



1 INTRODUCTION

Pour programmer dans un langage de programmation, il est possible d'utiliser un simple éditeur de texte.

1.1 IDE

Le Notepad de Microsoft reste trop limité, mais des outils comme Notepad++ ou Bracket.io ou SublimeText permettent de créer et modifier du code.

Cependant, les langages étant de plus en plus évolués, il est devenu habituel d'utiliser un environnement dédié. Voici quelques exemples :

- Java : **NetBeans, Eclipse, IntelliJ idea**
- C# : **SharpDevelop, Visual Studio**
- PHP : **PHPStorm**

1.2 INTÉRÊT

Ces environnements facilitent le travail des développeurs, en proposant de nombreuses fonctions et des outils dédiés aux langages. En général :

- la possibilité de **compiler** le code sans utiliser de ligne de commandes,
- de **souligner les erreurs** directement sur le code,
- de **compléter** les mots et parfois proposer des optimisations
- de **déboguer** en suivant le programme ligne par ligne,
- de **proposer de l'aide** sur la documentation...

2 OBJECTIFS

Nous allons donc installer l'environnement de développement de Microsoft : Visual Studio Community 2017.

Ce TP a un double objectif :

1. Permettre d'utiliser un environnement de développement pour nos projets
2. Savoir installer et utiliser les fonctions de base des IDE¹

¹ Integrated Development Environment



3 PRÉPARATION

La plupart des installations sont très simples : [Suivant], [Suivant], [Suivant], [Suivant], etc.

Il faut pourtant s'assurer que les conditions soient réunies pour avoir un environnement opérationnel et complet.

De plus, un bon IDE utilise des librairies (des DLL), des extensions qui deviennent caduques ou incompatibles. Parfois, les versions les plus récentes contiennent des bugs (comme dans les anciennes versions) mais la différence est qu'étant récents, ils ne sont pas commentés ou expliqués.

Dans un environnement de production, on préférera utiliser une version dite stable, voir LTS².

3.1 PRÉREQUIS

Microsoft fournit une page contenant les prérequis. Savez-vous les lire ?

<https://www.visualstudio.com/fr-fr/productinfo/vs2017-system-requirements-vs>

Vous voyez dès à présent qu'il faut vérifier votre système d'exploitation avec les touches [Win]+[Pause].

À noter : dans les grandes entreprises, il est habituel que les mises à jour des postes soient limitées au strict nécessaire. Les applications métiers ont elles-mêmes des prérequis qui peuvent bloquer les évolutions de versions (longtemps ce fût le cas pour Internet Explorer 6). Les administrateurs de l'infrastructure doivent être contactés si les prérequis de votre IDE ne sont pas remplis.

3.2 PAQUETAGES ET LIBRAIRIES

Microsoft fournit également des éléments sur les paquetages (ou librairies) disponibles.

<https://docs.microsoft.com/en-us/visualstudio/install/workload-component-id-vs-community>

Dans cette installation, nous utiliserons une image ISO fournie par l'établissement, qui limite drastiquement l'usage du réseau mais réduit les fonctionnalités disponibles

² Long Term Support : la version ne connaît pas d'évolution majeure mais est supportée et corrigée pendant 2 à 3 ans.



4 INSTALLATION

Les captures d'écran peuvent être légèrement différentes selon l'OS d'installation.

4.1 MONTAGE DE L'IMAGE ISO

Sous Windows 10, il est possible de monter (sous-entendu simuler) un lecteur CD/DVD en cliquant sur le fichier ISO avec le bouton droit de la souris (et choisir 'monter' dans le menu).

Dans les versions antérieures, il est peut-être nécessaire de télécharger un utilitaire permettant de le faire. Le logiciel recommandé est "Virtual CD-ROM Control Panel" (<https://www.microsoft.com/en-gb/download/details.aspx?id=38780>) ou bien, WinCDEmu en version portable (ne modifie pas le système, <https://portableapps.com/apps/utilities/wincdemu-portable>).

Une fois l'image montée, lancez l'installation avec le programme  vs_setup.exe

4.2 INSTALLATION

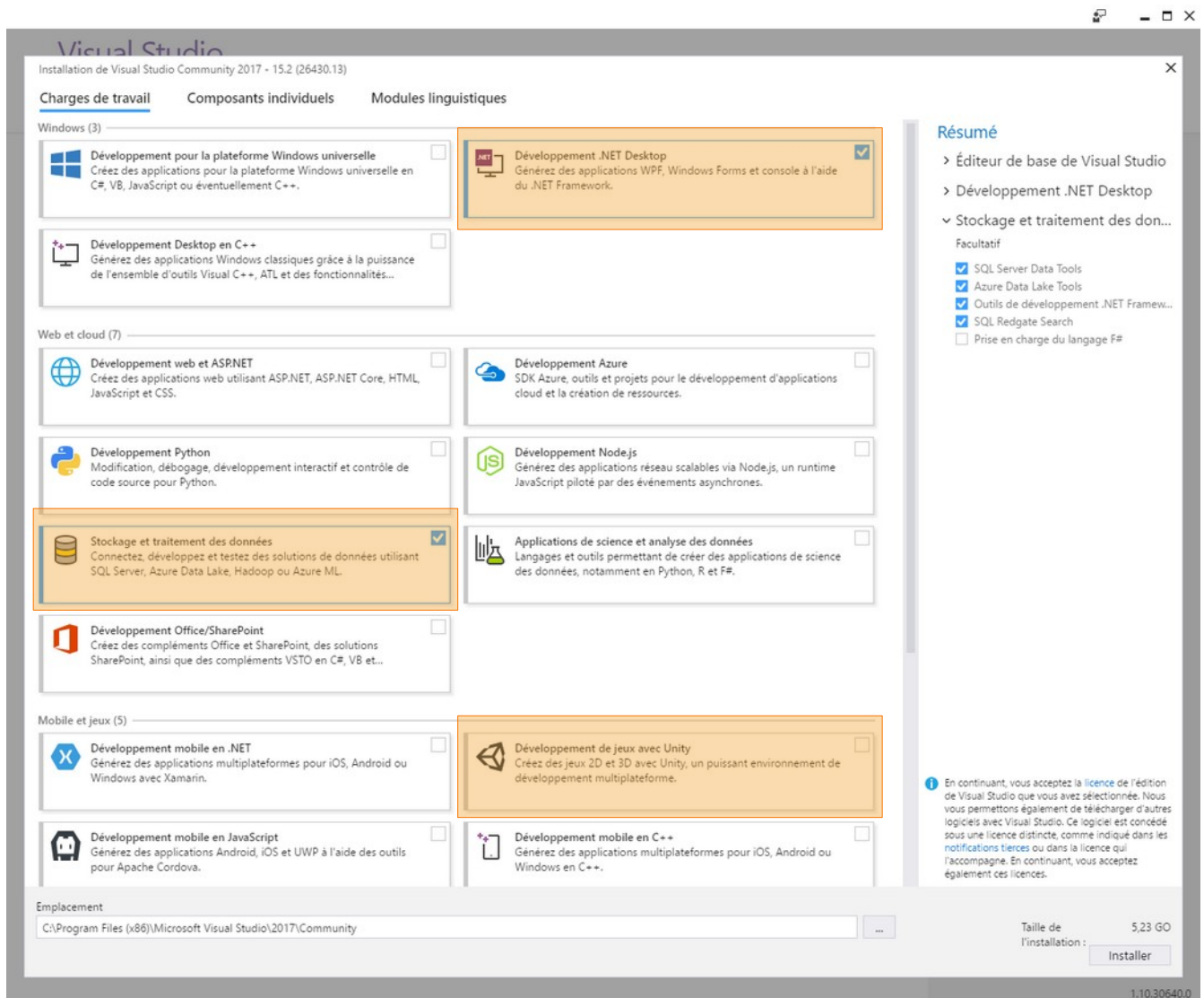
Acceptez les conditions de licence.



Le programme propose alors une liste de choix à cocher, par paquetages (onglet "Charges de travail") ou de manière plus fine, par composants (onglet "composants individuels").

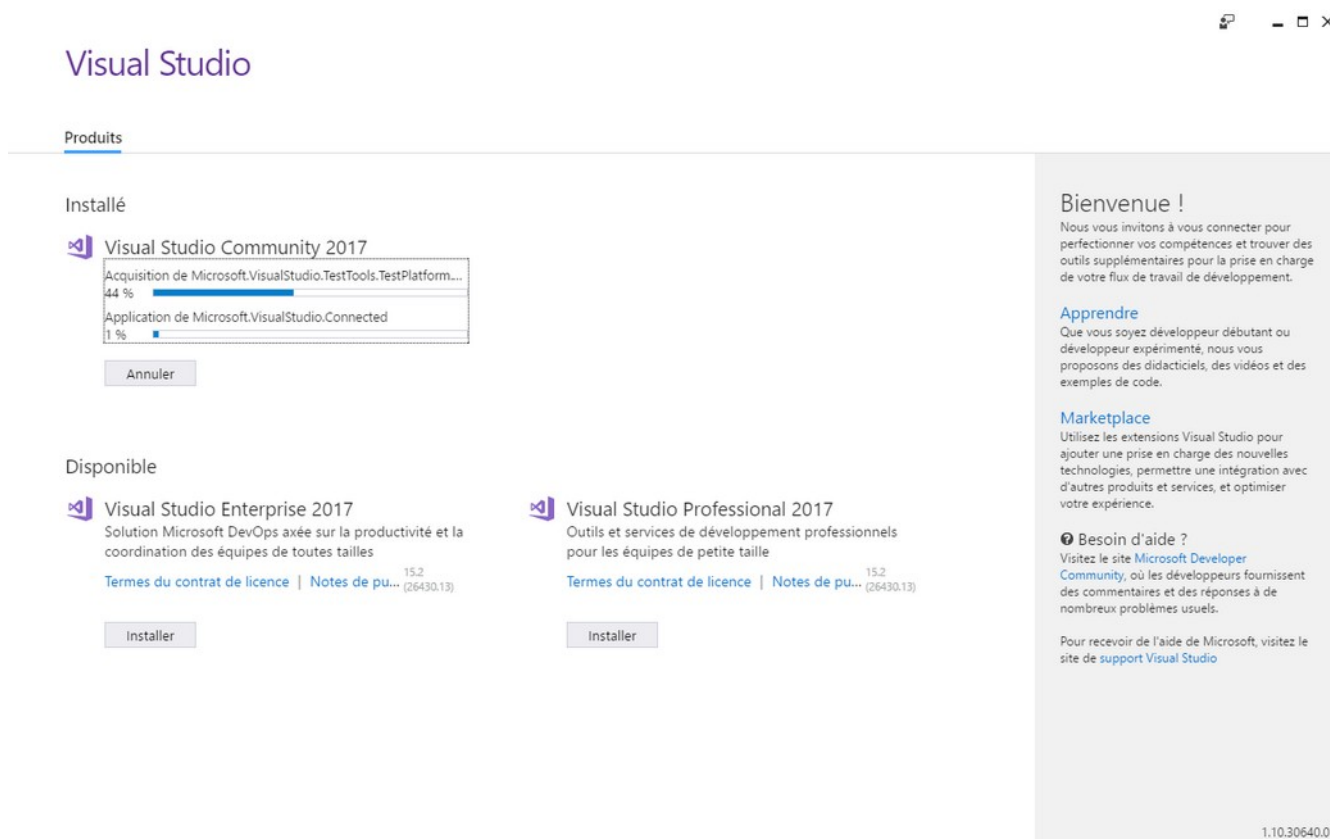
Choisissez seulement **Developpement .NET Desktop** et **Stockage et traitement de données**.

Si vous êtes en SLAM2, ajoutez le pack **Développement de jeu avec Unity**.



L'installation prend dans ce cas environ 5,3 Go sur le disque C:.

L'installation peut être très longue et nécessite parfois l'accès au réseau : ne cochez pas d'options supplémentaires sous peine d'avoir une installation qui ne fonctionnera pas.



L'installation devrait se terminer sans erreur.

4.3 GESTION DES PROBLÈMES

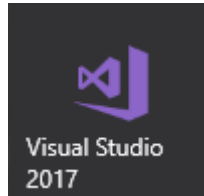
Il y a plusieurs cas de figures mais voici les principaux :

Problème rencontré	Solutions potentielles
Visual Studio indique un problème réseau	Vérifier que le poste peut atteindre le réseau Vérifier les paramètres du proxy d'Internet Explorer
Visual Studio requiert un framework .NET	Installer le framework fournit au même endroit que l'image ISO (NDP461...)



5 CONFIGURATION

Il est maintenant possible d'utiliser Visual Studio 2017 community sur votre poste. Pour cela, trouvez son icône et lancez-le :

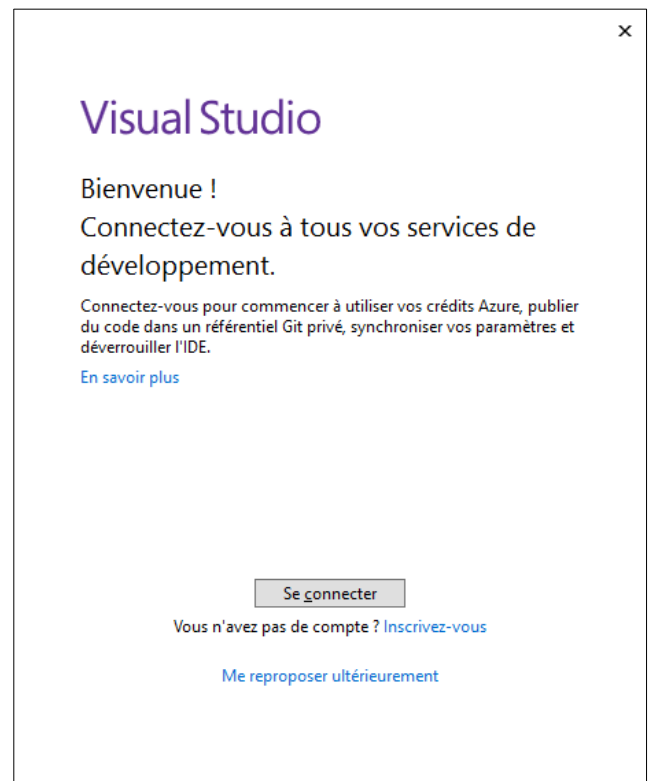
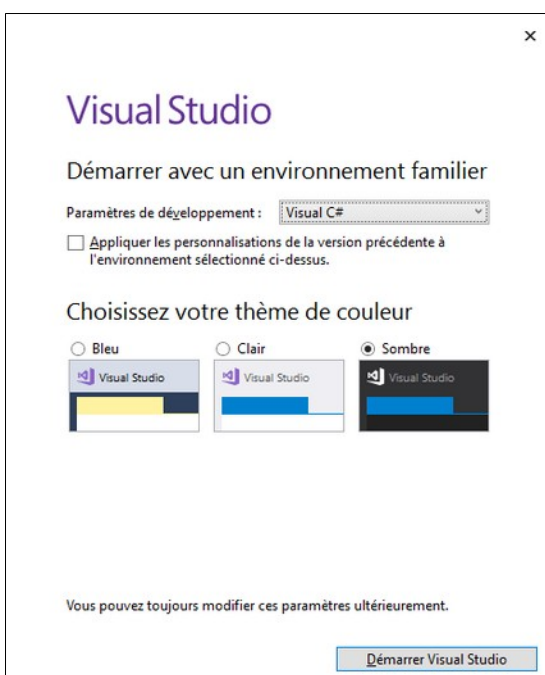


Il est conseillé d'avoir un compte Microsoft pour faire fonctionner pleinement l'IDE, mais ce n'est pas obligatoire.

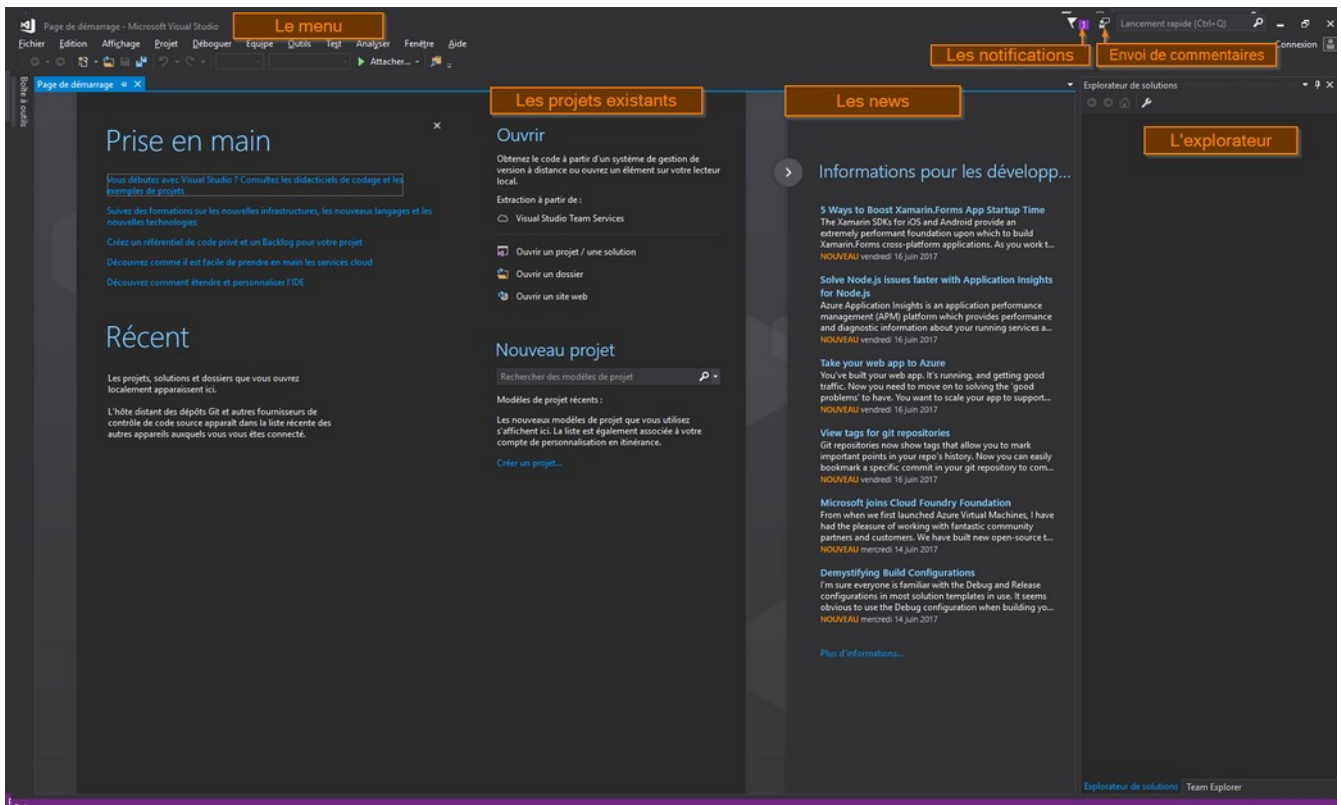
Il suffit de créer un compte que vous pourrez utiliser pour des fonctions avec le cloud Microsoft Azure (voir également dans les autres cours).

Il est également possible de cliquer sur "Me reproposez ultérieurement" pour pouvoir utiliser directement l'environnement de développement ;

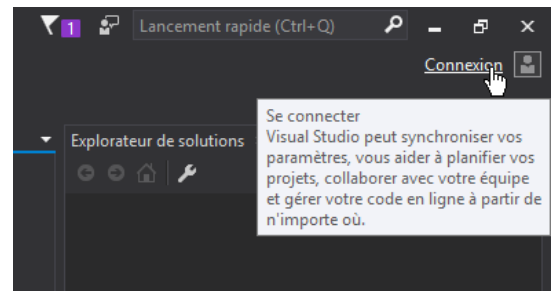
Je vous recommande par la suite de configurer votre environnement pour Visual C#.



Après quelques instants de préparation, l'environnement s'affiche.



Si vous voulez vous connecter et être reconnu en tant qu'utilisateur, il suffit de cliquer sur le mot "connexion" en haut à droite :



Si vous voulez connaître les avantages d'utiliser un compte Microsoft, vous pouvez consulter le lien suivant :

<https://www.visualstudio.com/en-us/docs/setup-admin/team-services/sign-up-for-visual-studio-team-services>

Il vous indiquera comment utiliser un projet sur un serveur web sécurisé de la forme

<https://{youraccount}.visualstudio.com>

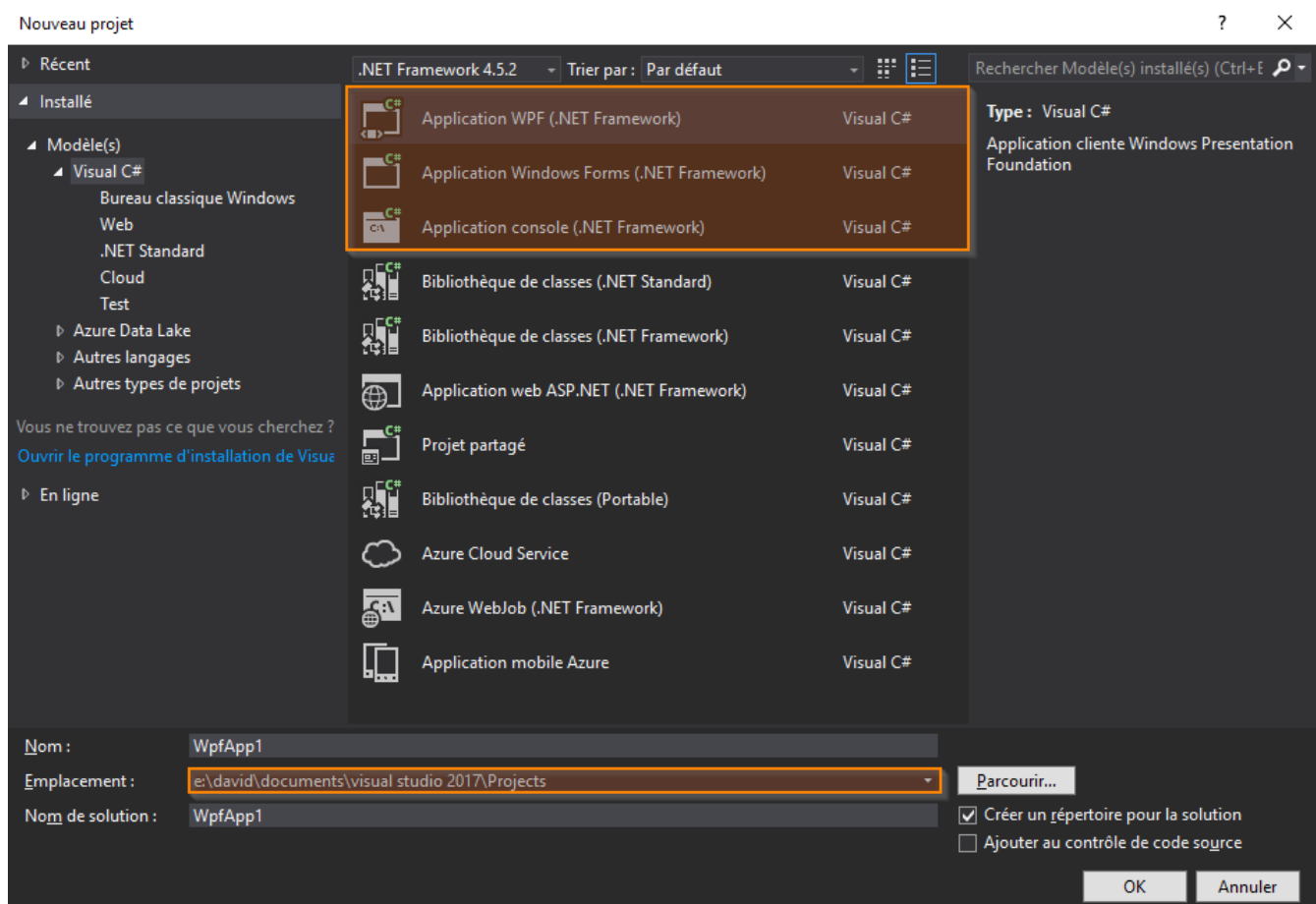


6 CRÉATION D'UN PROJET

Pour valider le fonctionnement de l'environnement, vous allez créer un petit projet.

Cliquez sur "Créer un projet..." tout en bas, sous [Nouveau projet](#)

Le raccourci-clavier pour créer un nouveau projet est CTRL+Maj+N. Les projets les plus fréquents sont en haut. Nous allons retenir un projet "Application console".



Il faut choisir un emplacement pour le projet : il doit permettre un lancement rapide, donc un disque-dur. Pourtant, il pourra être utile plus tard de choisir un système en réseau.

Dans tous les cas, il est recommandé de faire une sauvegarde du répertoire où seront tous les projets, avec un outil comme Yadis ! Backup, Synchronicity ou celui de votre choix.



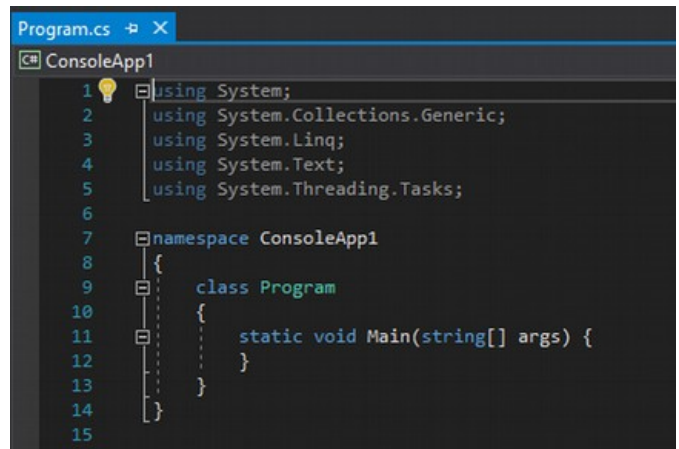
Après quelques secondes, Visual Studio devrait avoir créé les bases de notre projet :

On constate qu'il y a 14 lignes alors que le programme ne fait rien pour le moment.

Pour ce premier programme, nous allons seulement utiliser une fonction d'affichage.

```
Console.WriteLine("Hello Moto !");
```

La fonction `WriteLine()` fait partie d'une bibliothèque de fonction qui s'appelle *Console*. Plus exactement, *Console* est un objet sur lequel on peut agir avec les fonctions qui sont fournies par la classe.



On décrit toujours l'objet sur lequel on veut agir, puis la fonction à utiliser : les deux sont séparés par un point. `Chien.asseoir()` ou `Chat.miauler()` signifie donc que le chien doit s'asseoir ou que le chat doit miauler.

Il faut **ajouter cette ligne** entre les lignes 11 et 12, comme ceci :

```
static void Main(string[] args) {
    Console.WriteLine("Hello Moto !");
}
```

Vous pouvez maintenant tester ce premier code en C#. Pour cela, il existe 3 possibilités :

Par le menu	Par le raccourci-clavier [F5]	Par l'icône

Comme vous pouvez l'entre-apercevoir, il se passe quelque chose mais c'est rapide. C'est normal, nous n'avons pas demandé à notre programme d'attendre. Modifiez le code comme suit et recommencez l'exécution.

```
/* Voici le code d'exemple de base : Hello World */
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args) {
            Console.WriteLine("Hello World");
            Console.ReadKey();
        }
    }
}
```



6.1 MODIFICATION DU PROJET

À partir de là, il faut connaître des éléments de programmation que nous verrons en cours.

Pour s'amuser un peu, il est possible de proposer à la machine d'écrire votre nom, en utilisant une variable. Considérons qu'une variable est une boîte, dans laquelle on peut placer ce que l'on veut.

```
/* Voici le code d'exemple de base : Hello World */
using System;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args) {
            String nom = "";
            Console.WriteLine("Quel est votre nom ? ");
            nom = Console.ReadLine();

            Console.WriteLine("Hello " + nom);
            Console.ReadKey();
        }
    }
}
```

Ici, le programme vous demandera de saisir du texte au clavier et attendra patiemment l'appui sur la touche [Entrée].

Il affichera ensuite "Hello " suivi de votre saisie : peu importe que ce soit votre nom ou pas, l'ordinateur ne peut pas comprendre ce que vous avez écrit, il se contente de l'afficher.



La variable (la boîte) s'appelle "nom" mais vous pouvez en créer d'autres ou bien changer son nom. L'important, est de conserver le même nom lors de l'affichage.

Amusez-vous à saisir d'autres variables de votre choix et à afficher des messages.


7 COMPILATION ET OUTILS

Un IDE est un outil plus puissant qu'un simple éditeur de texte.

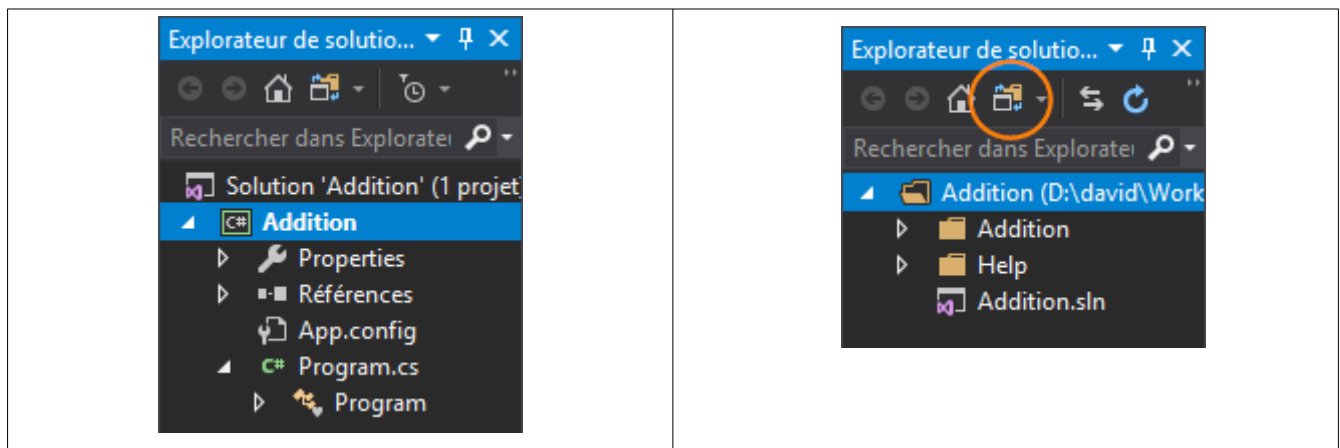
7.1 EXPLORER DE SOLUTIONS

Visual Studio dispose d'un explorateur pour afficher les fichiers liés au projet, ou bien les fichiers et dossiers sur le disque.

Pour activer l'explorateur : CTRL+Alt+L

 Explorateur de solutions Ctrl+Alt+L

La vue peut être modifiée en cliquant sur l'icône des dossiers.




7.2 PROPRIÉTÉS (INTERFACES GRAPHIQUES)

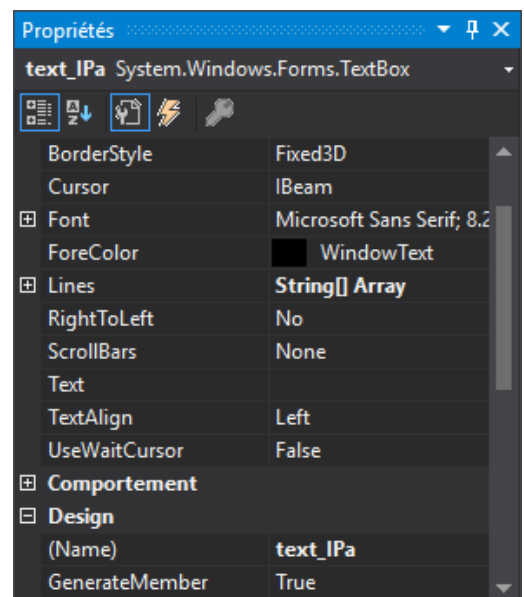
Lors de l'utilisation d'interfaces graphiques, toutes les propriétés des formes employées sont éditables dans la partie en bas à droite de l'IDE.

Semble s'activer avec Alt+Entrée.

Les zones suivantes sont utiles pour :

- **Apparence** : modifier les couleurs et les traits des champs.
- **Comportement** : Définir si le champ est visible, accessible en lecture et écriture, etc.
- **Design** : le nom du champ
- **Disposition** : l'emplacement sur la fenêtre.

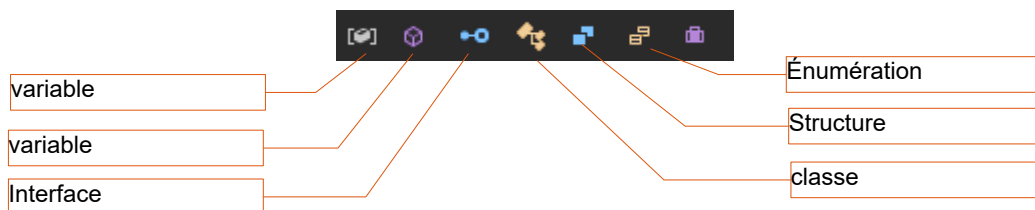
L'icône en forme d'éclair permet de voir les actions :  définir les fonctions sur les clics, double-clics et autres événements.



7.3 AIDES D'ÉCRITURE

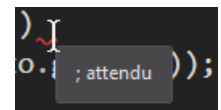
7.3.1 Complétion

Comme les éditeurs un peu spécialisés, un IDE permet de compléter les mots en cours d'écriture. L'environnement Visual Studio propose les mots avec un symbole devant, permettant de déterminer son utilité :



7.3.2 Syntaxe

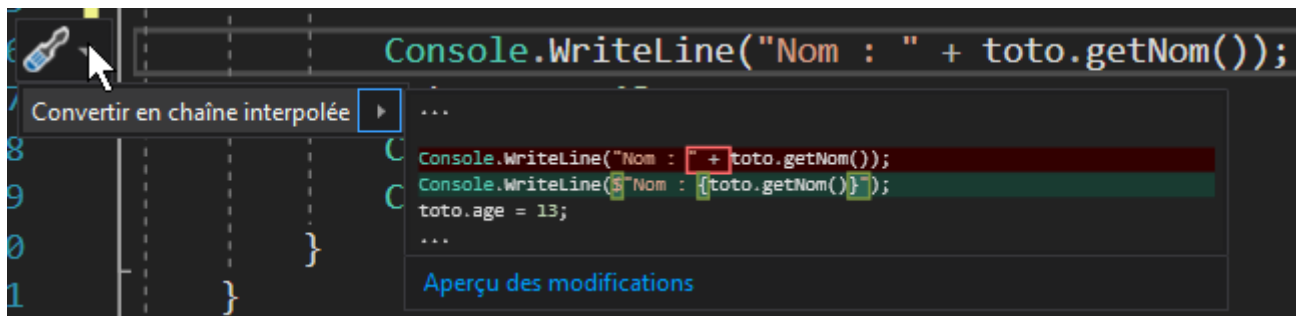
L'IDE est optimisé pour la syntaxe du langage utilisé. Il peut donc facilement indiquer pourquoi il ne compilera pas le programme.



7.3.3 Optimisation

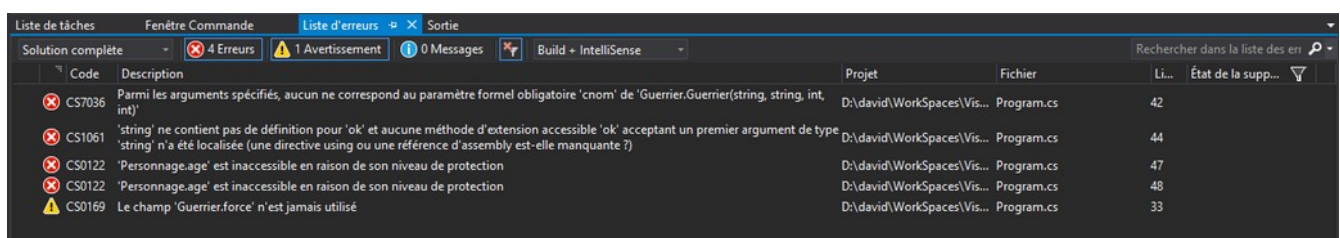
Lorsqu'une ligne a été saisie, l'IDE peut proposer une formulation différente, à l'aide d'une icône tournevis en début de ligne.

L'exemple ci-dessous, propose l'usage d'une chaîne interpolée.



7.3.4 Liste des erreurs

Lors d'une compilation, l'IDE indique où sont les erreurs. Conseil : Traiter les messages dans l'ordre !



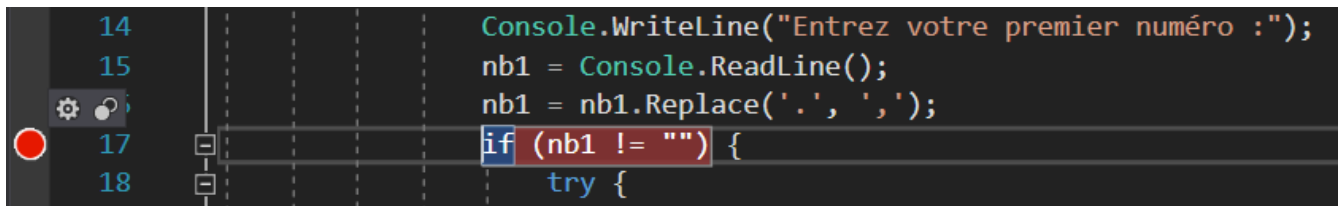
7.4 DEBOGAGE

La puissance des langages compilés vient de la rapidité d'exécution. Toutefois, l'analyse du programme devient plus compliquée, car il est humainement illisible (bytecode).

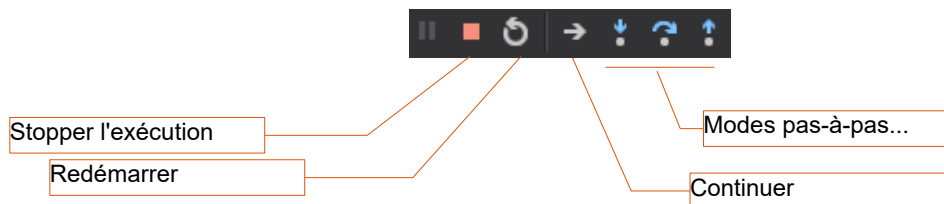
Il est alors intéressant d'utiliser le débogueur.

Le débogueur fonctionne en permanence lorsque l'application est lancée depuis l'IDE.

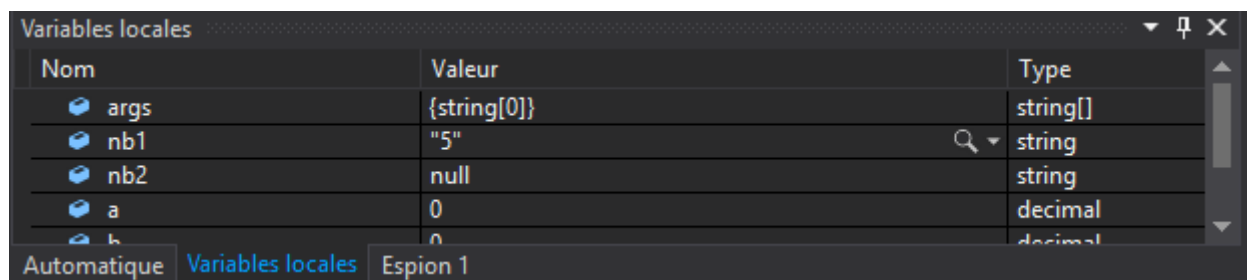
Il est possible de placer des points d'arrêt dans le code, en cliquant à gauche du numéro de ligne : à la prochaine exécution du programme, il s'arrêtera ici.



Le code sera arrêté jusqu'à l'utilisation des touches de reprise en pas-à-pas ou reprise normale.



Dans ce mode, l'IDE affiche une fenêtre contenant les variables et leurs valeurs au moment de l'arrêt :





7.5 TRANSMETTRE UNE SOLUTION (PAR EMAIL)

Il est possible de créer un fichier compressé (Bandizip, 7Zip, etc) et de l'envoyer par courriel. Cependant, les répertoires **bin** et **obj** doivent être d'abord supprimés. Ils contiennent les fichiers exécutables (votre code compilé) qui ne passeront pas les antivirus.

