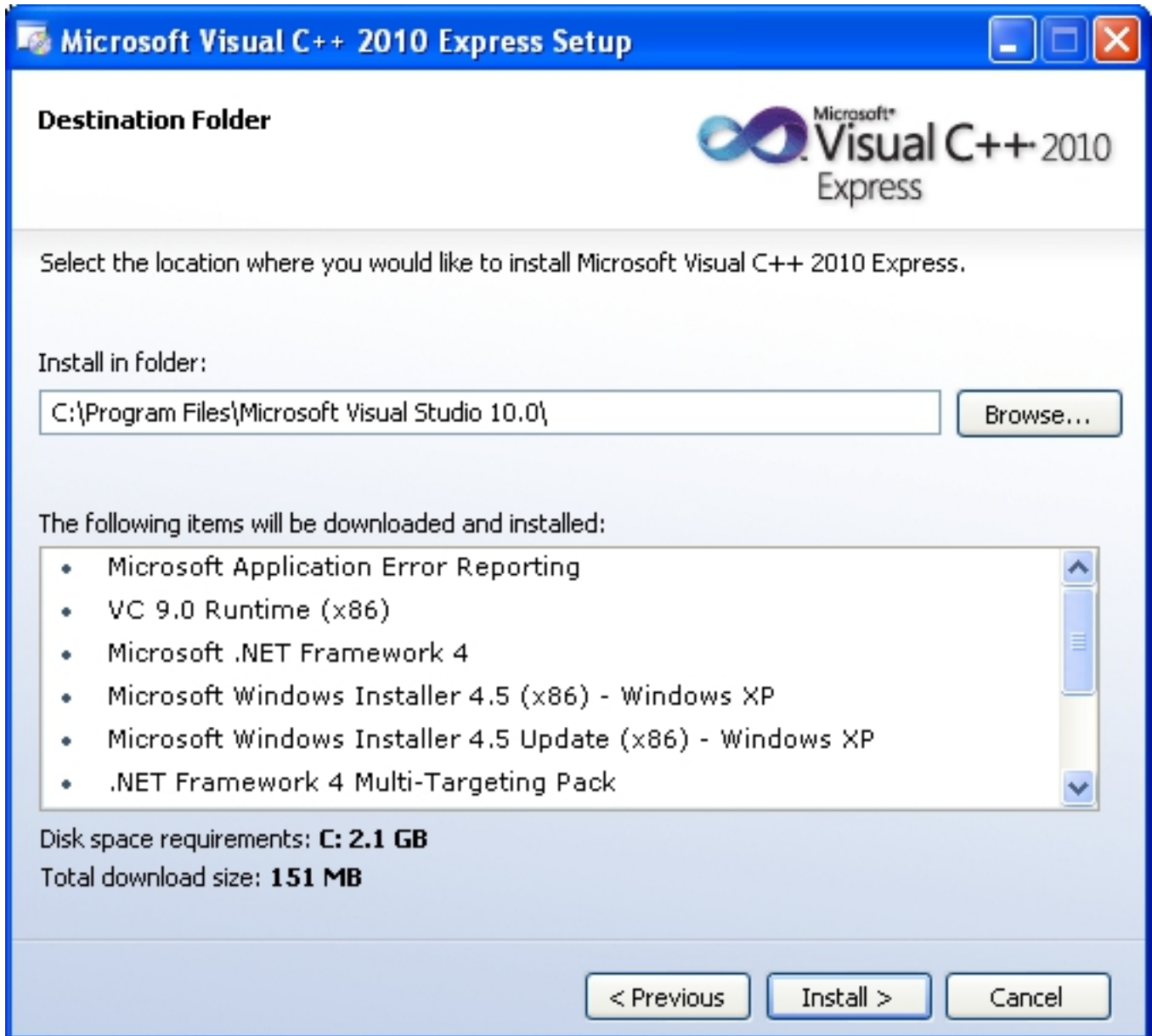
Après avoir cliqué sur suivant, nous arrivons sur les termes du contrat de la licence :



Veillez à bien lire les différents éléments et à en comprendre la teneur. Ceci fait, indiquez-le à l'assistant et passez à l'étape suivante :



Lors de l'installation de Visual C++ Express 2010, il vous est proposé d'installer également Microsoft Silverlight et Microsoft SQL Server 2008 Express Service Pack 1. Nous avons choisi de ne pas installer SQL Server mais d'installer Silverlight. Vous devez donc cocher la case correspondant à Silverlight et décocher celle de SQL Server. Il faut choisir maintenant le répertoire de destination :

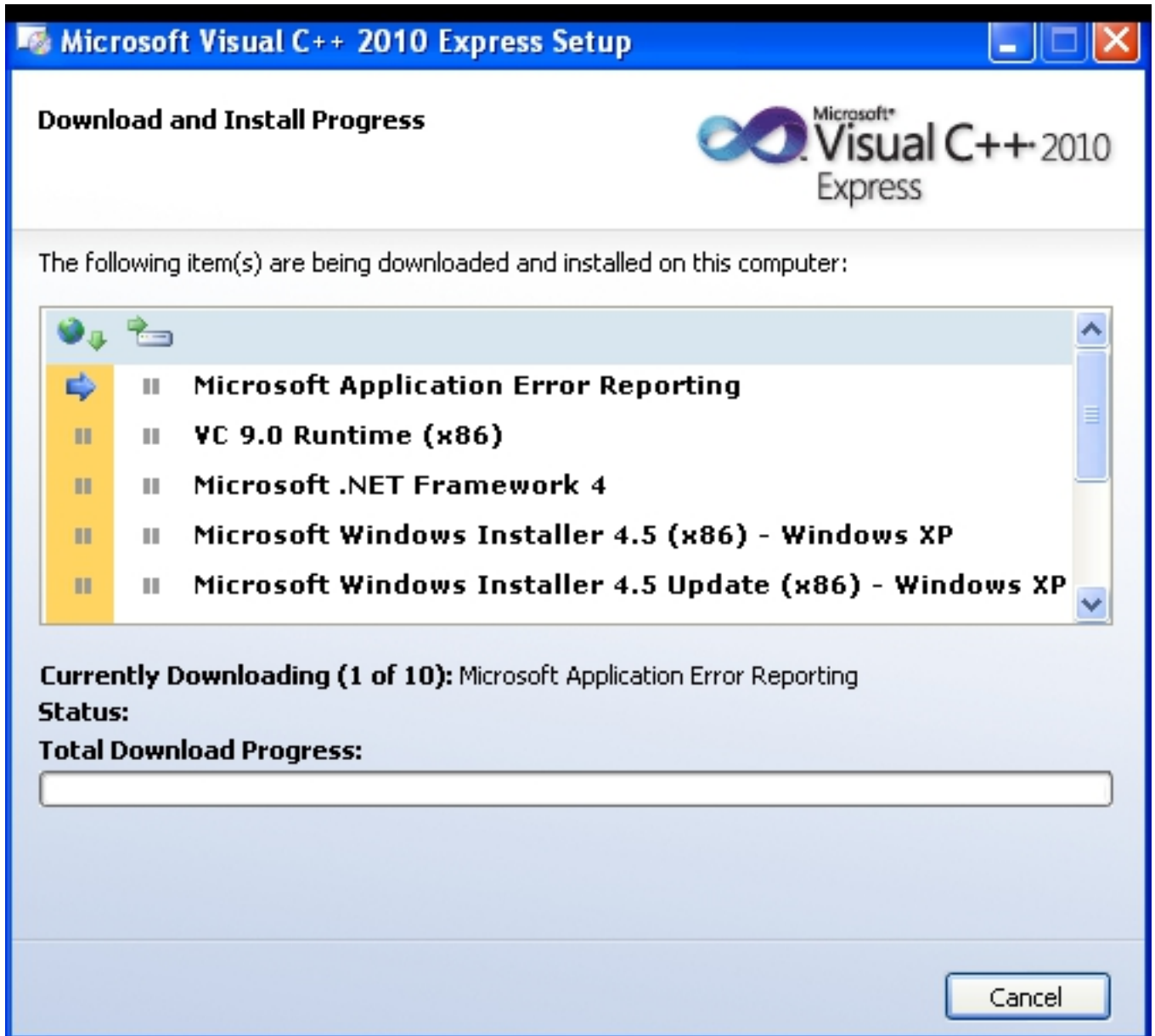


Outre le répertoire d'installation, cette boîte de dialogue nous donne deux informations intéressantes. D'abord, le volume de données téléchargées : ici un peu plus de 150 Mo. Nous allons donc devoir faire preuve de plus ou moins de patience selon la qualité de la connexion Internet... Ensuite, la liste détaille explicitement tous les composants nécessaires au fonctionnement de Visual C++ Express 2010. Cette liste dépend des éléments éventuellement déjà présents. Pour votre installation, cette liste peut varier puisqu'elle est directement liée à la configuration de votre poste. Ici, nous sommes parti d'un PC neuf sur lequel tout ou presque était à télécharger et à installer. Par conséquent, ce sont pas moins de dix composants qui sont traités :

- Microsoft Application Error Reporting ;
- VC 9.0 Runtime (x86) ;
- Microsoft .Net Framework 4 ;
- Microsoft Windows Installer 4.5 (x86) - Windows XP ;
- Microsoft Windows Installer 4.5. Update (x86) - Windows XP ;
- .NET Framework 4 Multi-Targeting Pack ;
- Microsoft Visual C++ 2010 Express ;
- Microsoft SQL Server Compact 3.5 SP2 (x86) ENU ;
- Microsoft SilverLight ;
- Microsoft Help Viewer 1.0 x86.

Si d'aventure vous souhaitez désinstaller la version Visual C++ Express Edition 2010, il faut veiller à désinstaller ces différents produits à partir du gestionnaire d'ajout et de suppression des programmes. Mais terminons d'abord cette installation.

Une fois l'étape précédente franchie, la procédure se décompose en deux phases. Au départ, l'ensemble des composants est téléchargé :



Selon le volume total de données à télécharger et le débit de votre liaison Internet, ce téléchargement peut être plus ou moins long. La petite barre située à gauche de la liste indique les éléments déjà téléchargés (petite coche verte), l'élément en cours de téléchargement (la flèche bleue) et les éléments restant à télécharger (les deux points gris) :

Microsoft Visual C++ 2010 Express Setup

Download and Install Progress

The following item(s) are being downloaded and installed on this computer:

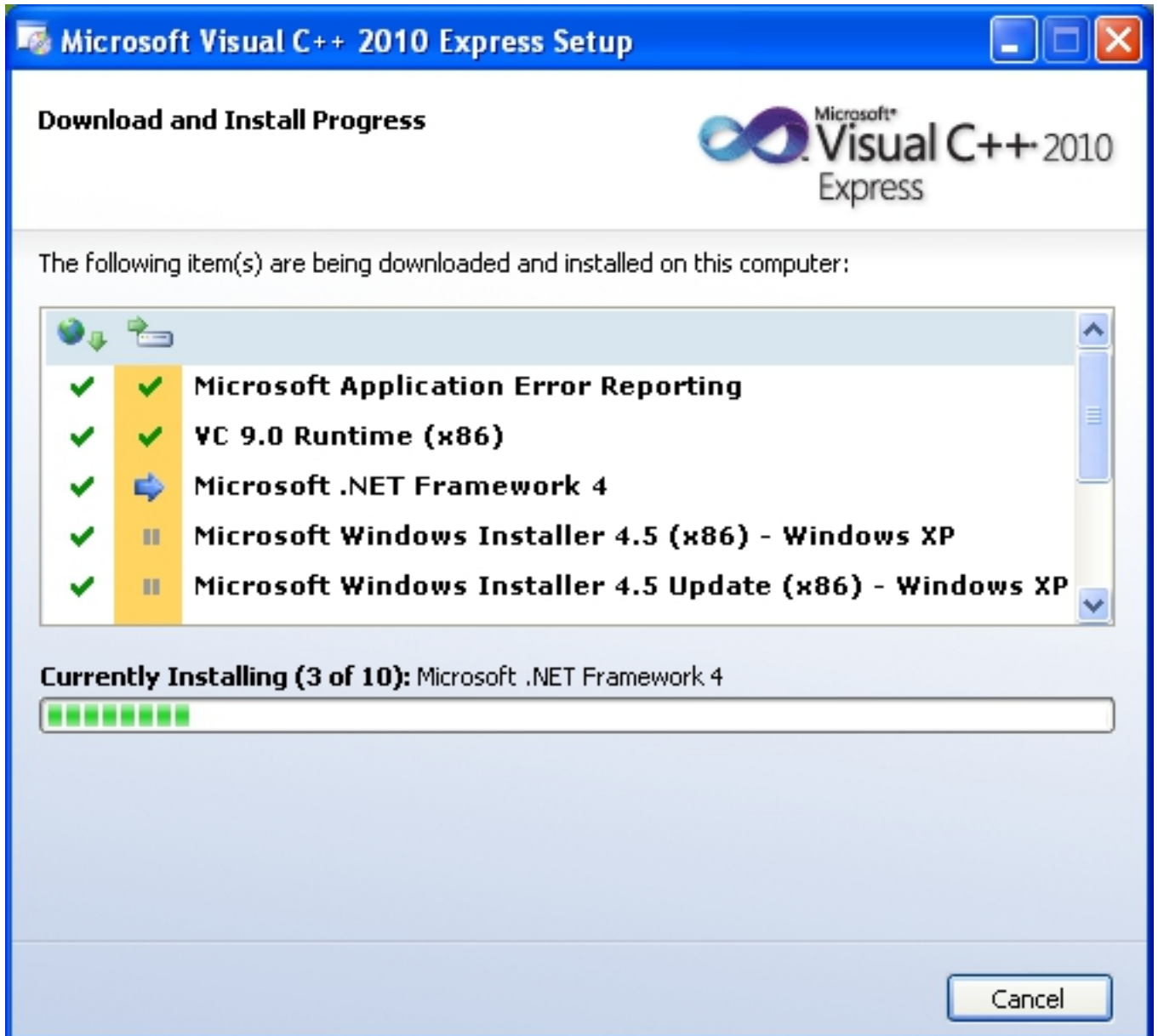
Statut	Item
✓	Microsoft Application Error Reporting
✓	VC 9.0 Runtime (x86)
✓	Microsoft .NET Framework 4
➔	Microsoft Windows Installer 4.5 (x86) - Windows XP
	Microsoft Windows Installer 4.5 Update (x86) - Windows

Eléments déjà téléchargés
 Téléchargement en cours
 Restant à télécharger

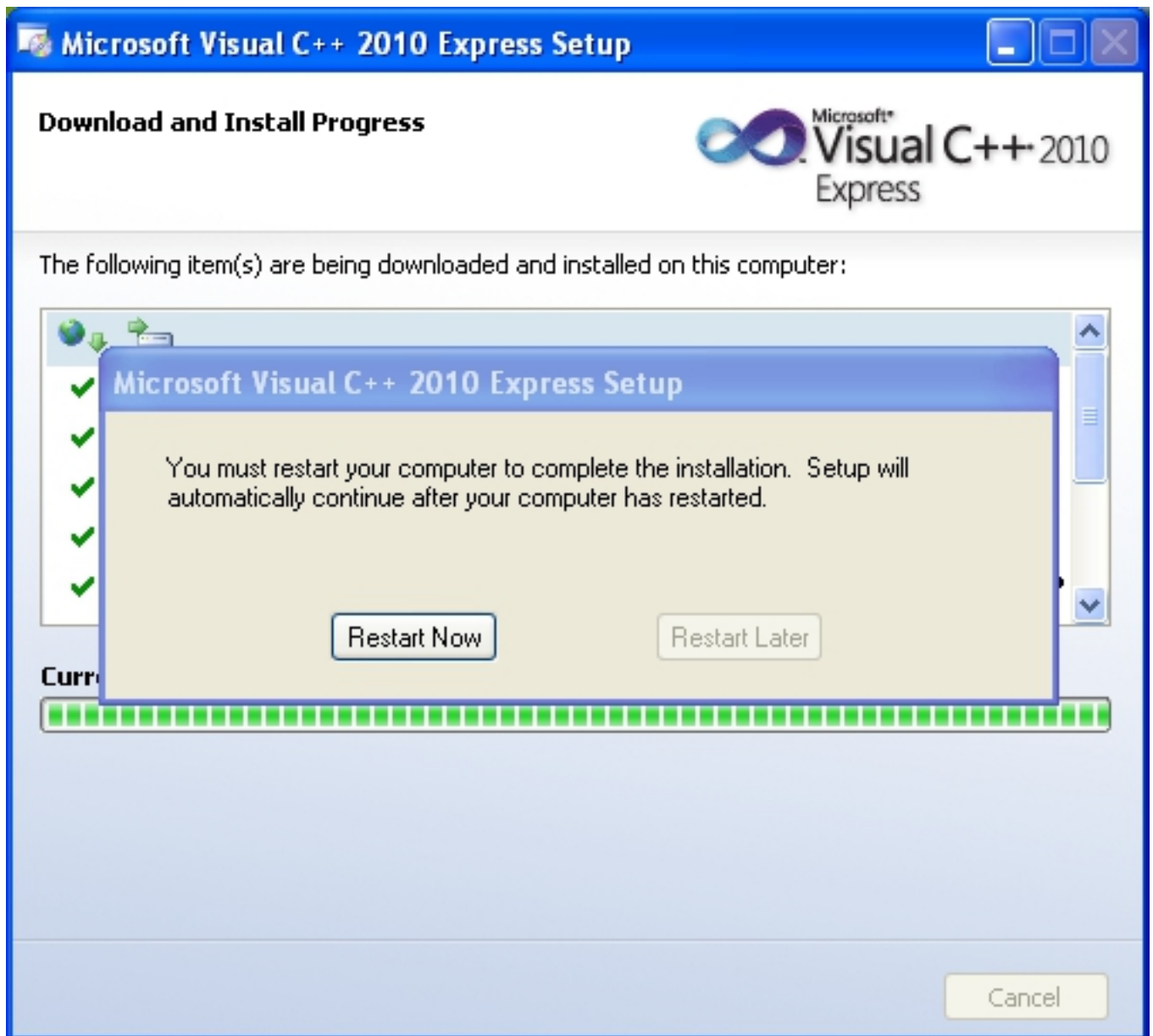
Currently Downloading (4 of 10): Microsoft Windows Installer 4.5 (x86) - Windows XP
Status: Current transfer rate is 49 KB/sec.
Total Download Progress: 54 MB / 151 MB

Barre de progression

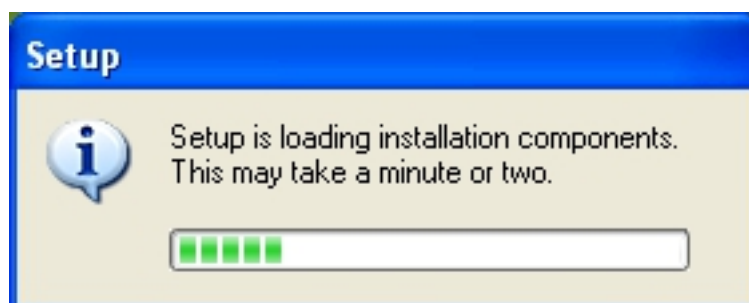
Une fois les téléchargements des différents composants terminés, l'installation commence. La même fenêtre d'avancement devrait éviter le dépaysement :



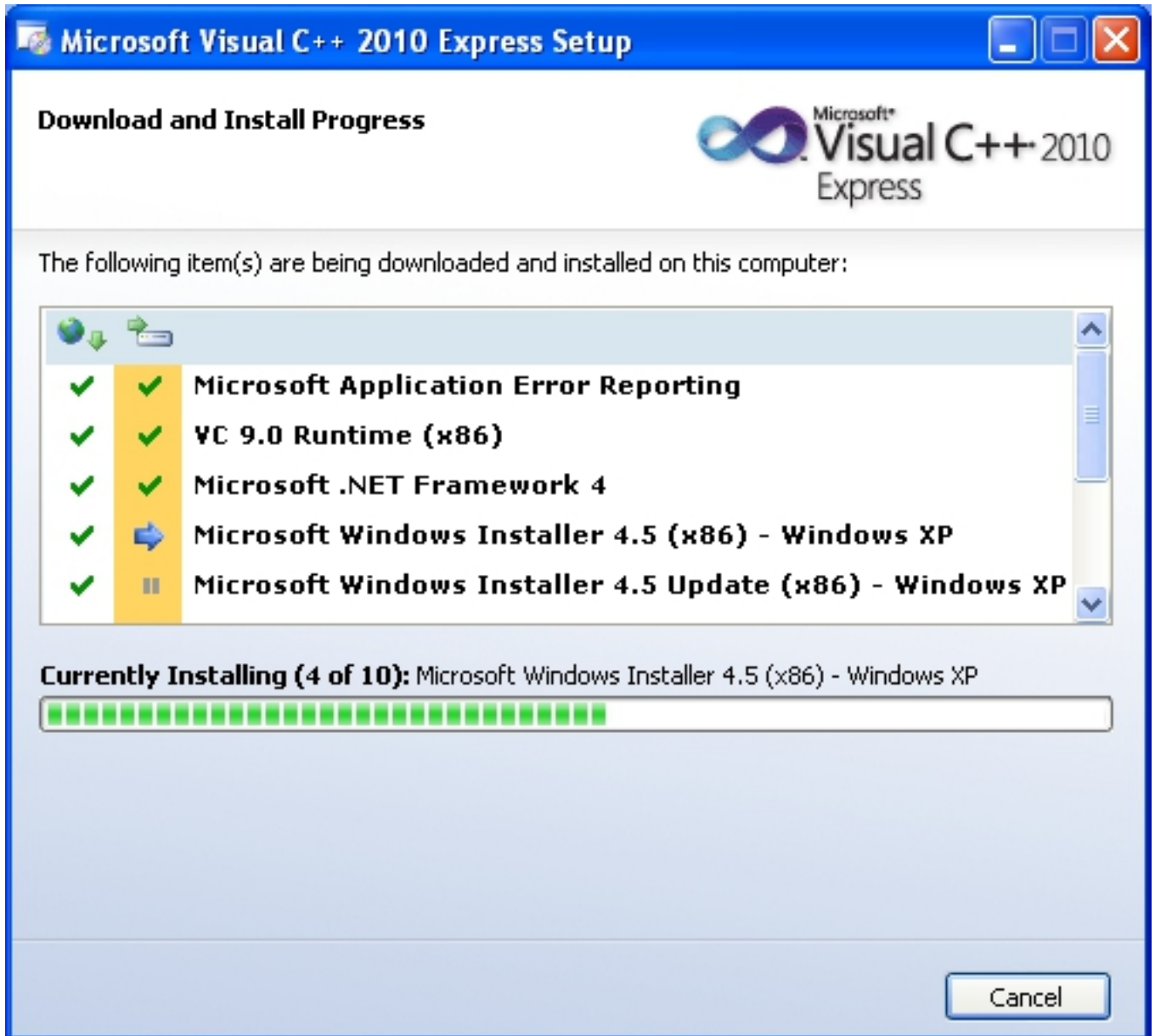
Selon votre environnement, certains composants peuvent nécessiter le redémarrage du système pour valider leur installation :



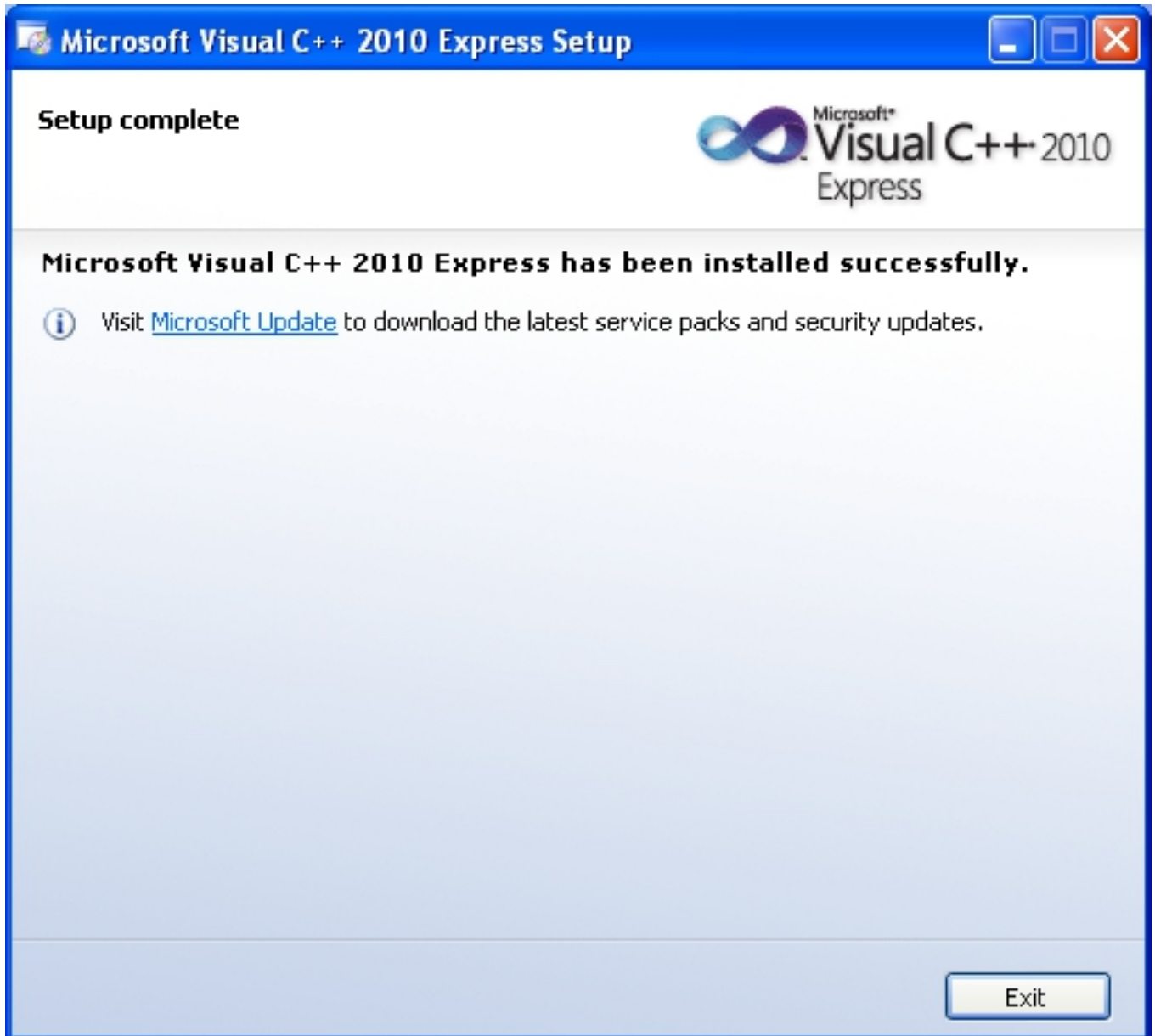
Cliquez simplement sur la demande de redémarrage. L'installateur arrête et redémarre le système. À cette reprise, l'installation repart par ce message :



Puis l'installation se poursuit avec la fenêtre d'avancement maintenant bien connue :



Une fois la liste des composants égrenée, l'installation de Visual C++ Express 2010 se termine :



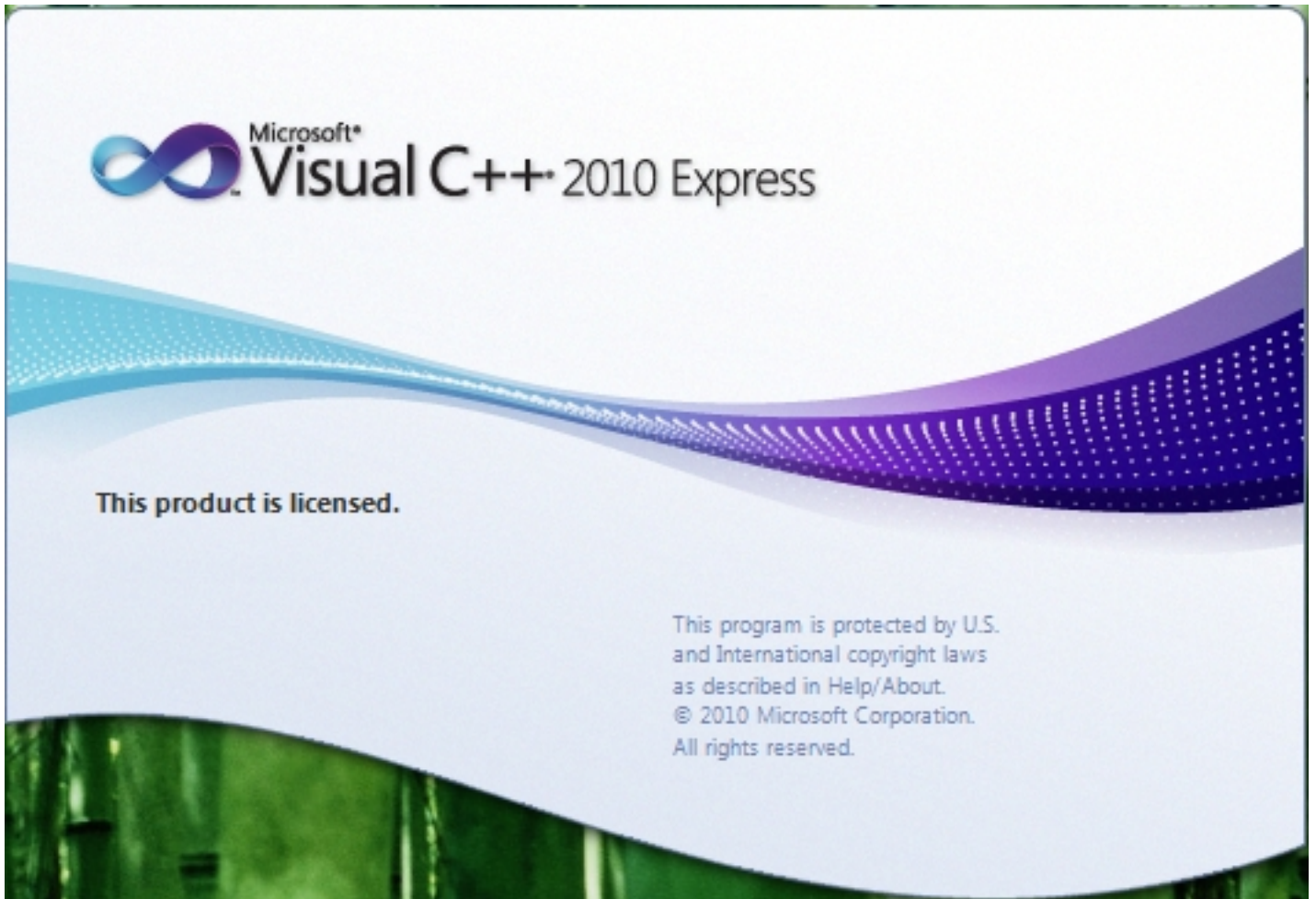
III - Premier lancement et enregistrement

Il est temps de lancer Visual C++ Express 2010. L'icône peut se trouver sur votre bureau, sur la barre de lancement rapide ou dans le menu démarrer. Si vous ne la trouvez pas, le programme de lancement se trouve (en fonction du répertoire d'installation) en suivant ce chemin :

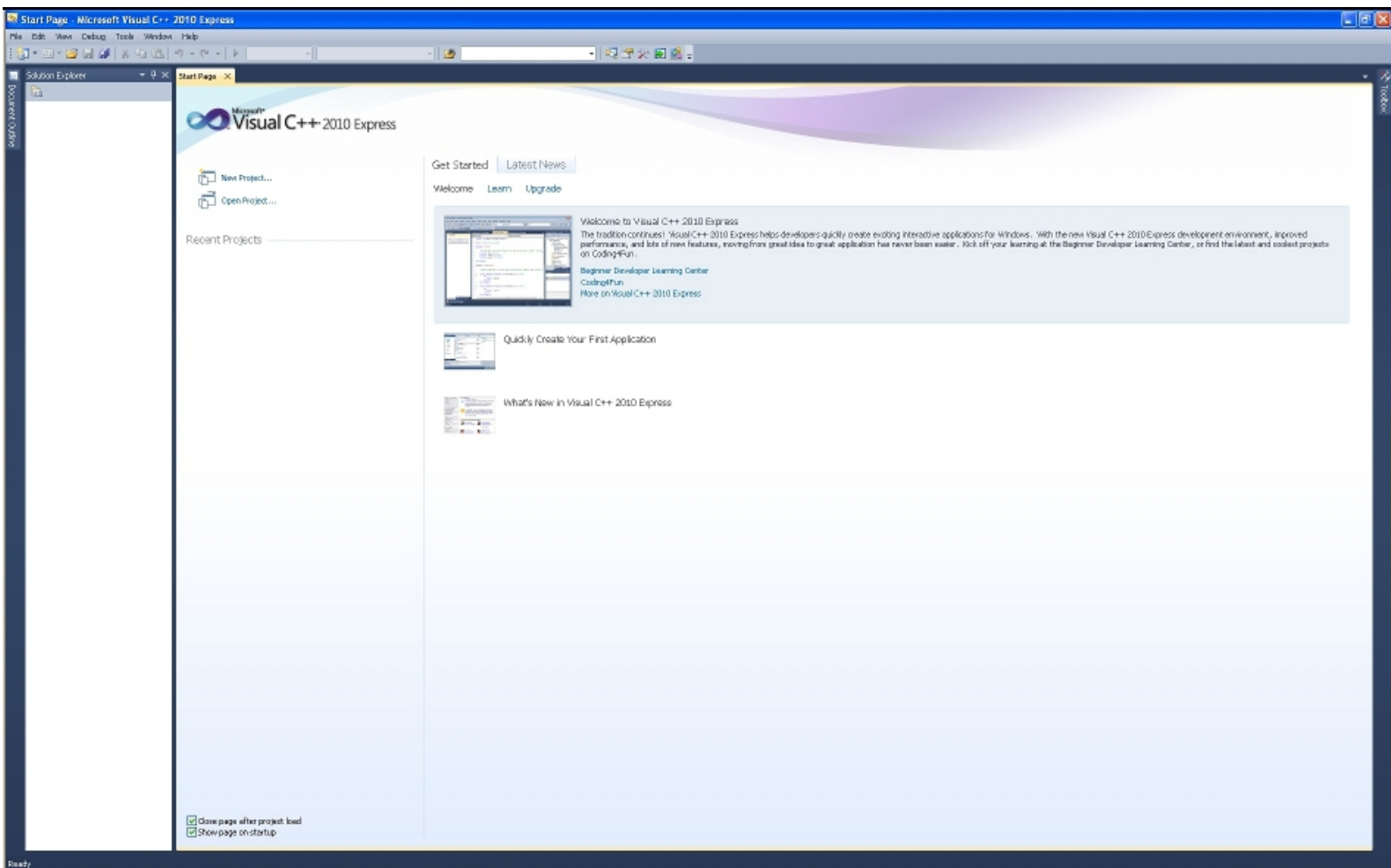
- C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE.

L'exécutable lançant le studio de développement est VCEXpress.exe.

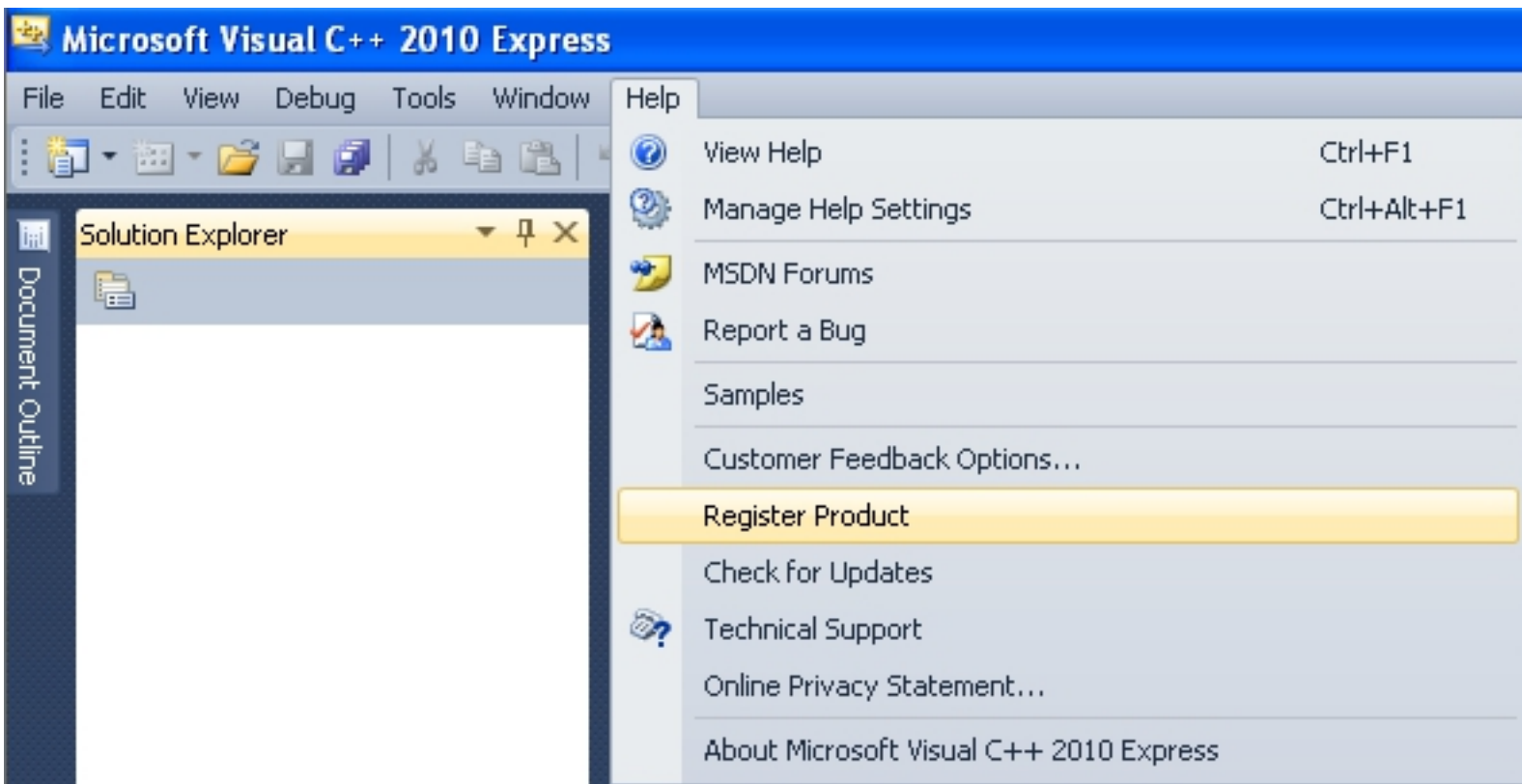
Le démarrage de Visual C++ Express 2010 est légèrement plus lent que son prédécesseur Visual C++ Express 2008 mais cela reste très raisonnable. Cela commence par l'affichage de l'image suivante :



Puis voilà la première page affichée :



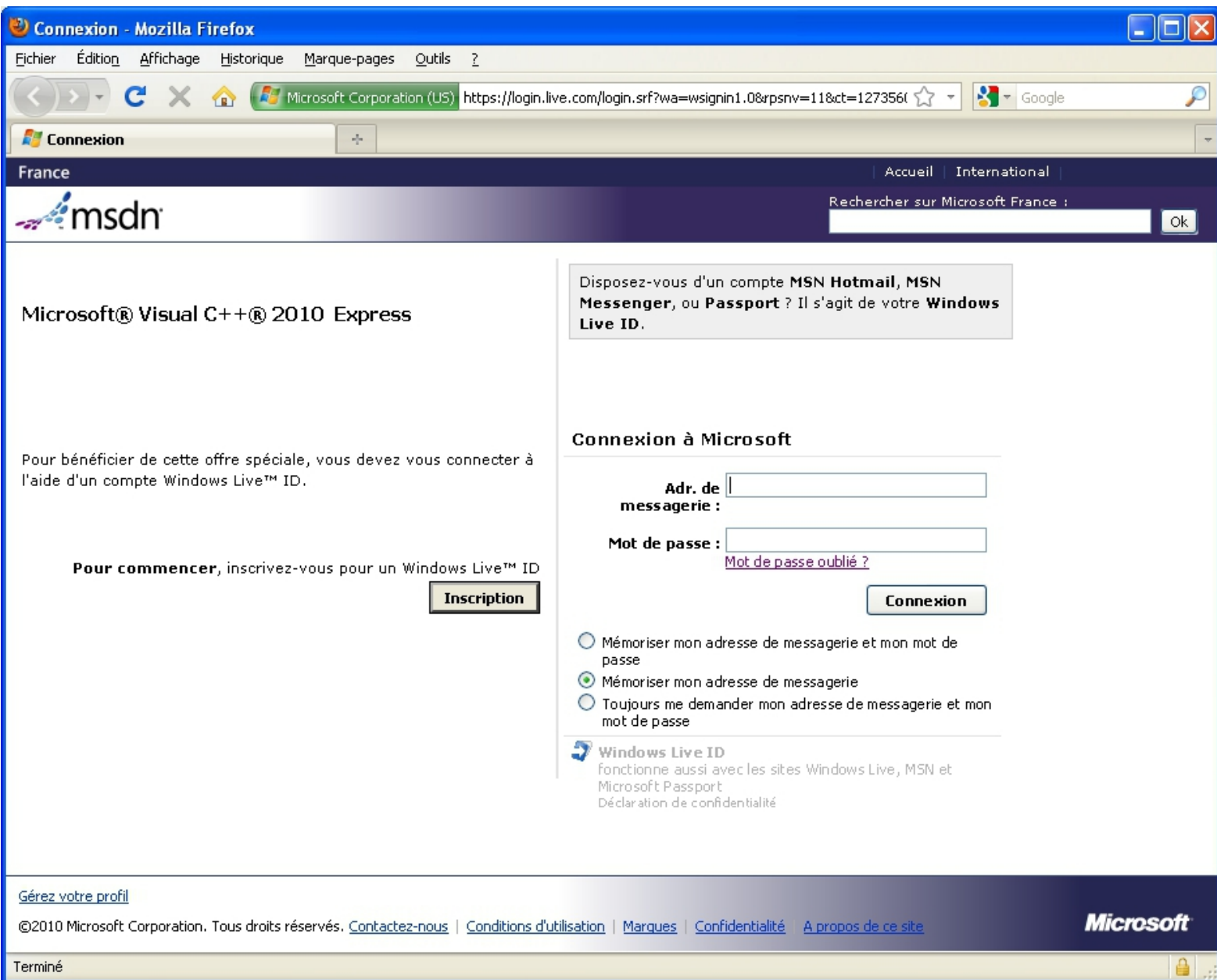
À ce moment de l'installation, Visual C++ Express 2010 n'est valide que pour 30 jours. Pour pérenniser cette installation et pouvoir bénéficier de cet outil sans durée de limitation, vous devez vous enregistrer auprès de Microsoft. Cet enregistrement est gratuit et n'aboutit pas à remplir votre boîte aux lettres de publicité. Allez dans le menu Help (Aide) et choisissez l'option d'enregistrement du produit :



Une boîte de dialogue vous invite alors à saisir une clé d'enregistrement. Pour l'obtenir, il faut appuyer sur le bouton dédié à cet effet :



Votre navigateur internet s'ouvre alors sur la page d'enregistrement :



Vous vous connectez avec votre identifiant MSN. Si vous n'en avez pas, vous pouvez en créer un.

Une fois connecté, le site vous fournit votre identifiant. Sélectionnez-le et copiez-le dans le presse-papier :

Product Registration

Thank you for Registering!

Tout d'abord, nous vous remercions pour votre inscription. Veuillez consulter les instructions sur la manière d'activer votre produit.

Le saviez-vous?

- Si vous êtes étudiant, Microsoft vous offre une version d'évaluation de Visual Studio 2010 Express pour Windows. Consultez le programme de l'étudiant.
- Vous pouvez optimiser vos applications en utilisant la nouvelle plateforme de Windows Phone. Consultez Visual Studio 2010 Express pour Windows Phone.
- Afin de bénéficier des nouvelles fonctionnalités et améliorations des produits Visual Studio, vous devez installer la version d'évaluation de Visual Studio 2010 Express sur la même machine.

Informations importantes



En tant que remerciement de votre inscription, nous vous offrons une version d'évaluation Visual Studio 2010 Express professionnelle valable 90 jours. Après avoir installé la version d'évaluation de Visual Studio 2010 Express, vous pouvez étendre la période d'essai à 90 jours.

[Extend Trial to 90 days](#)

Product Registration Key

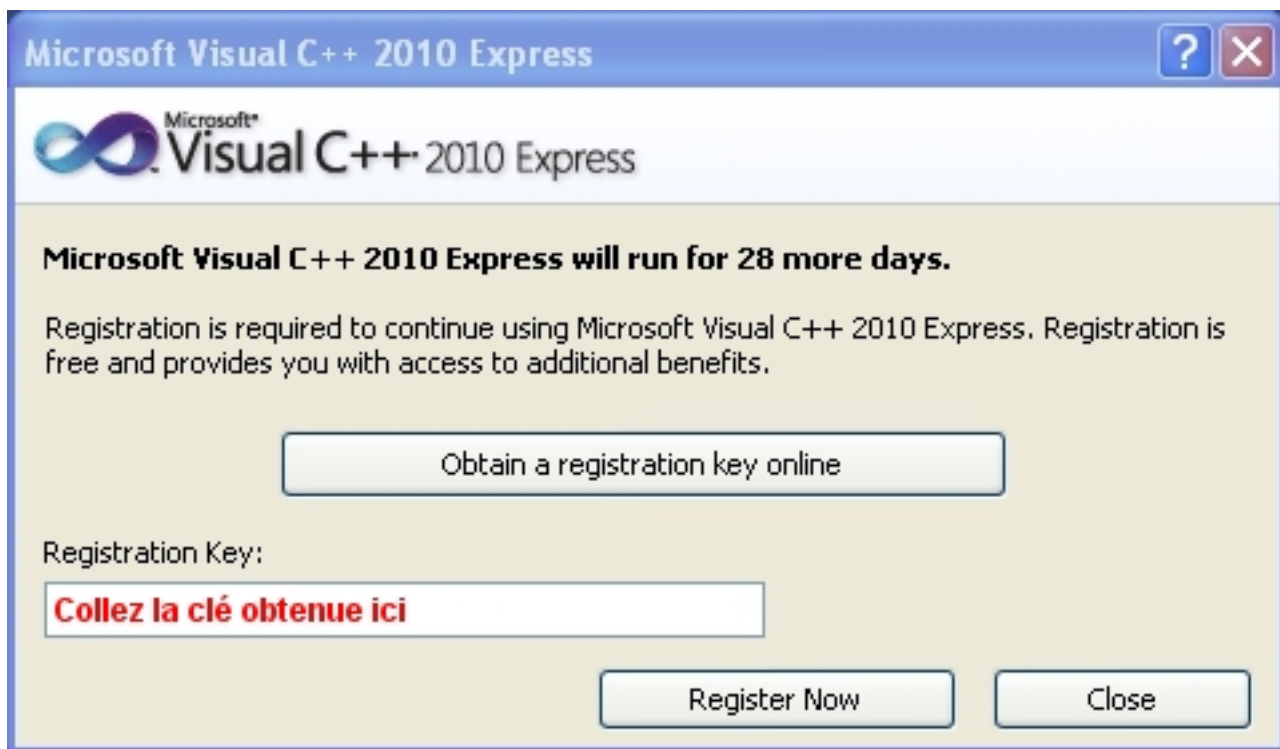
6U9*7R6 2PH-K(B18T XATK-3)W6

To copy the registration key, highlight the key, and press **Ctrl + C** or right-click the mouse button and click **Copy**.

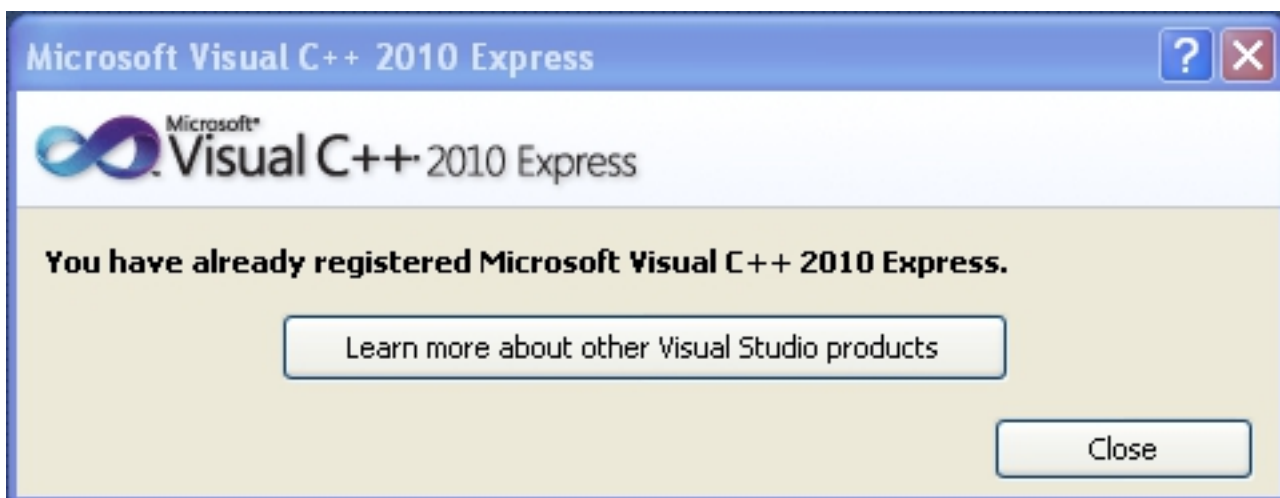
Note: An e-mail has been sent to your Windows Live ID account containing your product registration key. Please keep the e-mail for future reference. If you have not received the e-mail in the next few hours, please check your spam folder.

- Copier
- Tout sélectionner
- Recherche Google pour «...»
- Code source de la sélection

Il ne vous reste plus qu'à le coller dans la zone d'édition demandant la clé d'enregistrement et à valider :



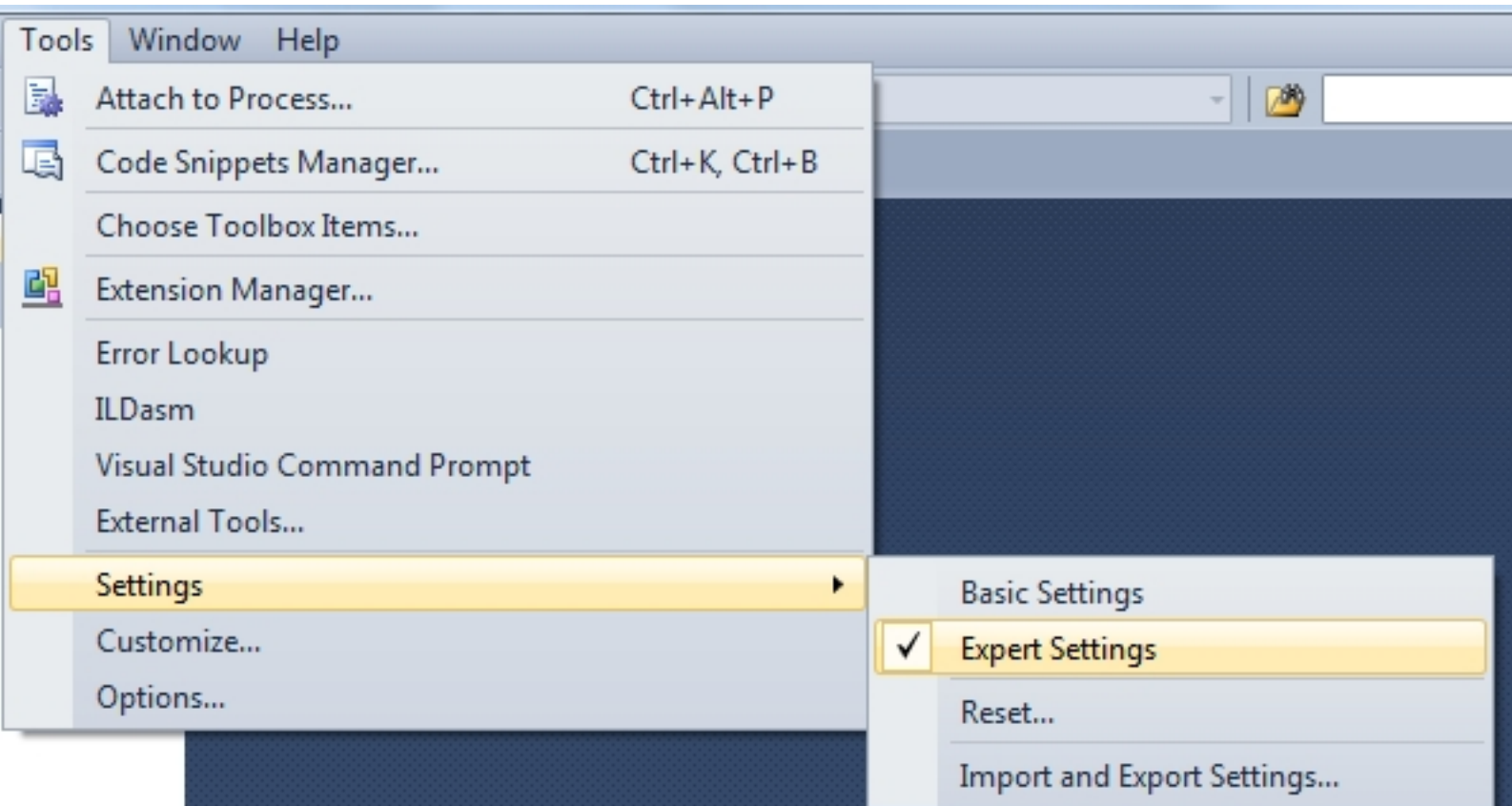
Un message vient vous confirmer de la réussite de l'enregistrement :



IV - Éléments de configuration

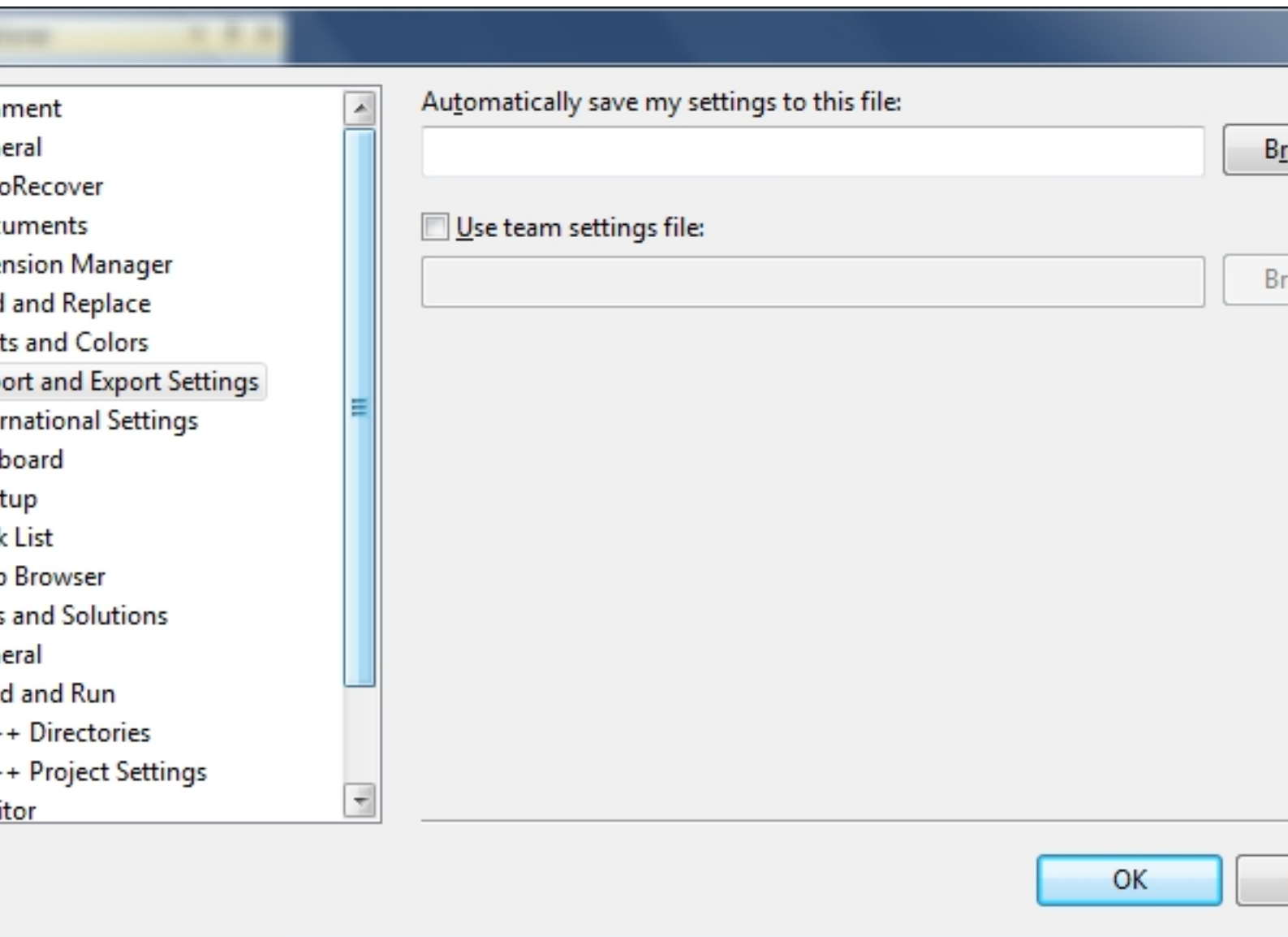
IV-A - Configuration basique/expert

Première surprise : au premier lancement, le menu de Visual C++ Express 2010 semble bien allégé par rapport aux versions précédentes. L'I.D.E. propose de fonctionner en mode basique ou en mode expert. Pour faciliter la configuration, positionnons-nous en mode expert :



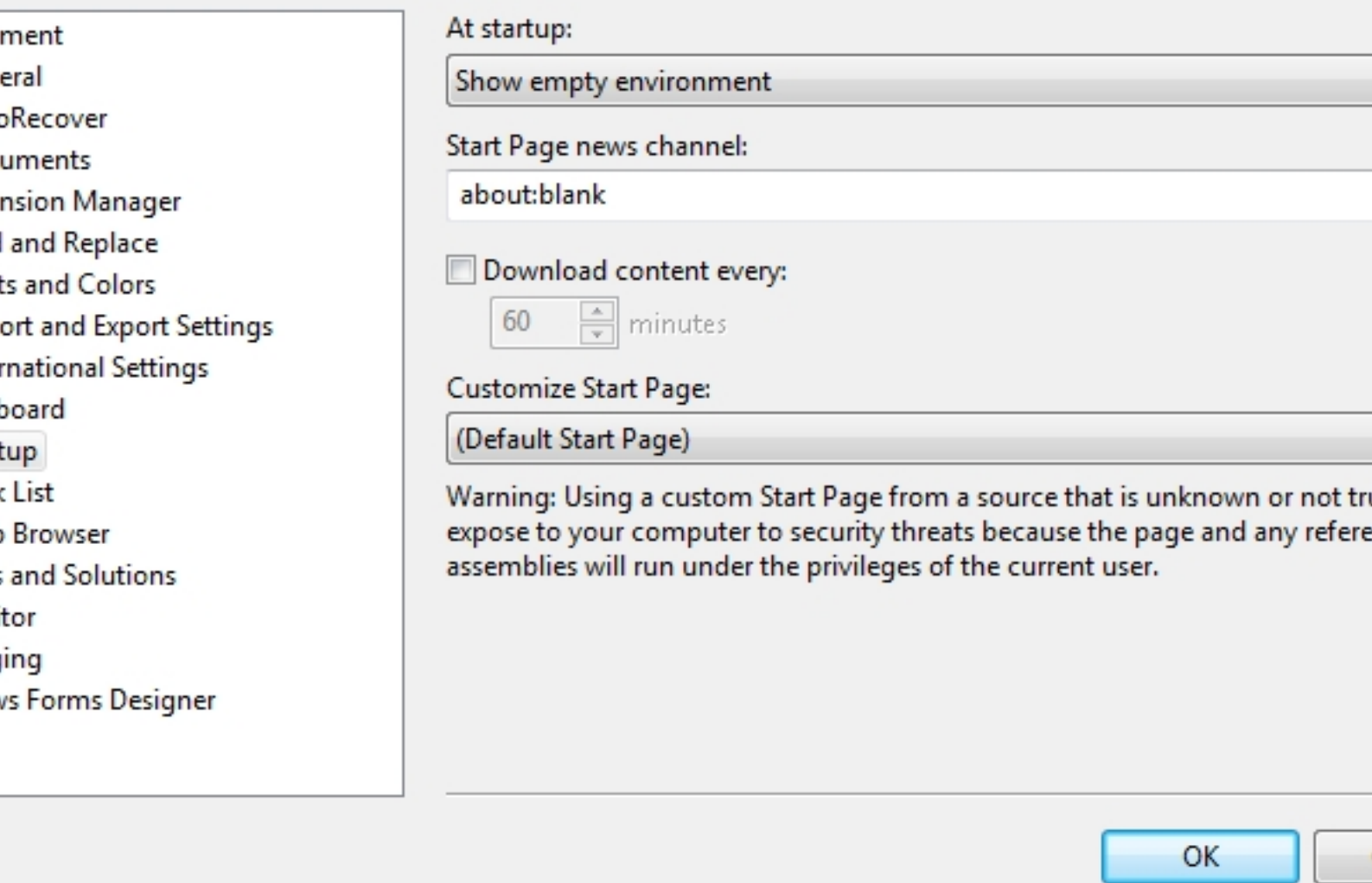
IV-B - Sauvegarde des paramètres de configuration

Il est possible de spécifier le répertoire et le fichier où sont sauvegardés les paramètres de configuration :



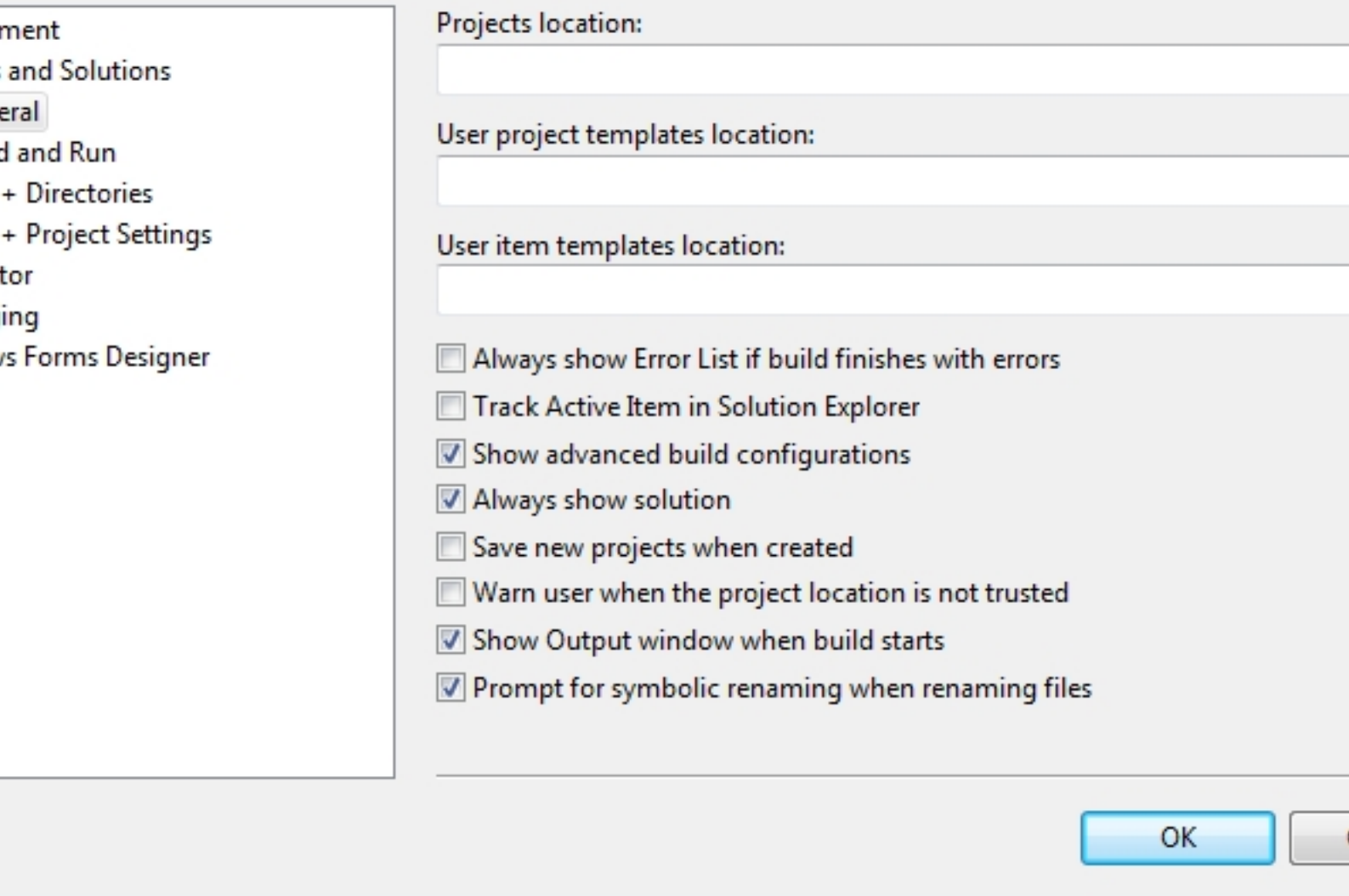
IV-C - Page d'accueil

La page affichée au démarrage de Visual C++ Express 2010 peut être modifiée :



IV-D - Répertoire des projets

Dans l'onglet *Général* des options de configuration pour les projets et solutions, vous pouvez spécifier le répertoire où Visual C++ Express 2010 doit créer ceux-ci :



IV-E - Configuration des répertoires des bibliothèques

La configuration des répertoires de recherche des fichiers d'en-têtes (.h) pour les bibliothèques tierces ne peut plus se faire au niveau global de Visual C++ Express 2010 mais se situe désormais au niveau projet :

ment
s and Solutions
eral
d and Run
+ Directories
+ Project Settings
itor
jing
ws Forms Designer

VC++ Directories editing in Tools > Options has been deprecated.

VC++ Directories are now available as a user property sheet that is added by all projects. Please click '?' for more information.

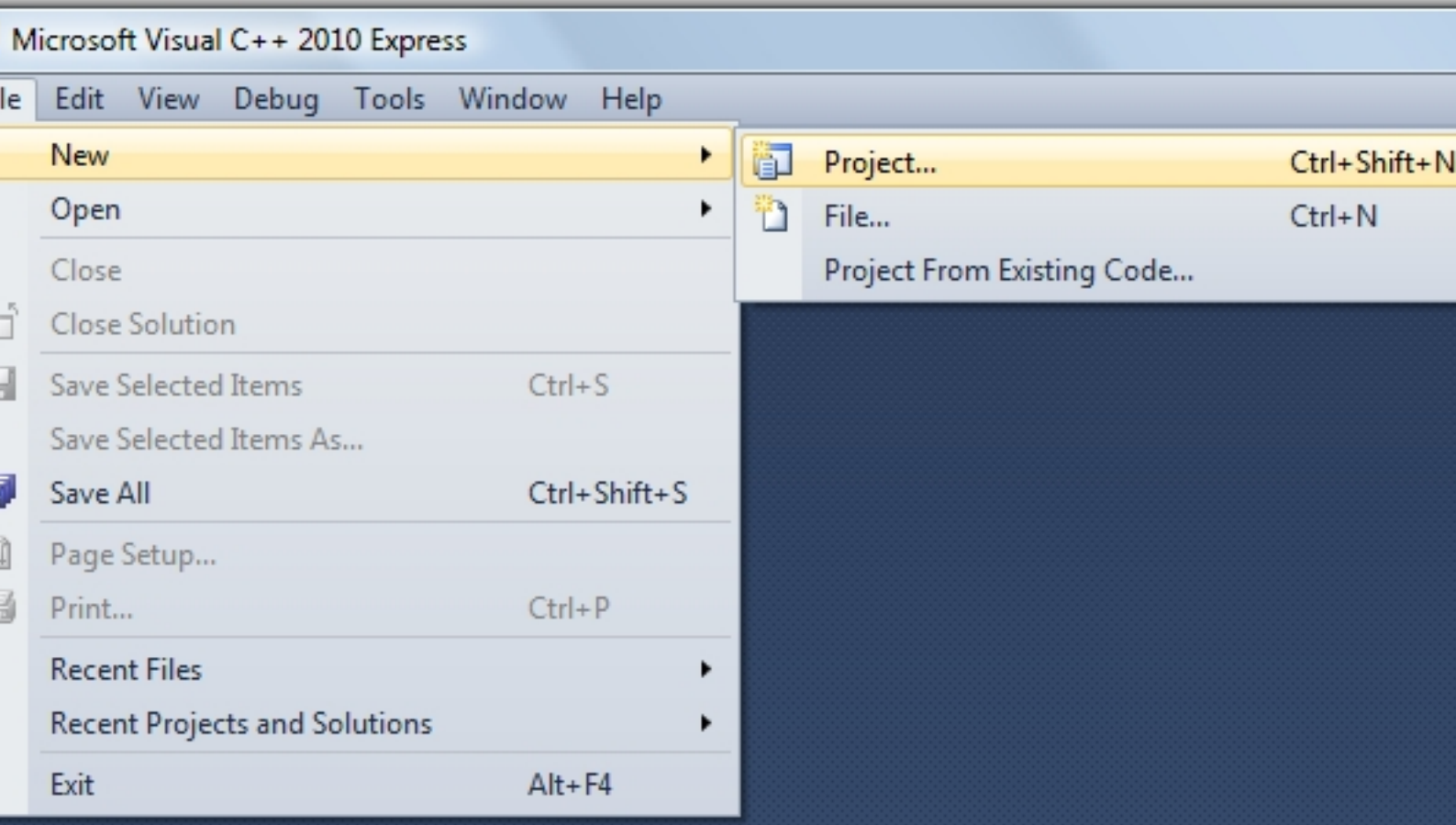
OK

Voyons donc comment créer un nouveau projet.

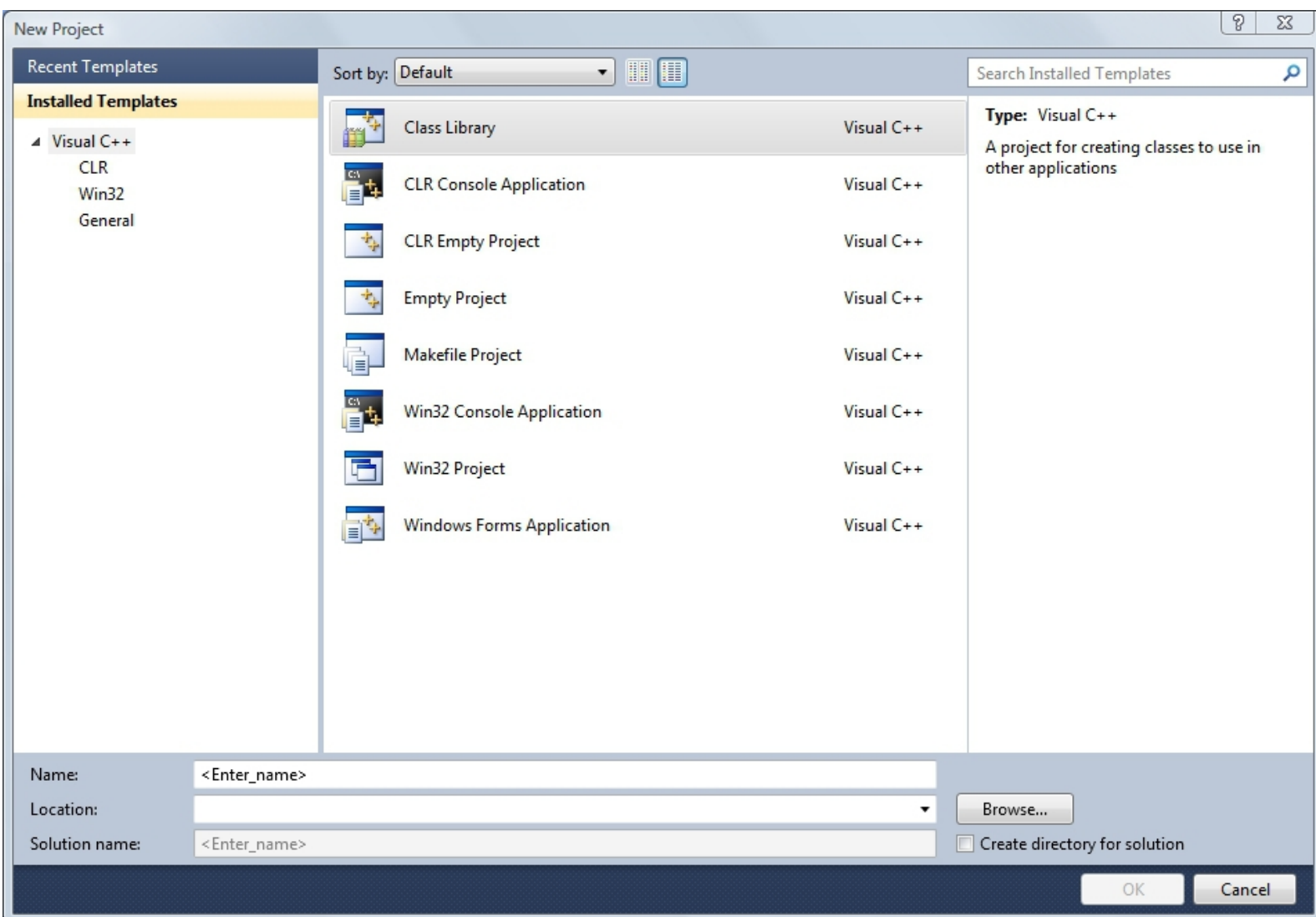
V - Créer un projet C++

V-A - Création du projet

Nous allons dans le menu 'Fichier' ('File'), puis l'item 'Nouveau' ('New') et choisissons 'Projet' ('Project') :



Un assistant s'affiche et nous propose les différents types de projets qu'il est possible de créer :



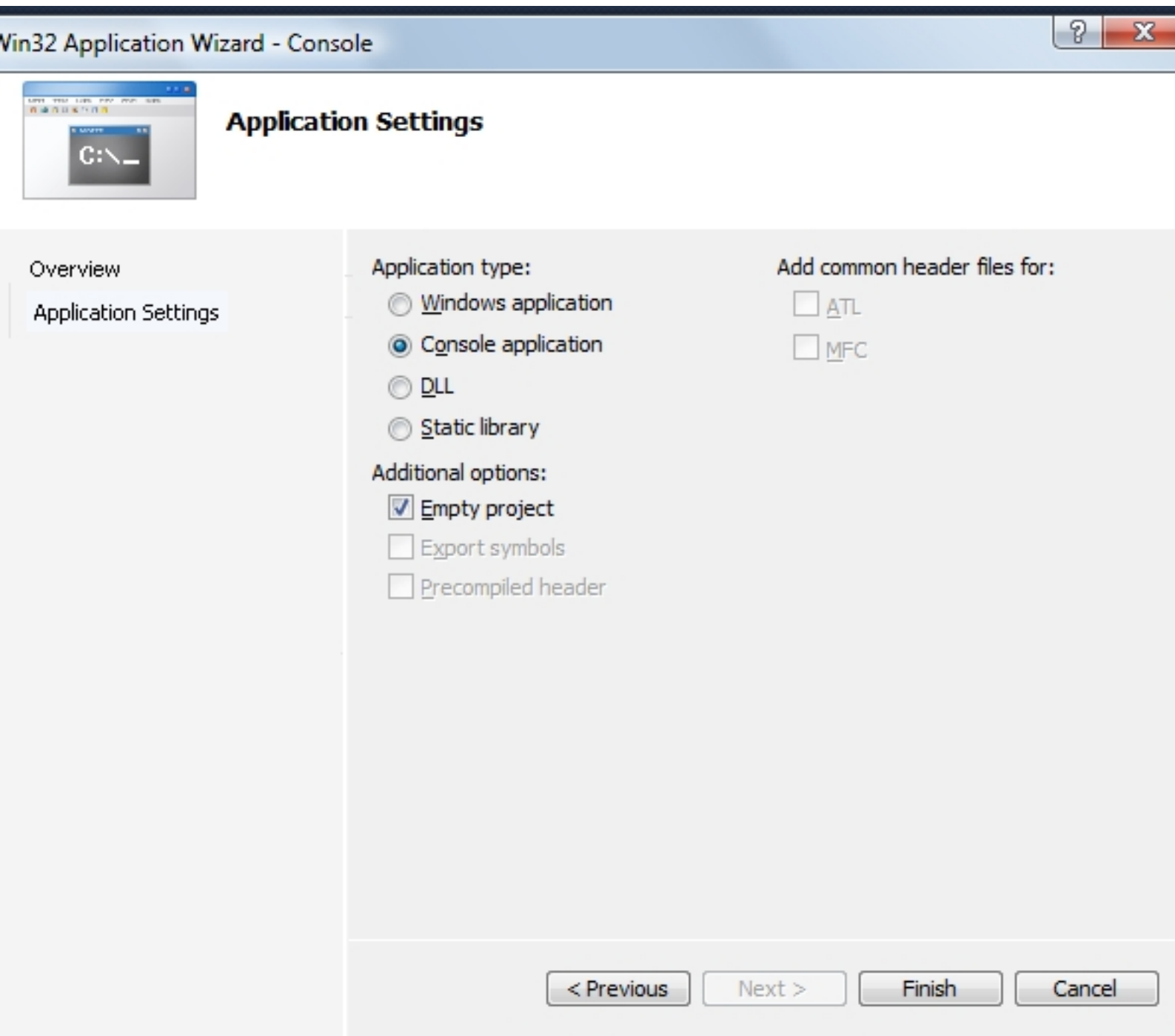
Les projets disponibles sont :

- Class Library : permet de créer une DLL pour .Net en C++/CLI ;
- CLR Console Application : permet de créer une application console en C++/CLI (.Net) ;
- CLR Empty Project : créer un projet vide configuré pour C++/CLI (.Net) ;
- Empty Project : un projet vide configuré à priori comme une application native ;
- Makefile Project : un projet de type makefile permet d'utiliser l'I.D.E. Visual Express avec une génération différente ;
- Win32 Console Application : une application console native en C++ ;
- Win32 Project : une application Windows native en C++ ;
- Windows Forms Application : une application C++/CLI (.Net) à base de Forms.

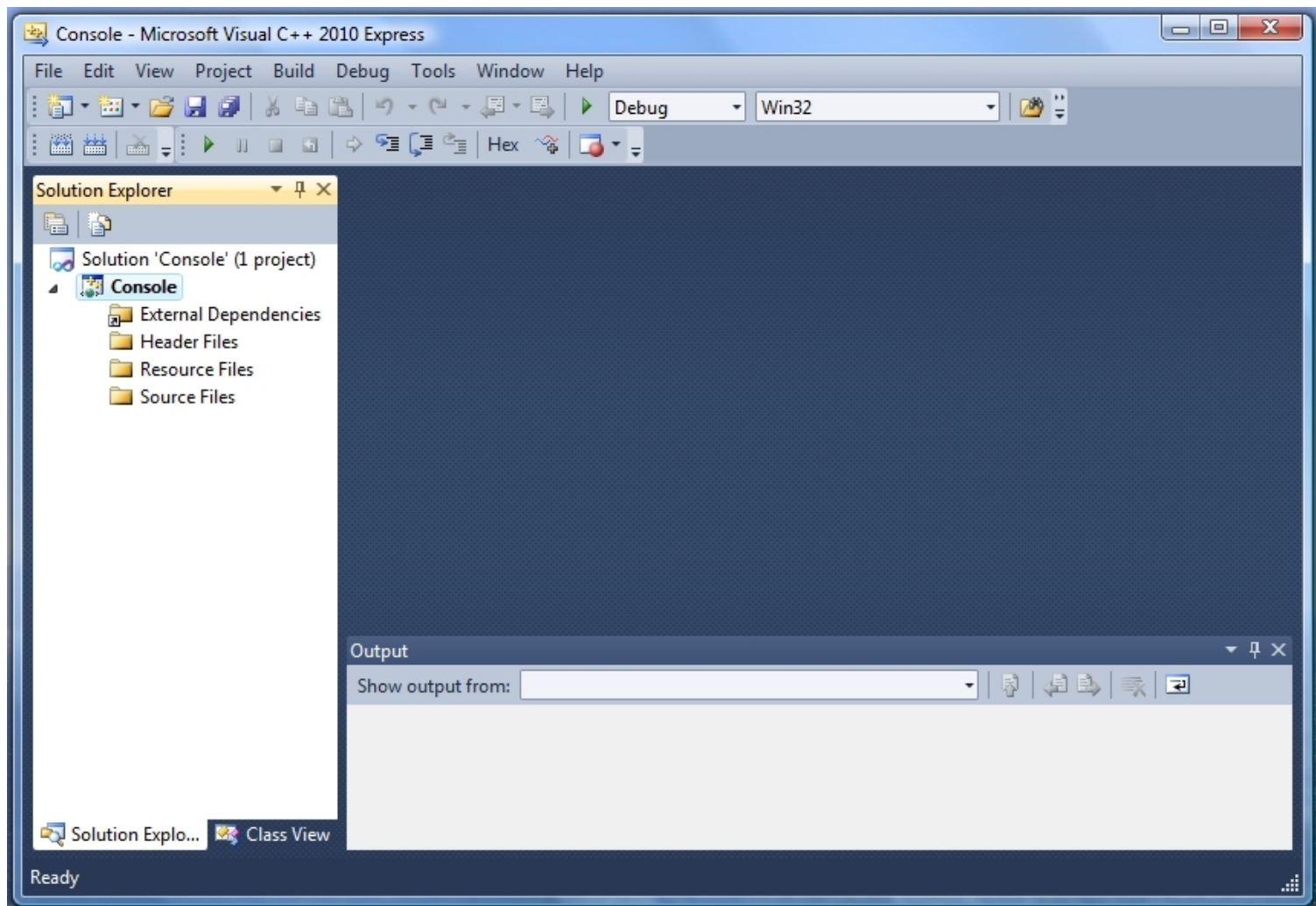
Pour notre exemple, nous choisissons une application Console native : *Win32 Console Application*. Saisissez un nom pour le projet et renseignez le nom du répertoire où enregistrer le projet si celui proposé par Visual ne vous convient pas. Ensuite, appuyez sur le bouton OK. L'assistant passe alors à la page suivante :



Cette page rappelle le type de projet choisi. Appuyez sur le bouton suivant ('Next') :



Persistons dans notre choix d'une application console (bouton radio 'Console Application'. Décochons l'option des en-têtes précompilés et choisissons la création d'un projet vide. Nous appuyons alors sur le bouton 'Terminer' (finish). Visual crée le projet et ouvre l'environnement de développement :



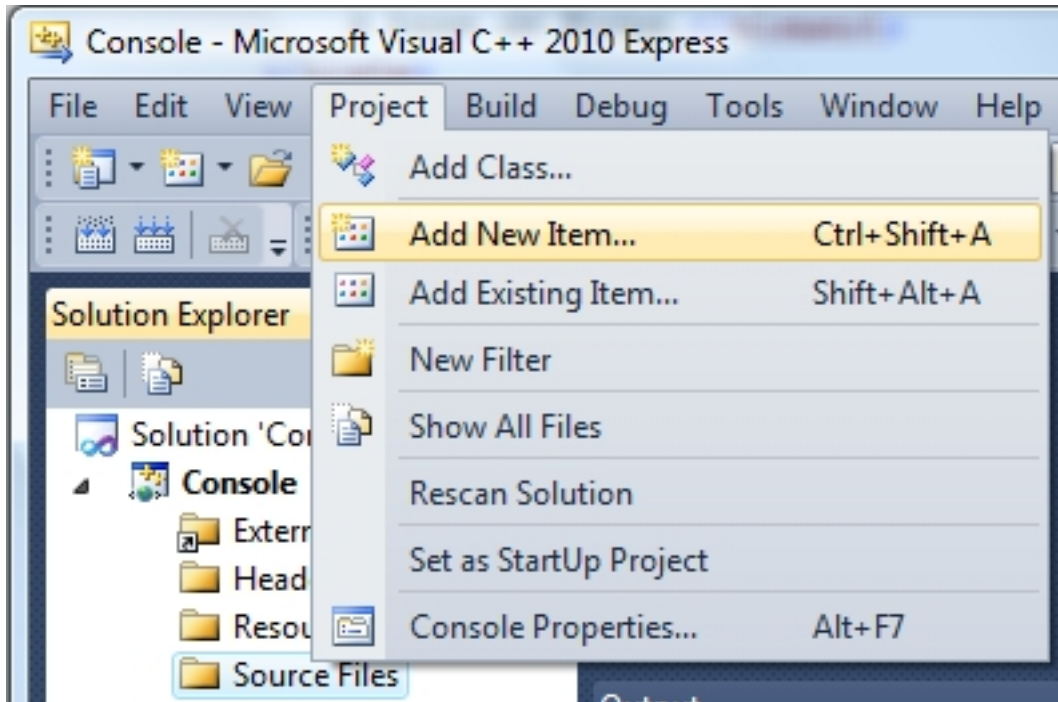
V-B - La solution

À gauche, l'explorateur de solution propose quatre filtres pour regrouper les fichiers :

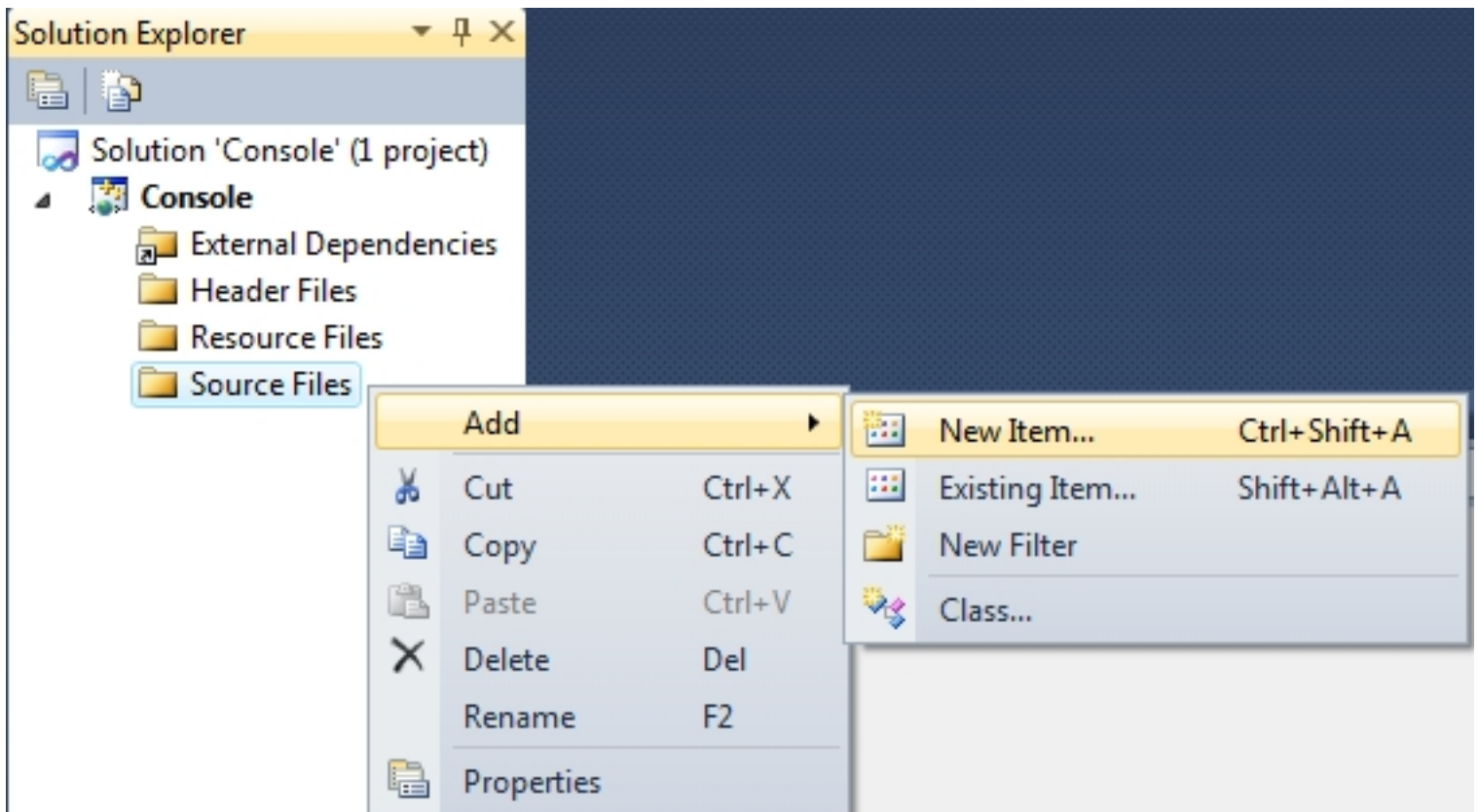
- External Dependencies : dépendances externes du projet ;
- Header Files : fichiers d'en-têtes (.h) ;
- Resource Files : fichiers ressources (.rc) ;
- Source Files : fichier sources (.cpp).

V-C - Créer un nouveau fichier

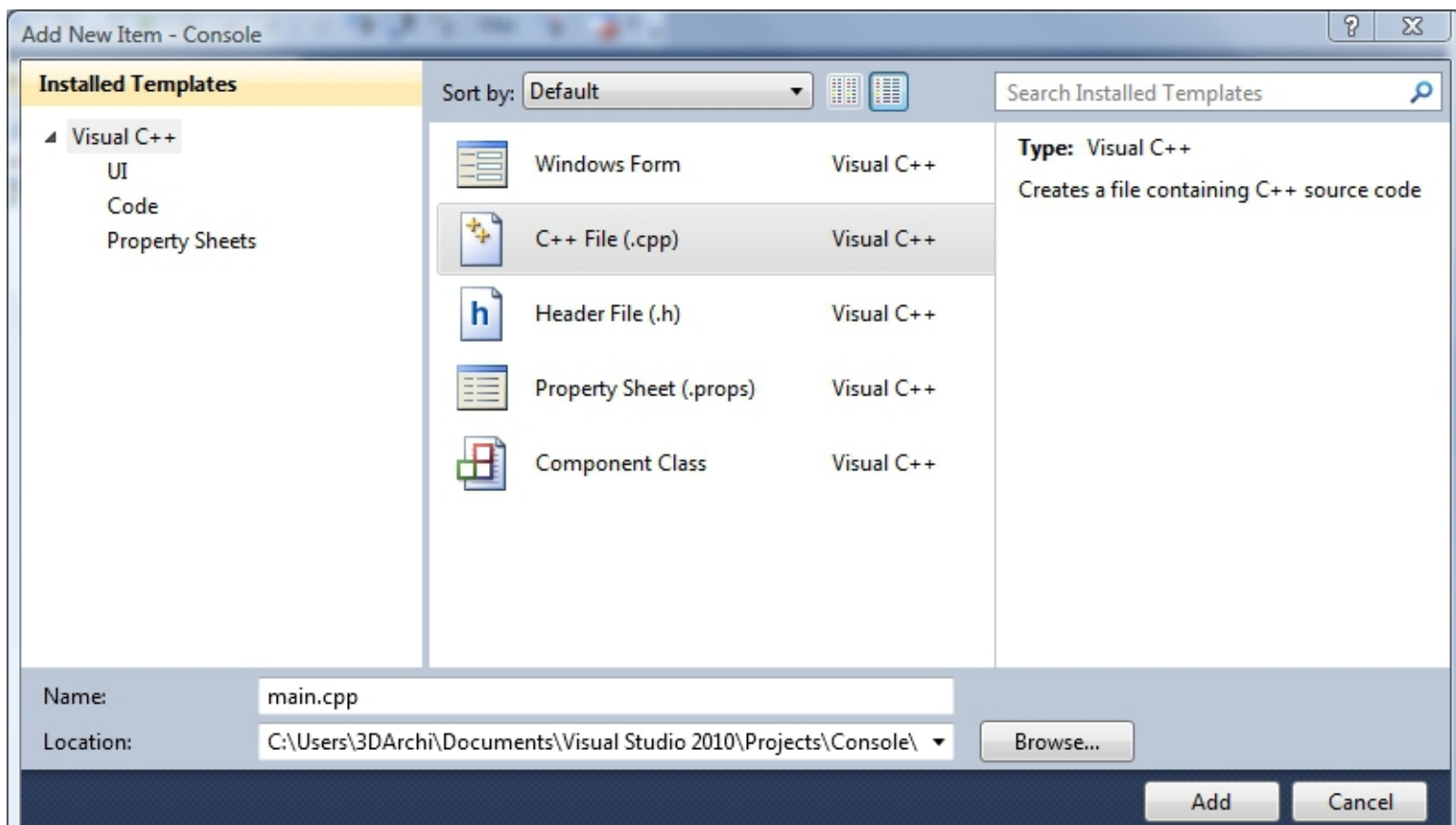
Notre projet ne contient pour l'instant aucun fichier et donc ni fonction ni classe. Commençons donc par créer un fichier principal qui sera amené à contenir la fonction d'entrée de notre application. Dans le menu 'Project' de la barre de menu principal, il faut choisir l'entrée 'Add new item...' :



Notez que ce menu est associé au raccourci clavier "Shift+Ctrl+A". Il est aussi disponible dans le menu contextuel du projet en se positionnant sur le filtre dans lequel le fichier est ajouté :



Un assistant s'affiche demandant des compléments d'information sur l'élément à ajouter :



Plusieurs catégories sont proposées pour ce nouveau fichier. Choisissez Fichier C++ (C++ File) pour créer un fichier source contenant la définition de notre fonction principale. Il faut aussi renseigner le nom du nouveau fichier et son emplacement. Je propose *main.cpp*. Si le nom du fichier n'a pas d'importance d'un point de vue C++, *main.cpp* est suffisamment explicite pour que n'importe quel développeur aguerri s'attende à y trouver le point d'entrée du projet.

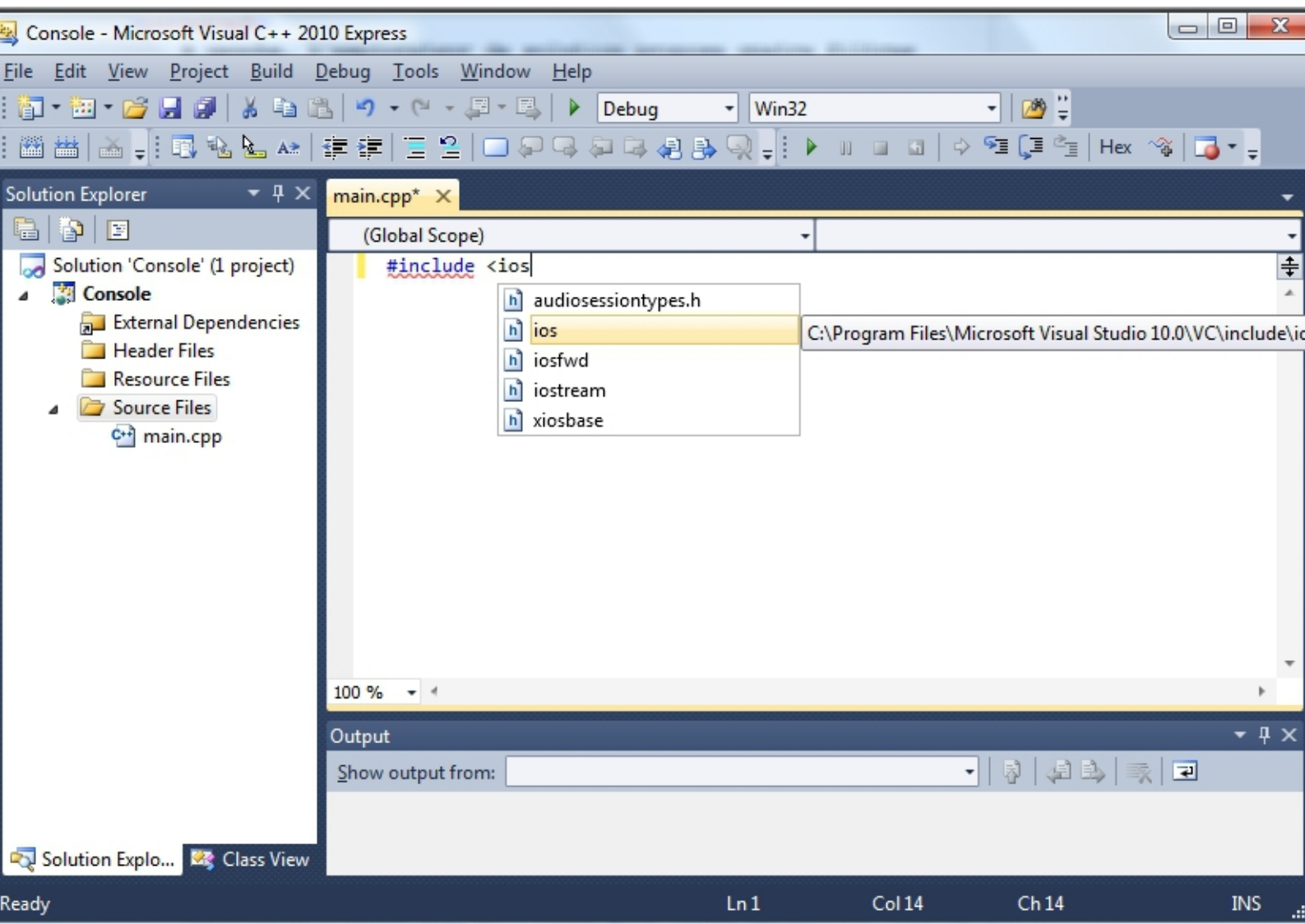
V-D - Ajout d'une fonction principale

Dans ce nouveau fichier créé, rajoutons le code ci-dessous :

```
main.cpp
#include <iostream>

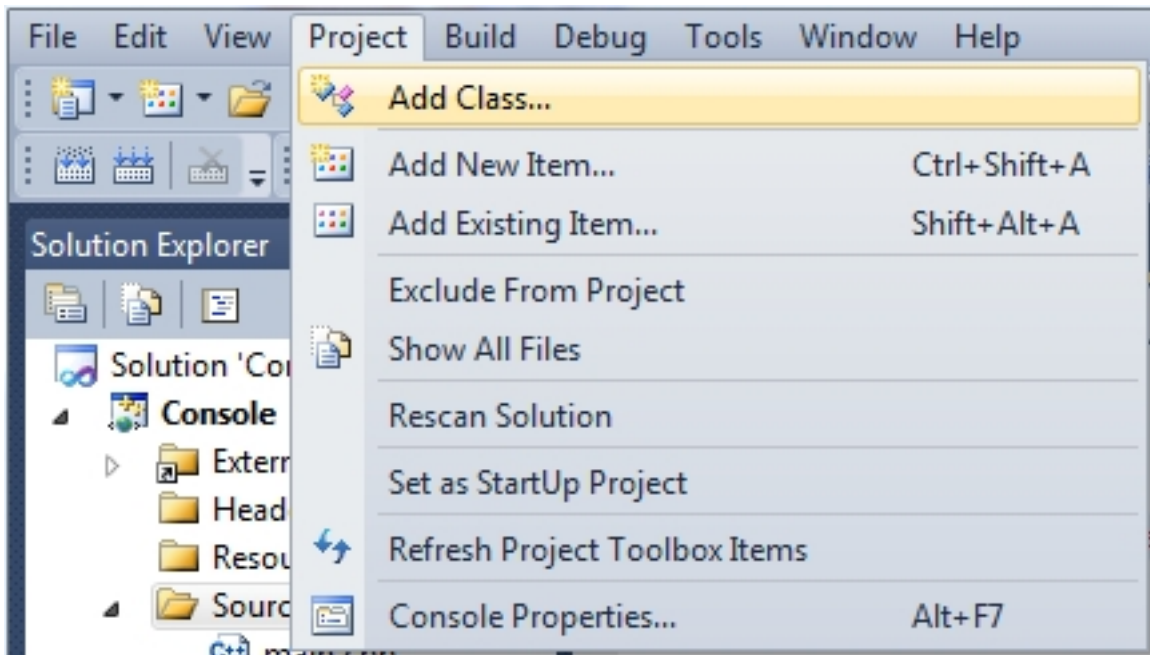
int main()
{
    std::cout<<"Mon application console\n";
    return 0;
}
```

Notez l'IntelliSense (l'autocomplétion de Visual C++) qui vient vous aider dans la saisie du code :

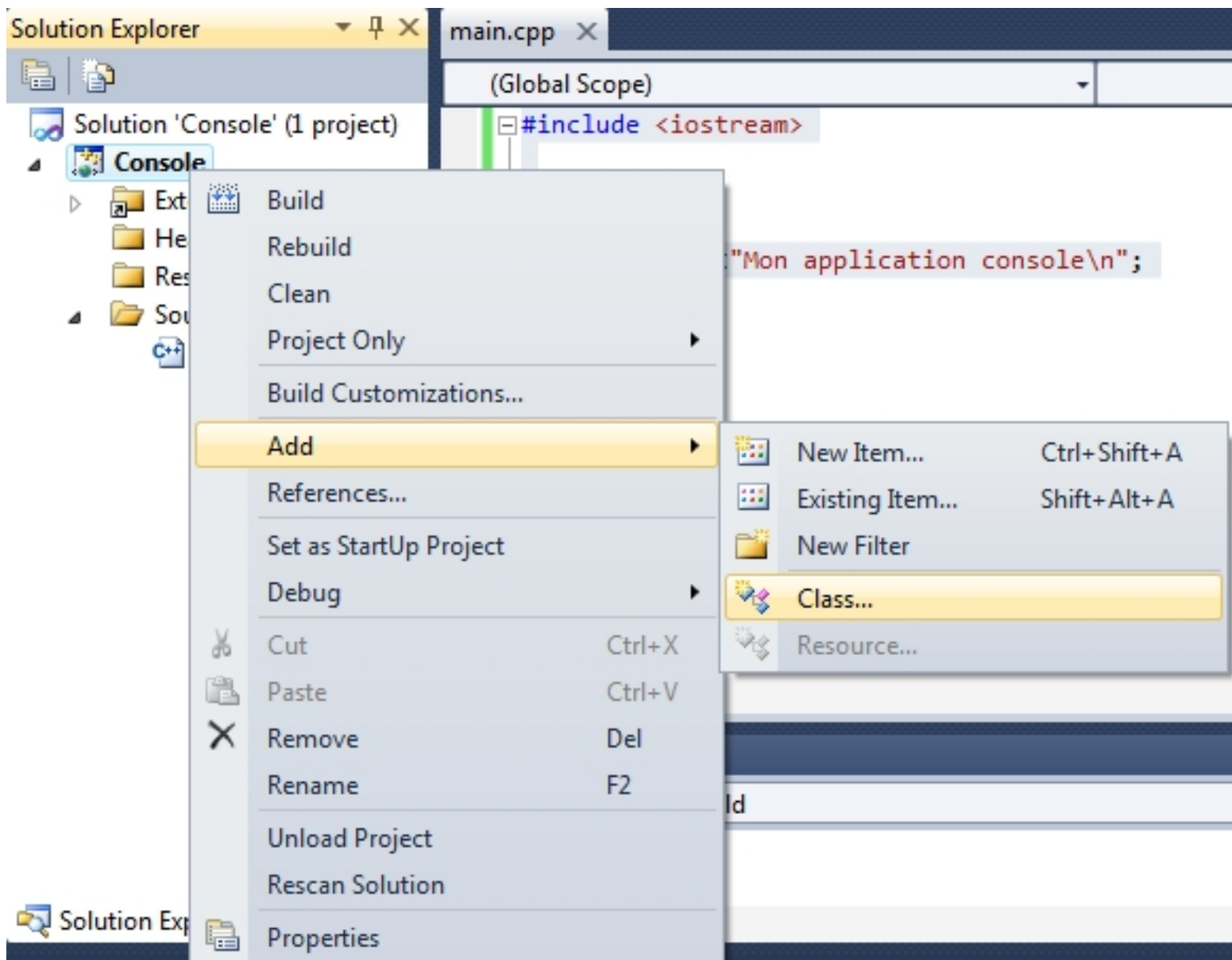


V-E - Ajouter une nouvelle classe

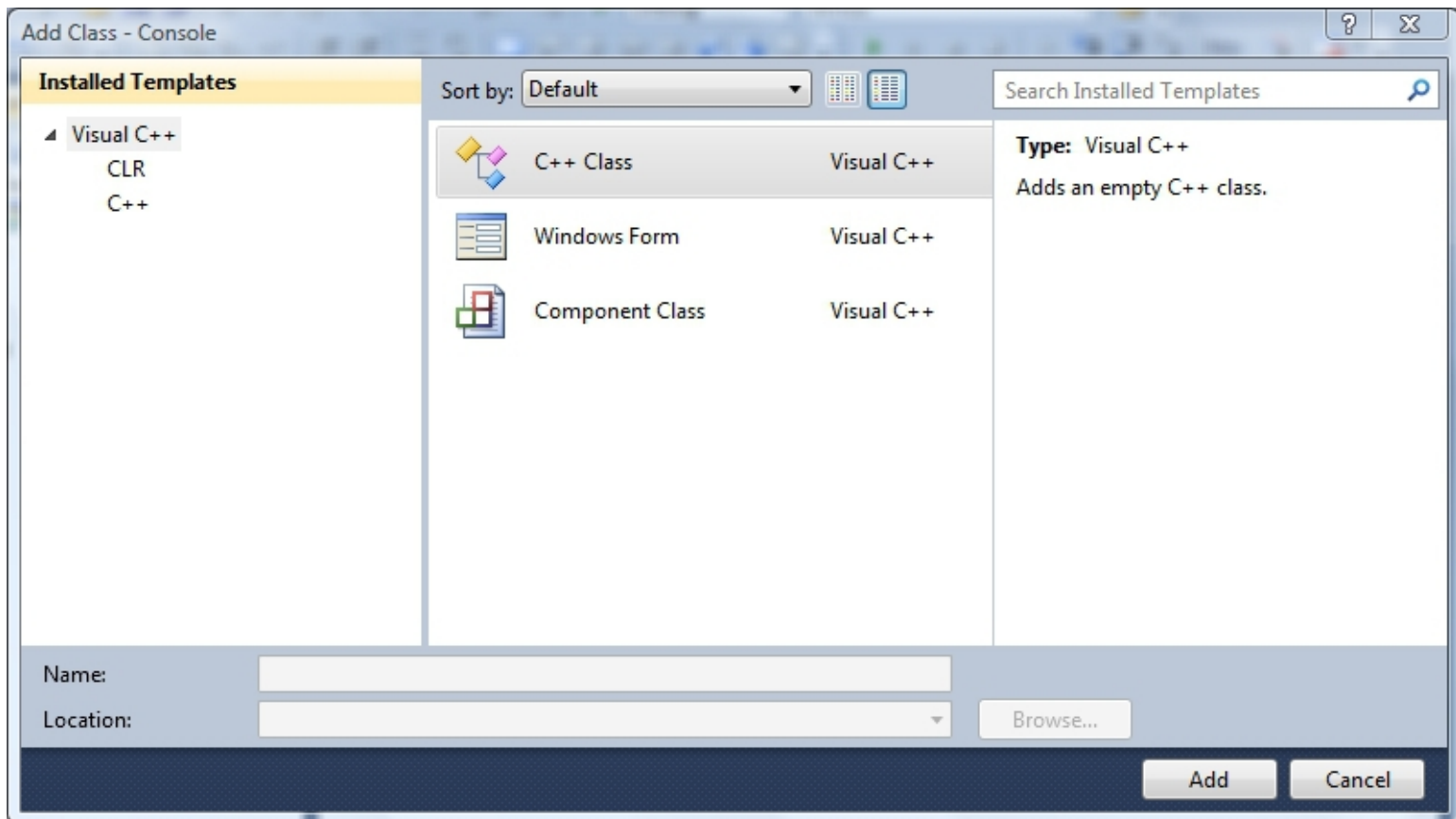
Pour parfaire notre premier projet, ajoutons une classe. Comme pour le projet, cela peut se faire via le menu 'Project' de la barre de menu principale :



Ou par le menu contextuel du projet dans le volet 'Solution' :



La première page de l'assistant s'ouvre pour nous demander le type de classe que nous souhaitons créer :



Optez pour le premier choix C++ Class : une classe C++... L'assistant ouvre la page suivante permettant la saisie des différentes informations de notre classe :

Generic C++ Class Wizard - Console

Welcome to the Generic C++ Class Wizard

Class name: MyApplication

.h file: MyApplication.h

.cpp file: MyApplication.cpp

Base class:

Access: public

Virtual destructor

Inline

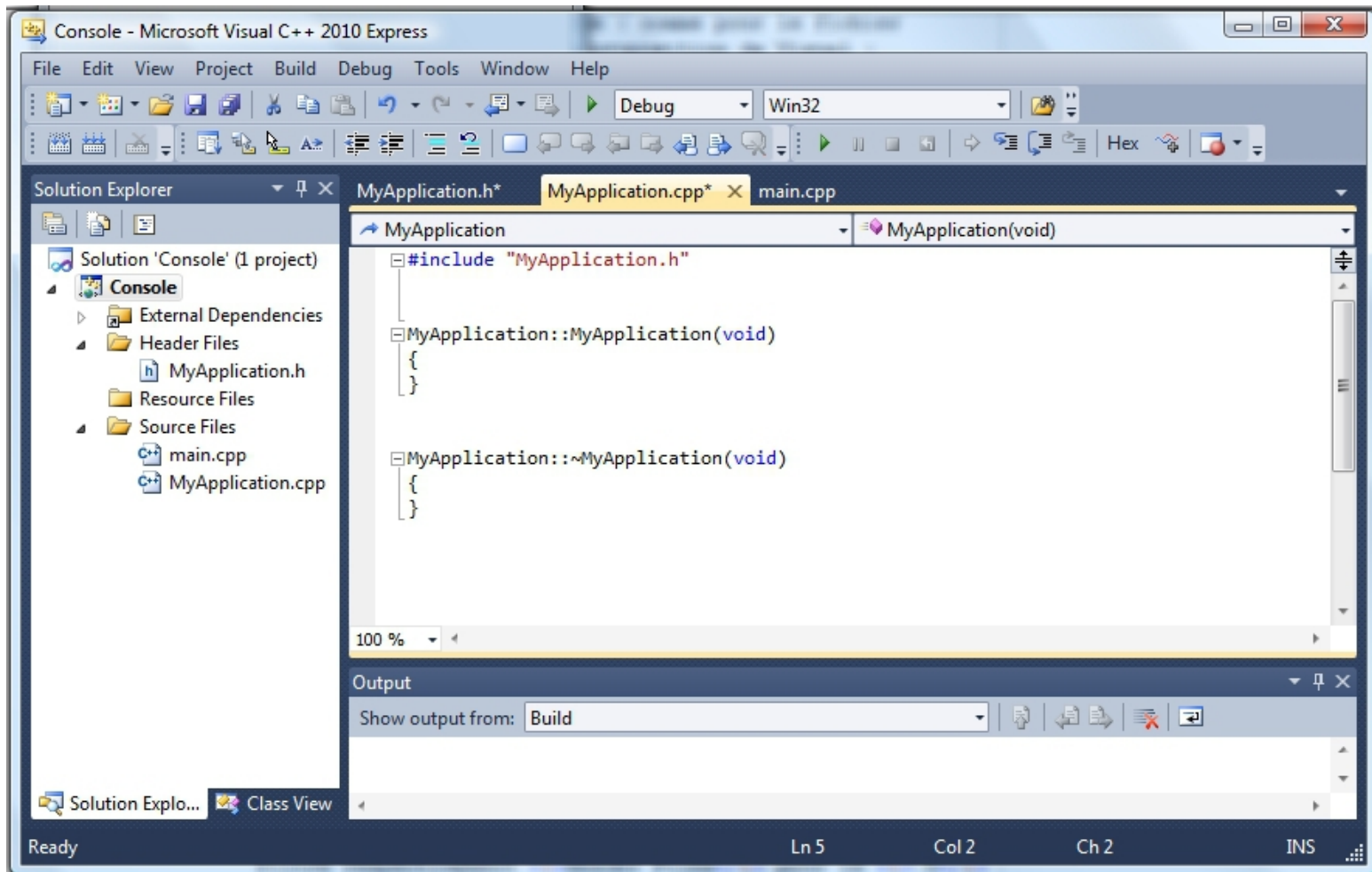
Managed

Finish Cancel

Cet assistant permet de saisir :

- le nom de la classe : choisissons *MyApplication* ;
- le nom du fichier d'en-tête : Visual propose automatiquement un nom de fichier avec le nom de la classe *MyApplication.h*. Celui-ci nous convient donc laissons-le en l'état ;
- le nom du fichier source : comme pour le fichier d'en-tête, laissons la proposition de Visual : *MyApplication.cpp* ;
- la classe de base : nous n'avons pas de classe de base, donc laissons la zone vide ;
- la portée de l'héritage : comme nous n'avons pas d'héritage, cette zone de saisie n'a pas d'importance ;
- destructeur virtuel : si la classe créée sert de base pour un héritage public, alors nous aurions certainement coché cette case. Dans notre cas, cela ne s'applique pas ;
- inline : en cochant cette case, l'ensemble des fonctions de la classe sont définies *inline* et le fichier source *.cpp* n'est pas créé. Dans notre cas, cela ne nous intéresse pas donc laissons la case décochée.

Il est temps de valider cette création en appuyant sur le bouton Finir ('Finish').
 Visual C++ Express crée automatiquement les fichiers MyApplication.h et MyApplication.cpp, renseigne ces fichiers avec la définition de la classe, un constructeur par défaut et un destructeur et les ajoute dans le bon filtre respectivement *Header Files* pour le .h et *Source Files* pour le .cpp. Les deux fichiers sont automatiquement ouverts :



Modifions le fichier d'en-tête comme suit :

```
MyApplication.h
#ifndef MyApplication_h_
#define MyApplication_h_

namespace my_project
{
    class MyApplication
    {
    public:
        MyApplication();
        ~MyApplication();
        int run();
    };
}

#endif /* MyApplication_h_ */
```

Le fichier source définit nos fonctions :

```
MyApplication.cpp
#include "MyApplication.h"
#include <iostream>
```

MyApplication.cpp

```

namespace my_project{
MyApplication::MyApplication()
{
    std::cout<<"Bonjour\n";
}

MyApplication::~MyApplication()
{
    std::cout<<"A bientôt\n";
}

int MyApplication::run()
{
    std::cout<<"Ceci est notre première application\n";
    return 0;
}
}
    
```

Enfin modifions notre fichier main.cpp pour utiliser notre nouvelle classe :

main.cpp

```

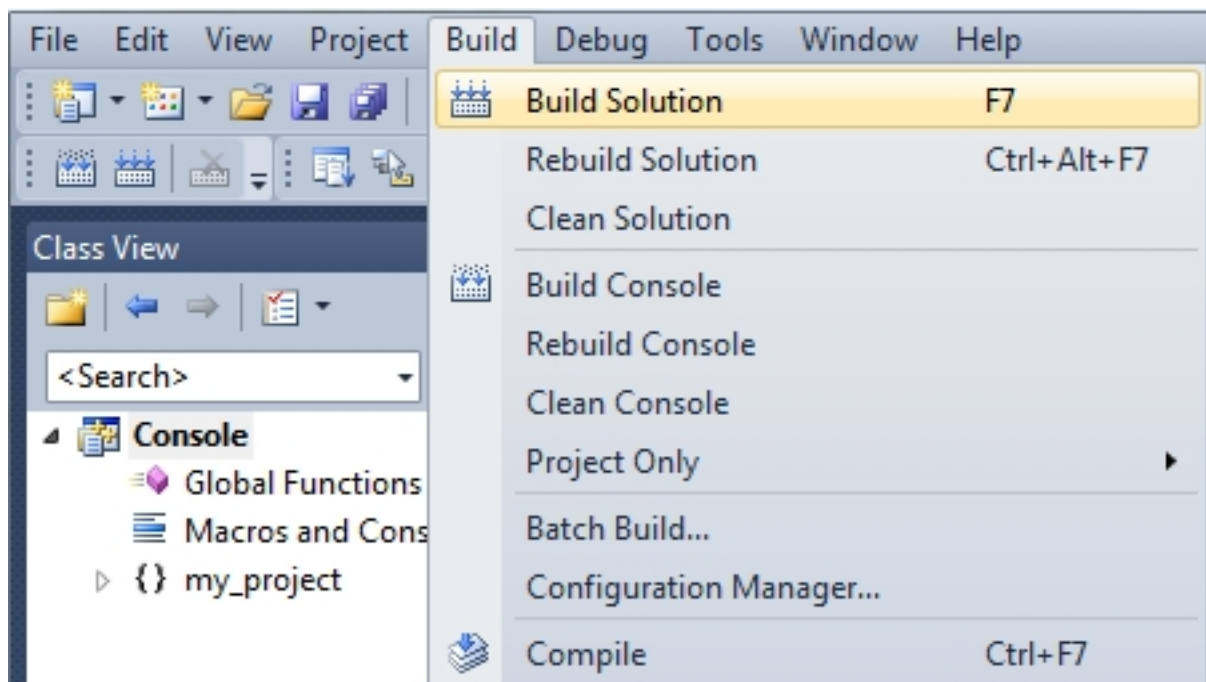
#include "MyApplication.h"

int main()
{
    my_project::MyApplication application_principale;
    int resultat = application_principale.run();
    return resultat;
}
    
```

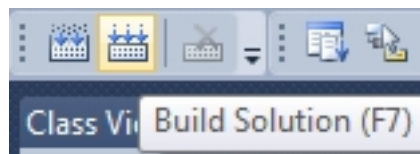
V-F - Compiler un projet

Il est temps de compiler notre projet. Plusieurs possibilités s'offrent à nous pour obtenir notre exécutable :

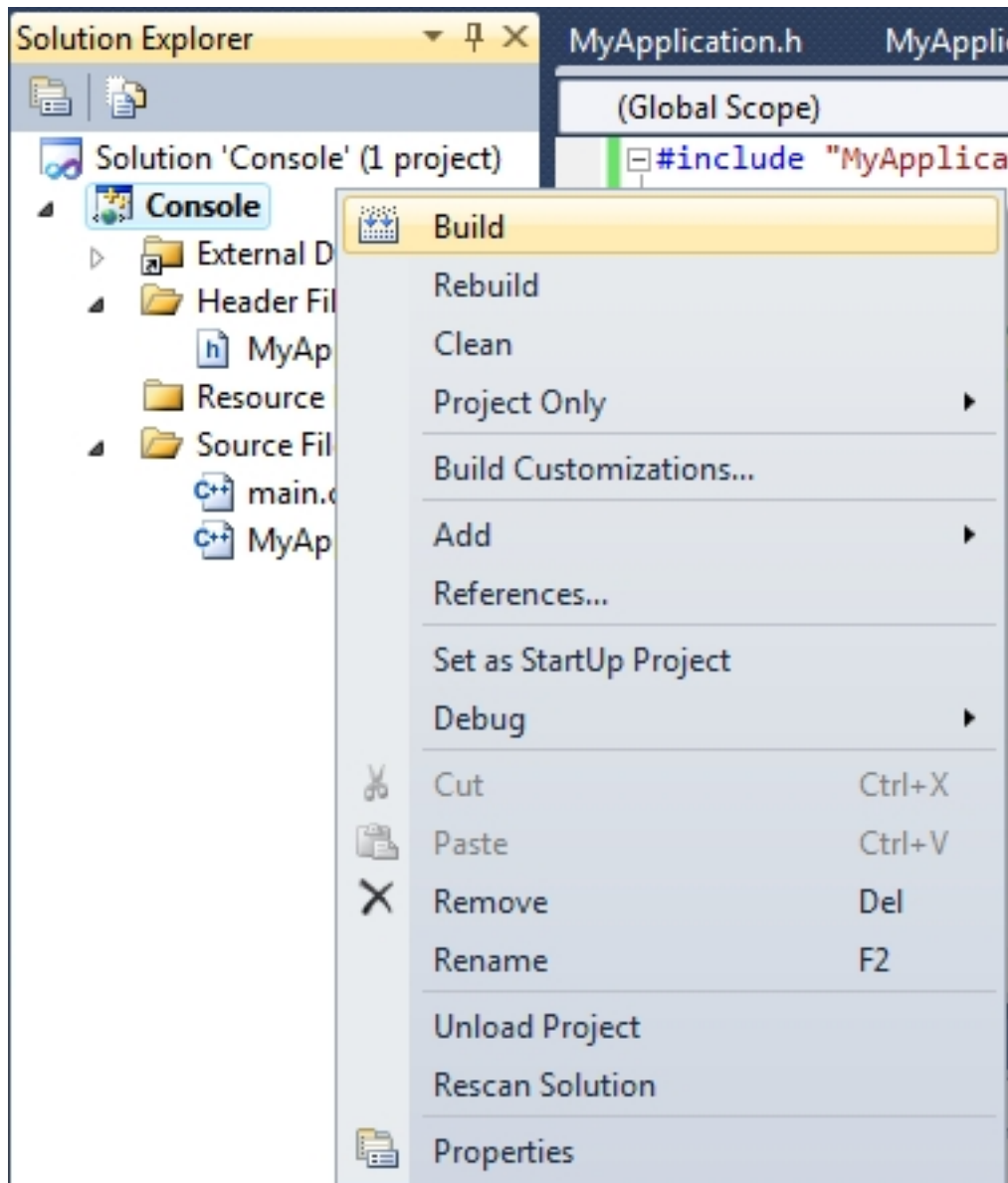
- > par raccourci clavier en appuyant sur "F7" ;
- > ou encore avec le premier item du menu 'Build' :



-> ou la barre des boutons :



-> ou le menu contextuel du projet :



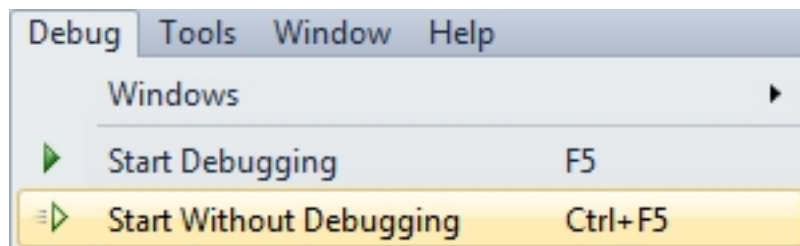
La fenêtre de sortie en bas de l'environnement de développement présente le résultat de la compilation :

```

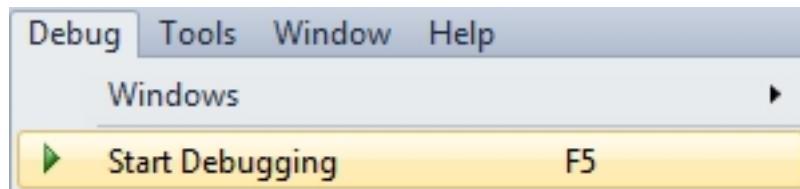
output from: Build
----- Build started: Project: Console, Configuration: Debug Win32 -----
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
  
```

V-G - Exécuter ou débogger un programme

L'exécution du programme se fait en utilisant le raccourci clavier "Ctrl+F5" ou en utilisant le menu 'Debug' et en choisissant 'Start Without Debugging' :



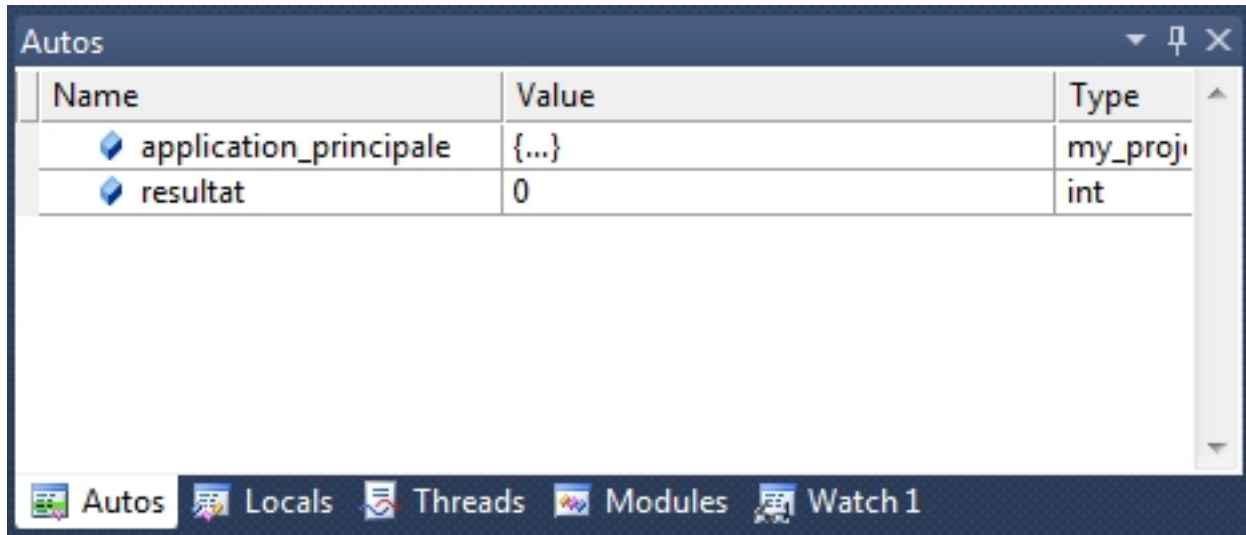
Pour débogger, le raccourci clavier est "F5". Ou avec le menu 'Debug' qui contient l'élément 'Start Debugging' pour lancer le programme en mode debug :



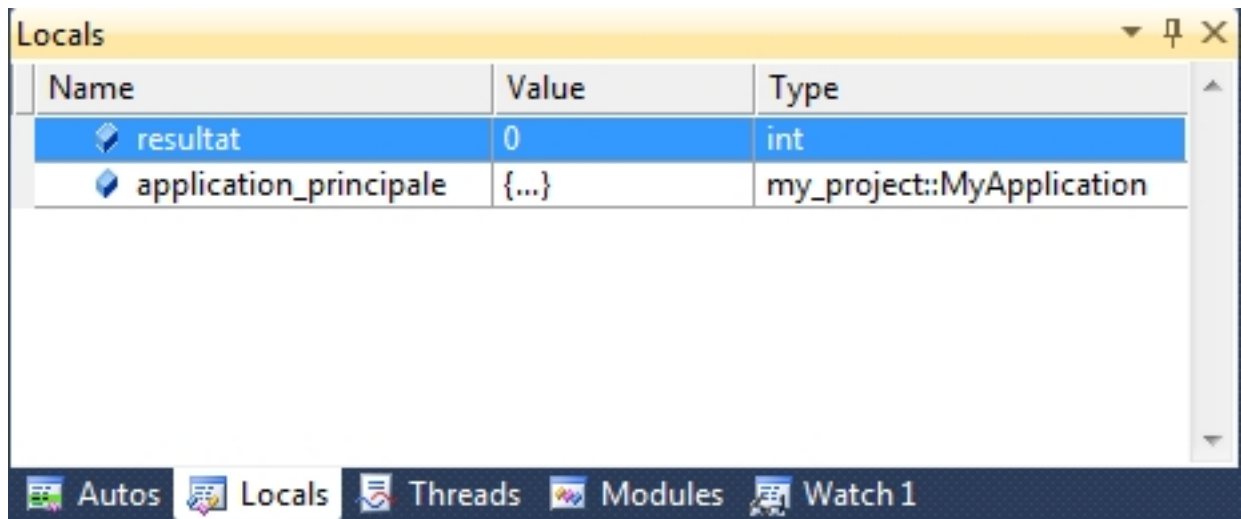
Les raccourcis clavier utiles au debuggage sont :

- F9 : ajout ou suppression d'un point d'arrêt ;
- Ctrl + Shift + F9 : supprimer tous les points d'arrêt ;
- F5 : continuer l'exécution ;
- Shift + F5 : arrêter le debuggage ;
- Ctrl + Shift + F5 : recommencer ;
- F11 : entrer dans la fonction ;
- F10 : exécuter l'instruction suivante sans entrer dans la fonction ;
- Shift + F11 : sortir de la fonction courante ;
- Shift + F9 : *QuickWatch* affichage de la valeur de la variable sous le curseur.

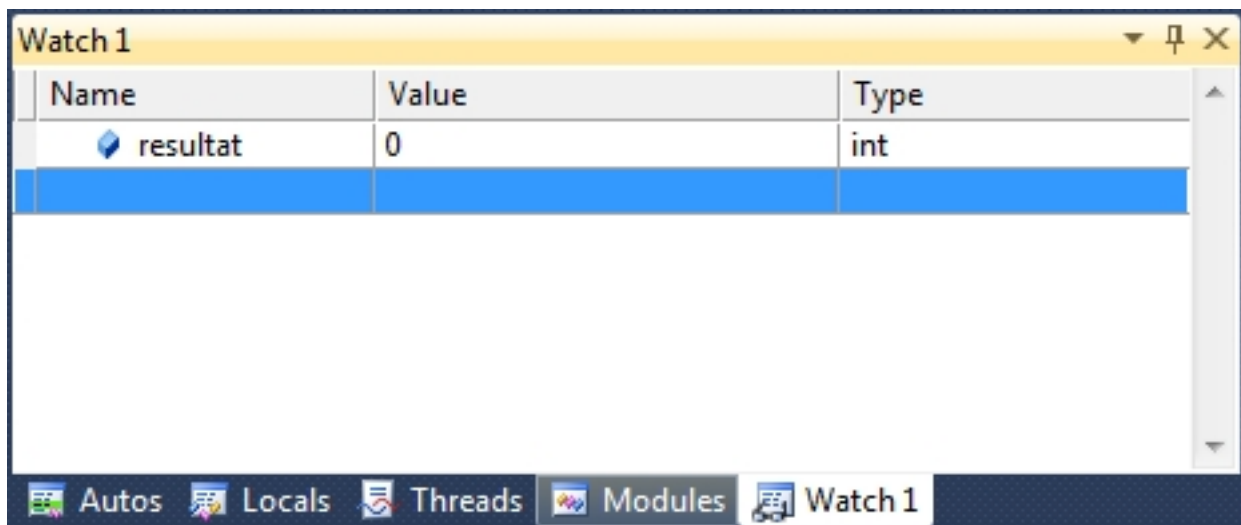
Plusieurs fenêtres sont proposées en bas de l'environnement pour vous accompagner dans la recherche d'anomalies. La fenêtre *auto* affiche les éléments que Visual juge pertinents en fonction du contexte (variables locales, retour de fonction, etc.) :



'Locals' affiche les variables locales de la fonction en train d'être exécutée :



'Watch' permet de saisir des variables particulières que l'on souhaite surveiller :



V-H - Directive de compilation Visual C++ 2010

`_MSC_VER` est la directive prédéfinie par Microsoft indiquant la version du compilateur Visual utilisée. Pour visual C++ Express 2010, cette constante vaut 1600 :

```
#include <iostream>

int main()
{
    #if _MSC_VER == 1600
        std::cout<<"Ceci est un programme compile avec visual C++ 2010\n";
    #else
        std::cout<<"Ceci est un programme non compile avec visual C++ 2010\n";
    #endif
    return 0;
}
```

Pour rappel, les valeurs sont les suivantes selon les différentes versions de Visual C++ :

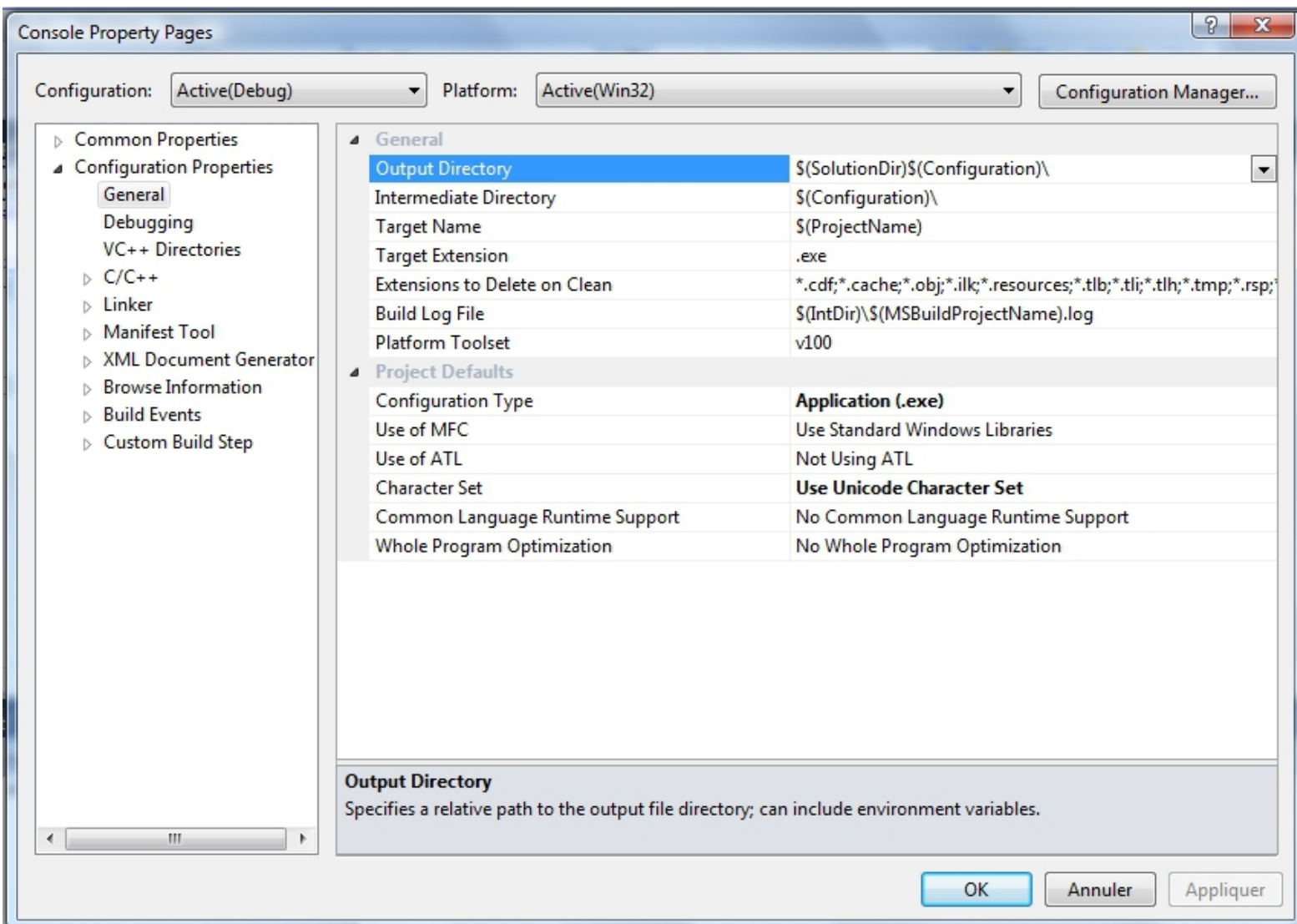
- 1200 : MSVC 6.0 ;
- 1300 : MSVC 7.0 Visual C++ .NET 2002 ;
- 1310 : Visual C++ .NET 2003 ;
- 1400 : Visual C++ 2005 ;
- 1500 : Visual C++ 2008 ;
- 1600 : Visual C++ 2010.

VI - Configurer un projet C++

Comme pour la plupart des actions possibles, la configuration d'un projet est accessible par un raccourci clavier (Alt +F7), par la barre de menu principal ('Project', puis en bas 'Properties') ou via le menu contextuel en se positionnant sur le projet ('Properties').

VI-A - Configuration générale

L'onglet de configuration général donne accès aux paramètres globaux de l'application :



VI-A-1 - Sortie de la compilation


Les trois premières entrées permettent de spécifier différents éléments issus de la compilation :

- Output Directory : précise le répertoire où est généré le résultat de la compilation (exe, dll, lib, etc.) ;
- Intermediate Directory : répertoire où sont écrits les fichiers intermédiaires de la compilation (.obj, .log, etc.) ;
- Target Name : nom du fichier produit par la compilation, c'est à dire nom de l'exécutable ou de la bibliothèque ;
- Target Extension : extension du fichier précédent (.exe pour une application par exemple) ;
- Extension To Delete On Clean : liste des extensions des fichiers à supprimer lors d'un 'Rebuild All' ou d'un 'Clean'.

Pour des projets importants, il devient rapidement capital d'avoir un répertoire intermédiaire différent de celui où est généré l'exécutable ou la bibliothèque. Les deux premières entrées permettent de faire cette séparation. Par défaut, le nom de l'exécutable ou de la bibliothèque est lié au nom du projet créé. La troisième entrée sert à renseigner un nom différent pour par exemple distinguer les noms selon différentes configurations.

VI-A-2 - Compiler en C++0x ou en C++03

L'entrée 'Platform Toolset' permet de choisir le compilateur utilisé. La liste propose deux choix : *v100* et *v90*. *v100* utilise le compilateur installé avec Visual C++ Express 2010. Ce compilateur utilise les nouveautés de la future norme

du C++. Si vous devez produire du code compilable sans ces nouveautés alors vous devez préciser l'entrée *v90*. En choisissant dans 'Platform Toolset' l'entrée *v90* vous demandez à Visual C++ Express 2010 d'utiliser le compilateur de la version précédente. Ce compilateur n'est pas intégré à cette installation. Vous devez donc l'avoir installé par ailleurs. Vous pouvez consulter ce tutoriel pour vous aider à l'installer :  **Premiers pas avec Visual C++ Express Edition 2008.**

VI-A-3 - Exécutable ou bibliothèque ?

La ligne 'Configuration Type' informe sur le type de projet : application (.exe), bibliothèque statique (.lib) ou bibliothèque dynamique (.dll).

VI-A-4 - Utilisation des MFC et de l'ATL

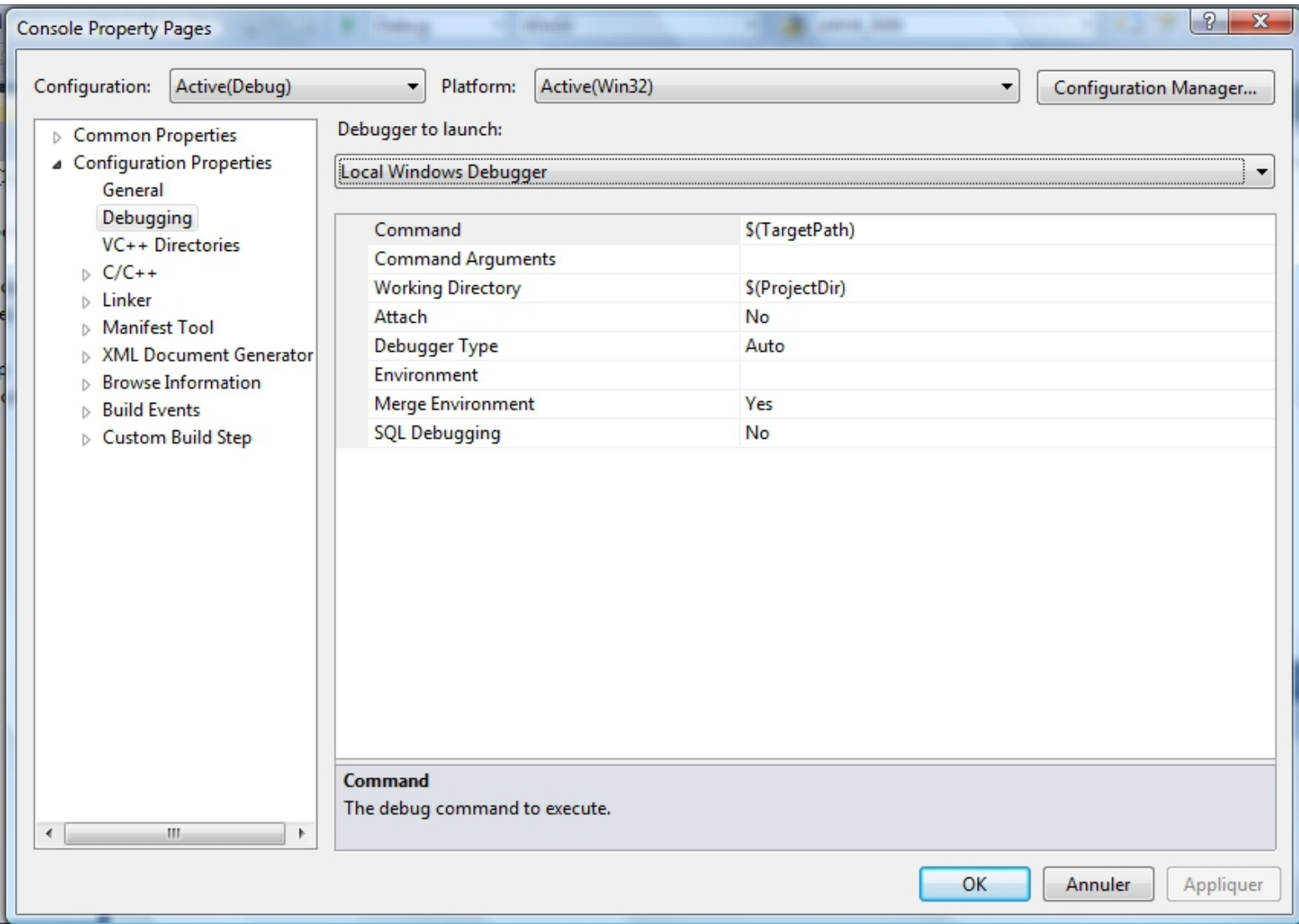
La version Express de Visual C+ **n'inclut pas** les MFCs ni les classes issues de l'ATL. Il ne sert à rien de modifier ces entrées des valeurs par défaut issues de la création du projet.

VI-A-5 - Unicode ou pas ?

Détermine si les chaînes de caractères passées à l'interface Win32 sont Unicode (*Use Unicode Character Set*) ou ASCII (*Not Set*).

VI-B - Configuration des options de Debug

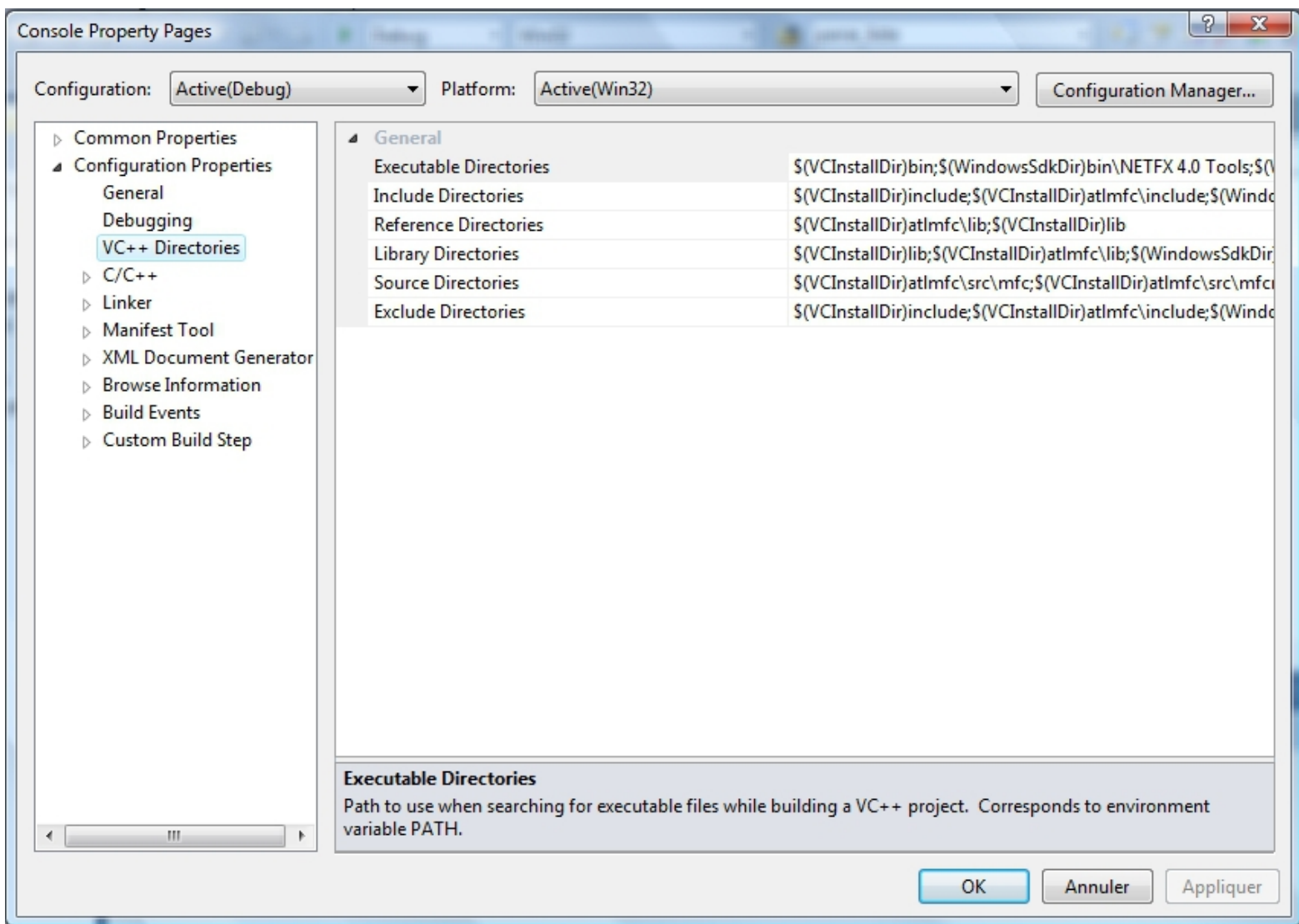
Debugging ouvre l'onglet de configuration en mode debug :



Dans la zone d'édition, sous le label *Command Arguments*, vous pouvez saisir les arguments passés par le debugger à l'application. Vous les retrouvez dans votre programme dans les paramètres `argc` et `argv` de la fonction d'entrée du programme : `int main(int argc, char *argv[])`.

VI-C - Configurer les répertoires du projet

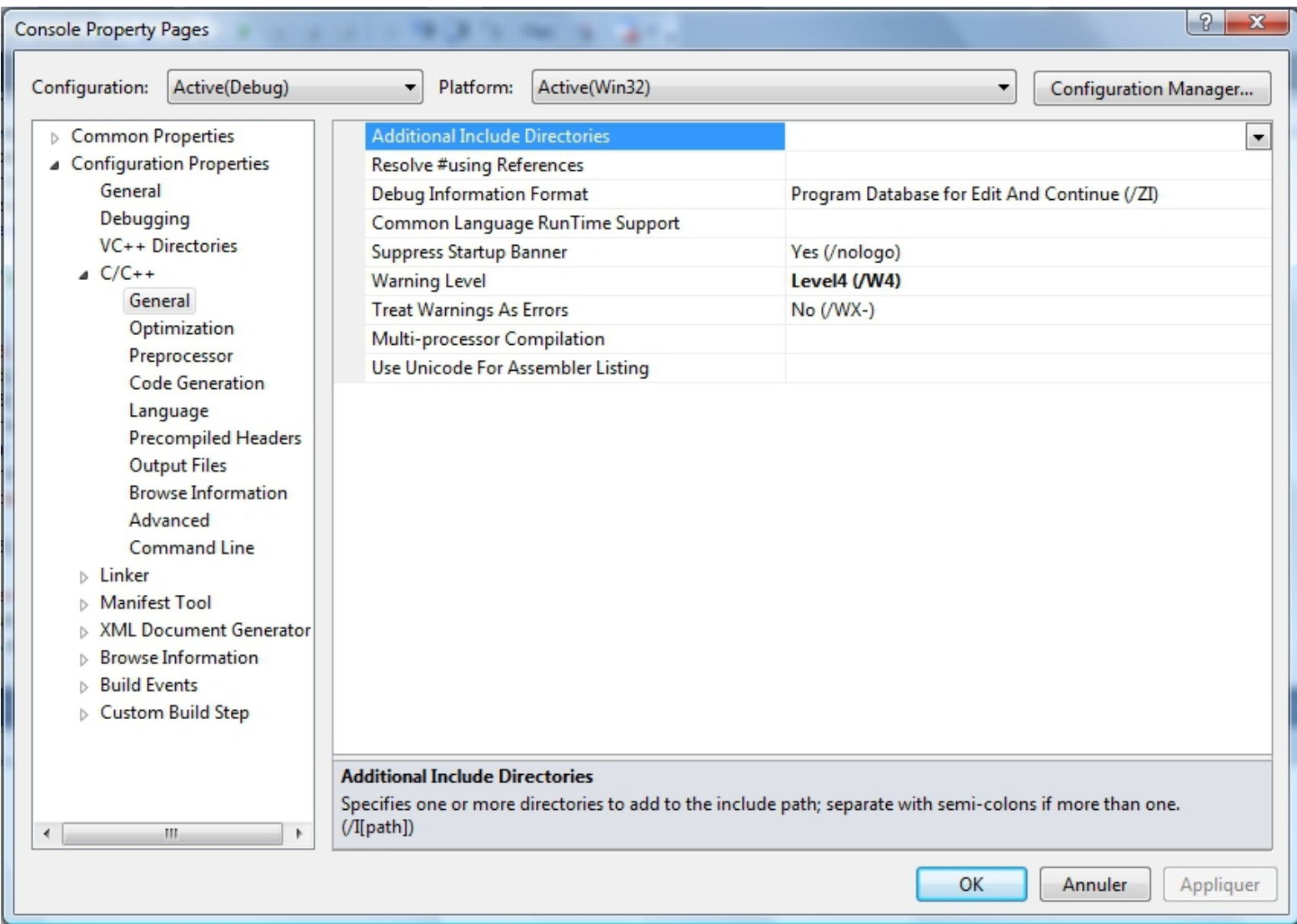
Avec Visual C++ Express 2010, tout comme dans les versions payantes, les répertoires de recherche des fichiers d'en-têtes et des fichiers .lib pour les bibliothèques externes ne peuvent plus se configurer au niveau global, mais seulement par projet :



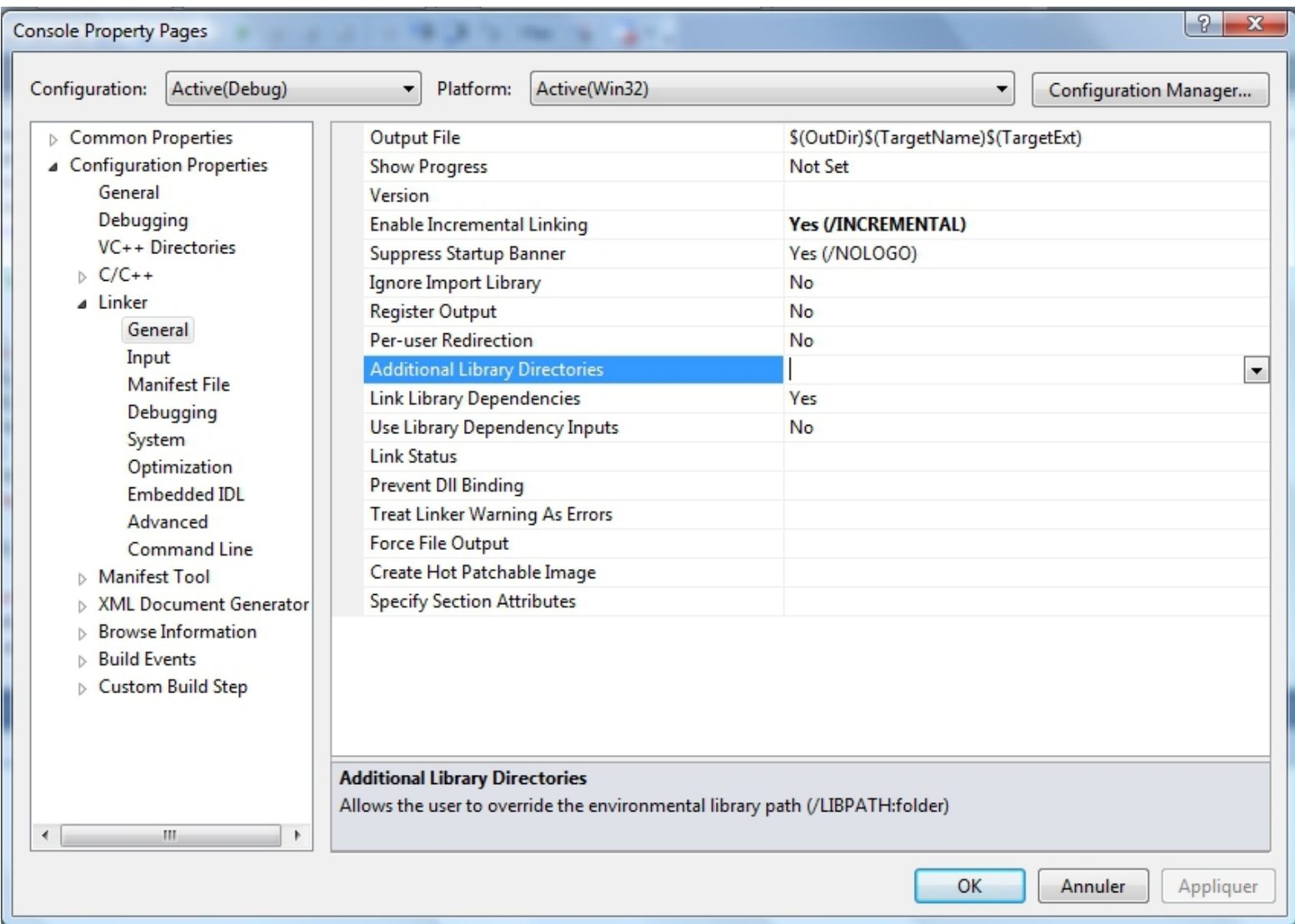
Include Directories sert à définir les répertoires de recherche des fichiers .h.

Library Directories et *Reference Directories* contiennent les répertoires où le linker cherche les .lib.

Les répertoires contenant les fichiers d'en-tête peuvent aussi se spécifier dans les options C++ du projet :



De la même façon, les options du linker permettent d'ajouter des répertoires pour rechercher les fichiers .lib :



Se pose alors la question de savoir où mettre les répertoires dont a besoin un projet : au niveau global ou au niveau de l'outil (compilateur ou éditeur de lien) ? Pour cela faisons un détour sur la façon dont Visual C++ Express effectue sa recherche.

VI-C-1 - Dans quels répertoires Visual C++ Express cherche-t-il les fichiers d'en-tête ?

Les fichiers d'en-tête (.h) sont recherchés dans l'ordre suivant :

- 1 le répertoire du fichier en cours de compilation ;
- 2 les répertoires précisés en ligne de commande (/I) ;
- 3 les répertoires contenus dans la variable d'environnement INCLUDE.

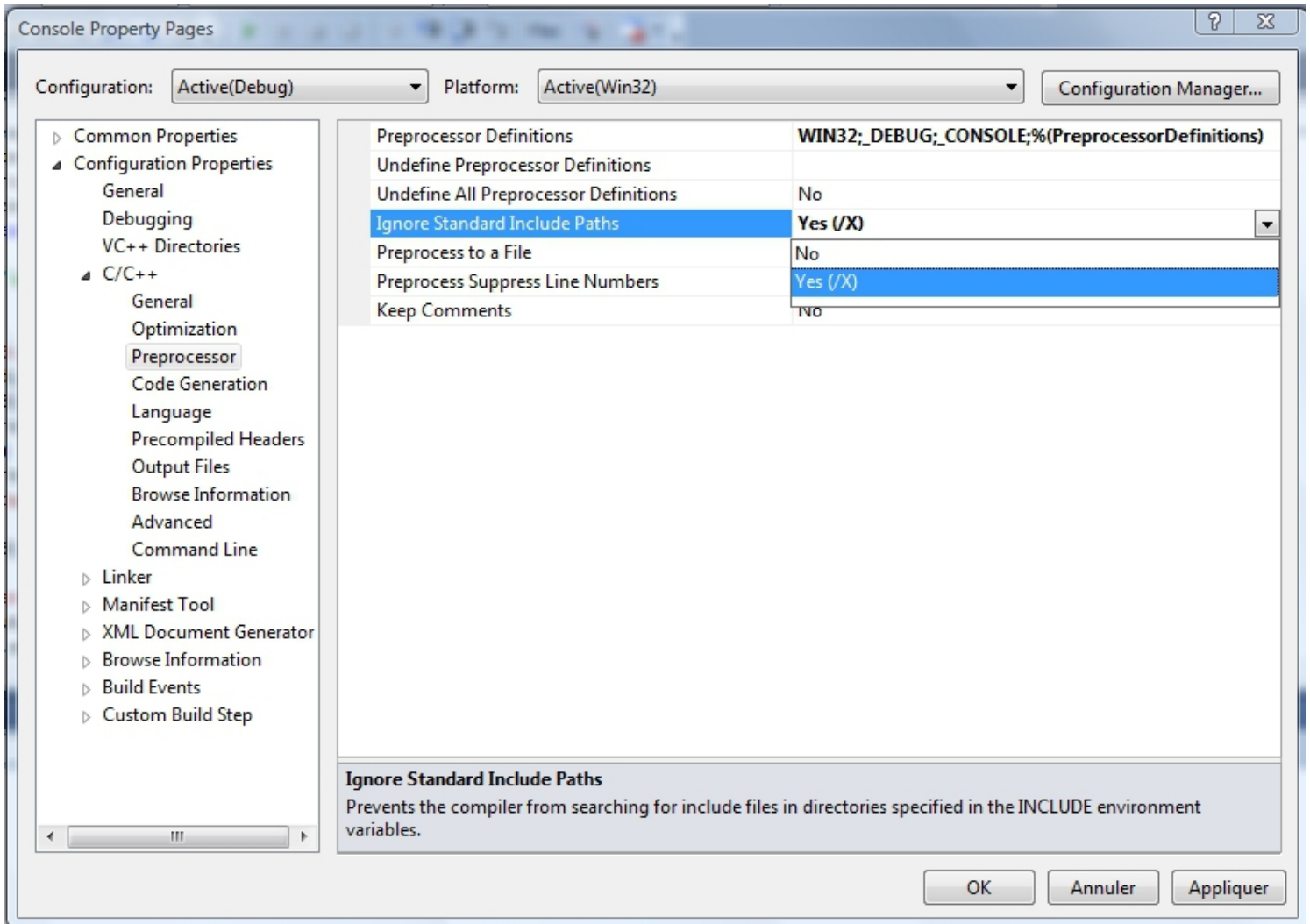
Les répertoires précisés en ligne de commande (/I) correspondent à ceux indiqués dans les options C++ : *Additional Include Directories* des options générales de C/C++.

Les répertoires définis par la variable d'environnement INCLUDE sont ceux définis au niveau de la configuration générale : *Include directories* des options VC++ Directories.

Savoir où mettre les répertoires de recherche peut être plus crucial dans les solutions comportant plusieurs projets. Afin de prendre des habitudes convenant pour tout type de projet, je conseille la démarche suivante :

- dans *Include directories* des options *VC++ Directories* (variable d'environnement INCLUDE) : les bibliothèques générales ayant vocation à être utilisées dans tous les composants du projet (par exemple Boost) ;
- dans *Additional Include Directories* des options générales de C/C++ : les répertoires propres au projet lorsque ses sources sont réparties dans plusieurs répertoires ainsi que les bibliothèques ayant une portée très locale (par exemple tinyXML pour uniquement un composant qui utilise cette bibliothèque).

L'option de compilation *Ignore Standard Include Paths* désactive la recherche des répertoires précisés dans la variable d'environnement INCLUDE, c'est-à-dire les répertoires précisés dans *Include directories* des options *VC++ Directories*.



VI-C-2 - Dans quels répertoires Visual C++ Express cherche-t-il les fichiers .lib pour l'édition de lien ?

Les fichiers .lib sont recherchés dans l'ordre suivant :

- 1 les répertoires précisés en ligne de commande (/LIBPATH) ;
- 2 les répertoires contenus dans la variable d'environnement LIB.

Les répertoires précisés en ligne de commande (/LIBPATH) correspondent à ceux indiqués dans les options du linker : *Additional Library Directories* des options générales du linker. Sont aussi ajoutés les répertoires précisés dans la ligne *Reference Directories* des options *VC++ Directories*.

Les répertoires définis par la variable d'environnement LIB sont ceux définis au niveau de la configuration générale : *Library directories* des options *VC++ Directories*.

En cohérence avec les fichiers d'en-tête, les répertoires doivent être précisés comme suit :

- dans *Library directories* des options *VC++ Directories* (variable d'environnement LIB) : les bibliothèques générales ayant vocation à être utilisées dans tous les composants du projet ;
- dans *Additional Library Directories* des options générales de C/C++ et dans *Reference Directories* des options *VC++ Directories* : les répertoires propres à la solution lorsqu'elle est composée de plusieurs projets ainsi que les bibliothèques ayant une portée très locale.

VI-D - Configuration C++ générale

The screenshot shows the 'Additional Include Directories' configuration window. At the top, there are dropdown menus for 'Platform: Active(Win32)' and a 'Config' button. The main area contains a table of compiler options:

Additional Include Directories	
Resolve #using References	
Debug Information Format	Program Database for Edit And Continu
Common Language RunTime Support	
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level4 (/W4)
Treat Warnings As Errors	No (/WX-)
Multi-processor Compilation	
Use Unicode For Assembler Listing	

At the bottom, there is a section titled 'Additional Include Directories' with the text: 'Specifies one or more directories to add to the include path; separate with semi-colons if more than one (/I[path])'. There are 'OK' and 'Annu' buttons at the bottom right.

Il est intéressant de positionner le niveau d'avertissement à 4 pour obtenir un maximum de warning sur un projet. Un nouveau niveau est apparu avec Visual C++ 2010 : *EnableAllWarnings*. Celui-ci est beaucoup plus sévère en terme d'avertissement, et après quelques essais, il me semble même beaucoup trop sévère. Sur notre exemple créé ci-dessus, ce niveau génère plus de 560 messages d'avertissement sur les fichiers de la STL. Ces trop nombreux

messages d'avertissement masquent ceux qui pourraient être liés uniquement au code que vous avez produit. Aussi, je pense que le niveau le plus intéressant reste le niveau 4.

VI-E - Désactiver les extensions de langage

Configuration: All Configurations Platform: Active(Win32) Configuration Manager...

Disable Language Extensions	Yes (/Za)
Treat WChar_t As Built in Type	Yes (/Zc:wchar_t)
Force Conformance in For Loop Scope	Yes (/Zc:forScope)
Enable Run-Time Type Information	
Open MP Support	

Disable Language Extensions
Suppresses or enables language extensions. (/Za)

OK Annuler Appliquer



La première entrée de l'onglet *Langage* permet de désactiver les extensions de langage mises en œuvre par Microsoft. Attention, cette option est orthogonale avec l'option de compilation en C++0x ou pas. En effet, les nouveautés introduites par C++0x ne sont pas des extensions proposées par Microsoft mais bien des éléments (bientôt) standards du langage. En désactivant les extensions de langage, le compilateur reste strictement conforme à la norme C++0x si *Platform Toolset* est à *v100* et strictement conforme à la norme C++03 si *Platform Toolset* est à *v90*.

Il semble donc tentant de préciser cette option afin d'être garanti d'avoir un code le plus portable possible. Que nenni ! Il semblerait que cette option induise des bugs dans le compilateur (en n'arrivant pas à compiler du code qui devrait être compilable) et que Microsoft envisage même de rendre cette option obsolète. Il est légitime de regretter de ne pas pouvoir compiler au plus près de la norme mais à vous de mesurer le risque d'utiliser une option qui empêcherait de compiler correctement un projet (1) .

VII - C++0x

Le support C++0x n'est pas complet. Beaucoup de nouveautés pourtant intéressantes ne sont pas proposées par Visual Express 2010. Citons par exemple les variadic template, les évolutions sur les constructeurs, *default/delete*, les classes d'énumération, les nouvelles boucles *for*, l'assouplissement sur les types *union*, etc.

En revanche, Visual Express 2010 propose un bon support des nouveautés de la STL. On peut aussi noter le support des lambdas, du typage automatique (*auto*) ou de la détermination de type (*decltype*).

Farscape propose dans son article  [Visual C++ 2010 premières impressions](#) une matrice indiquant le support de C++0x par Visual 2010 :  [Support partiel de C++0x](#).

1 : Voir par exemple ce **rapport d'anomalie**. Stephan T. Lavavej, développeur chez Microsoft tient des propos semblable sur la mailing list de Boost. Merci mille fois à Arzar pour ces précieuses informations.