



## Module 1

# Introduction à C# et au Framework .NET

# Présentation du Module

- Introduction au Framework .NET
- Ecrire une application C#
- Déboguer des Applications avec Visual Studio

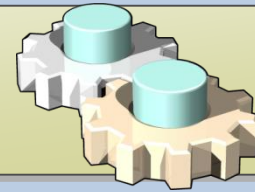
# Leçon 1: Introduction au Framework .NET

- Qu'est-ce que le Framework.NET?
- Le Principe de C#
- Qu'est-ce qu'un Assembly?
- Comment le Common Language Runtime Charge, Compile, Exécute des Assemblies
- Quels sont les outils fournis par le Framework .NET Framework?

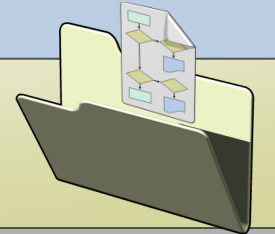
# Qu'est-ce que le Framework .NET Framework?

## Microsoft® .net Framework

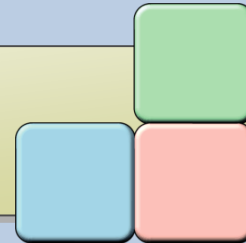
Common Language Runtime



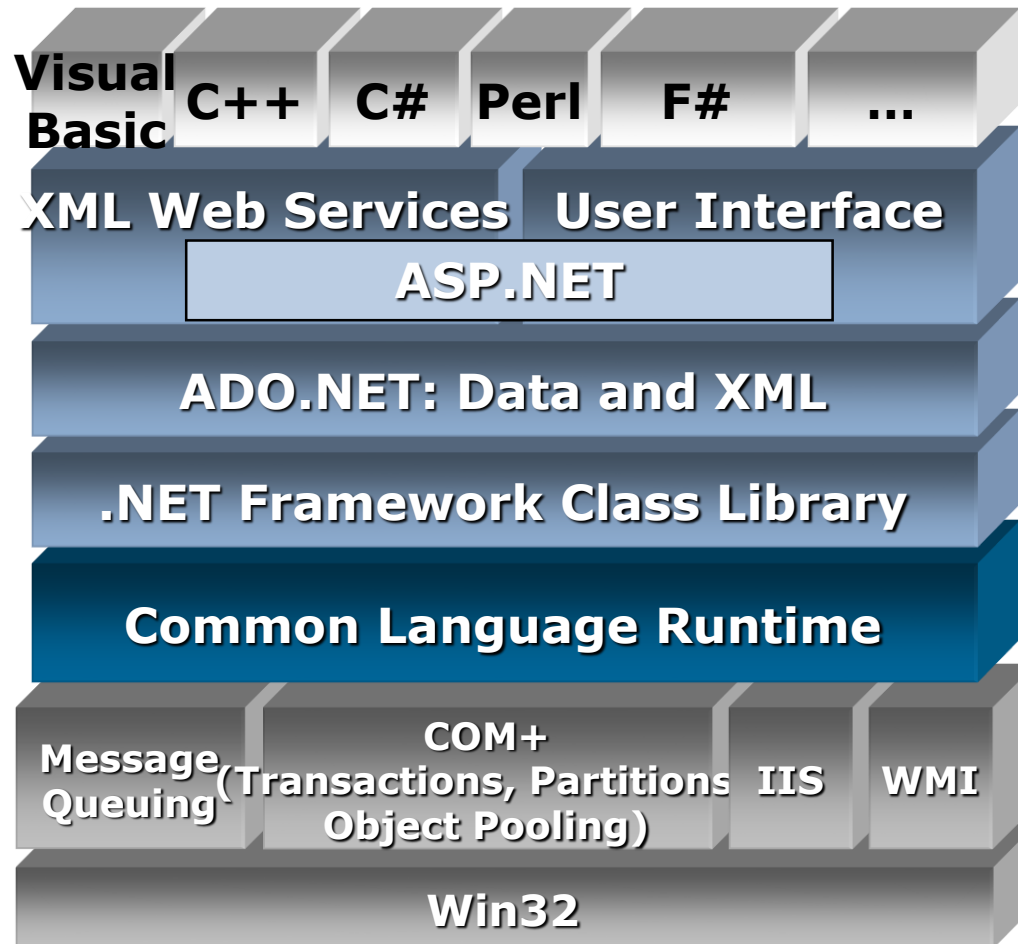
Class Library



Development Frameworks



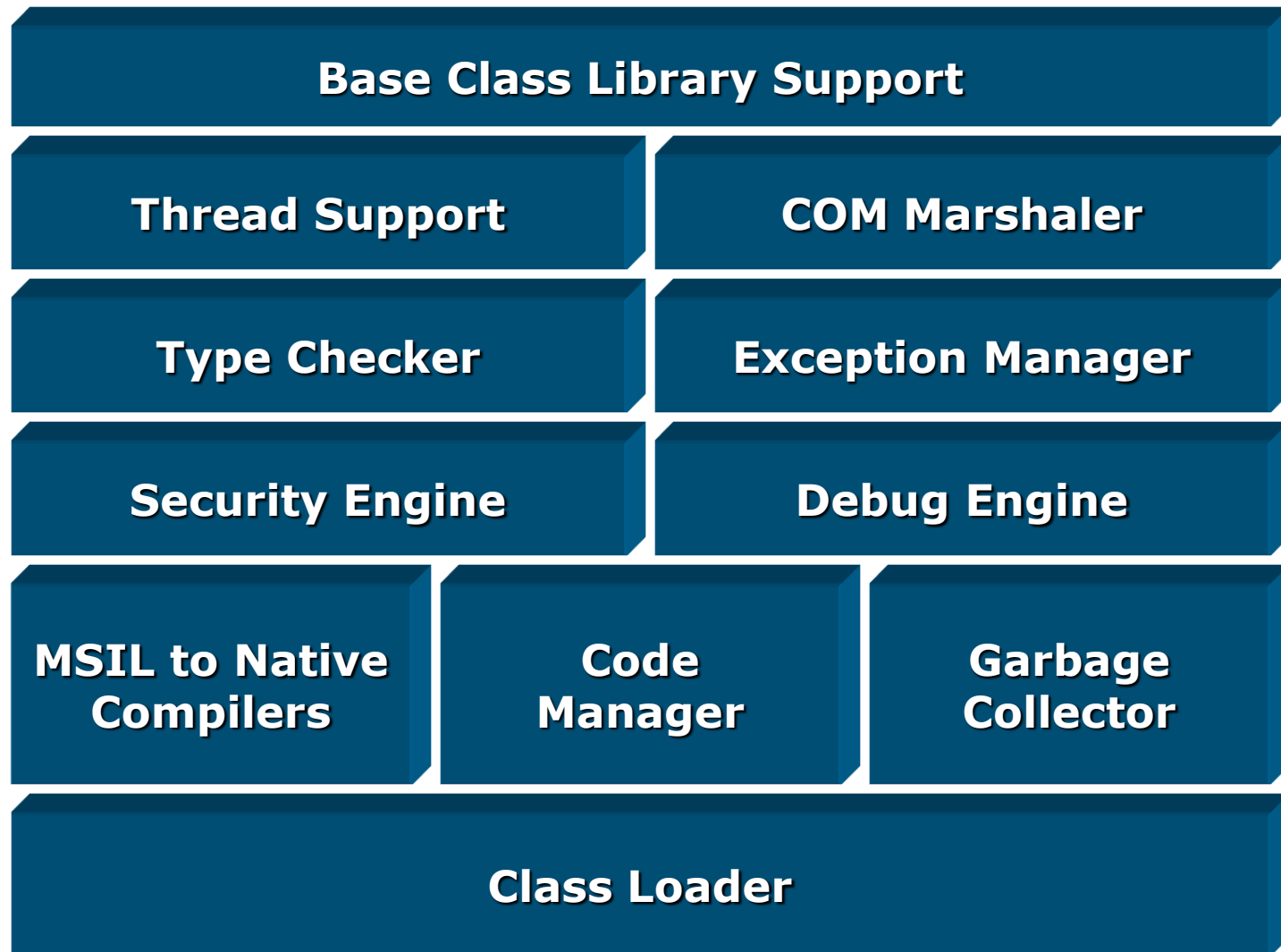
# Aperçu du Framework .Net



# Les composants du Framework .Net

- Common Language Runtime
- .NET Framework Class Library
- ADO.NET: Data and XML
- WPF et XML Web Services
- etc

# Common Language Runtime



# Le Principe de C#

## **C#**

C# est le langage de predilections pour les développeurs qui souhaitent contruire des application basées sur .NET

C# utilize une syntaxe très proche de C, C++, et Java

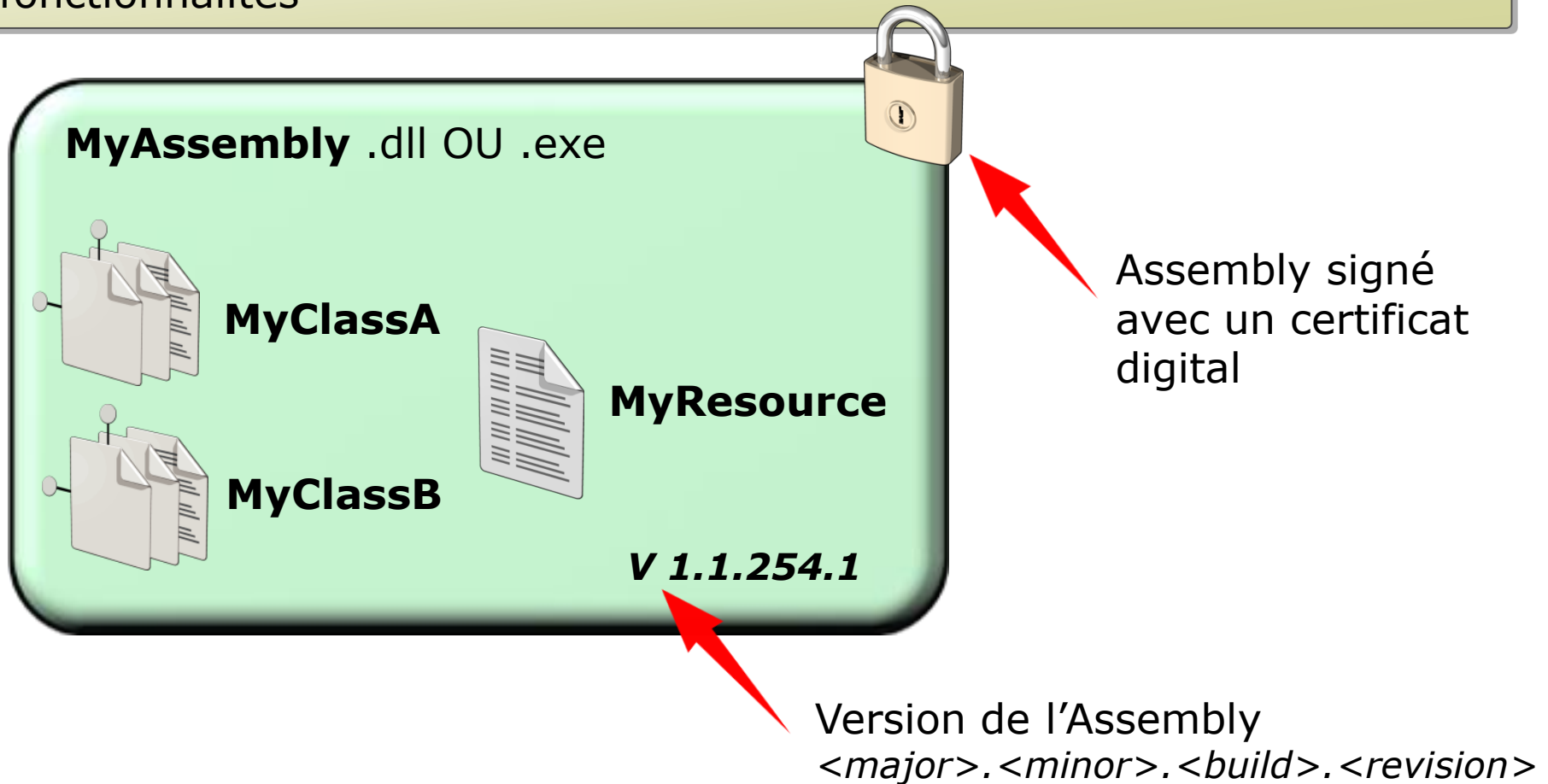
C# a été standardisé et décrit par la spécification ECMA-334 C# Language Specification



# Qu'est-ce qu'un Assembly?

Blocs de construction des applications .NET Framework

Collection de types et de ressources qui forment une unité logique de fonctionnalités



# Comment le Common Language Runtime Charge, Compile et Exécute des Assemblies

Les Assemblies contiennent du code MSIL code, qui n'est pas executable

Le CLR charge le MSIL depuis un assembly et le convertit en code machine requise par le device

**1**

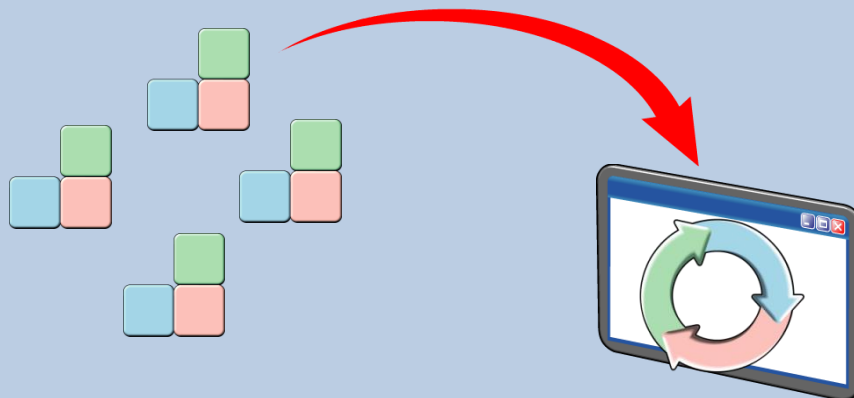
Chargement des assemblies que l'application références

**2**

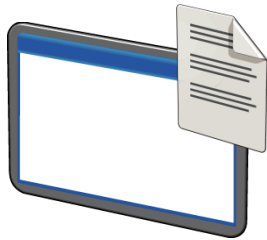
Verification et compilation des assemblies en code machine

**3**

Exécution de l' assembly



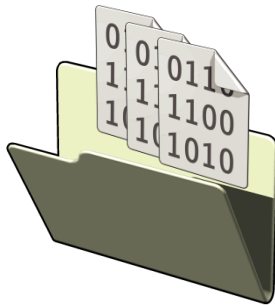
# Quels sont les outils fournis par le Framework .NET Framework?



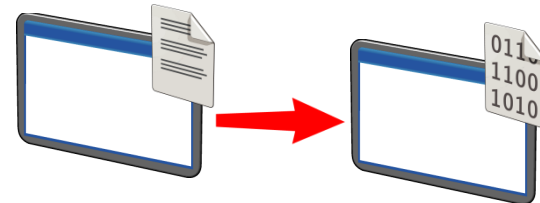
**Caspol.exe**



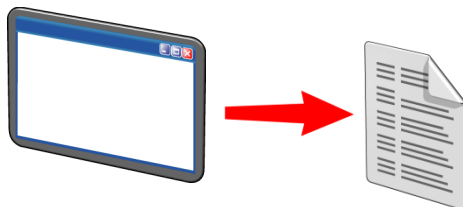
**Makecert.exe**



**Gacutil.exe**



**Ngen.exe**



**Ildasm.exe**



**Sn.exe**

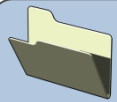
## Lesson 3: Ecrire une Application C#

- Qu'est-ce que les Classes et les Namespaces?
- Structure d'une Application Console
- Réaliser des opérations d'Input et Output dans une Application Console
- Bonnes pratiques pour les Commentaires

# Qu'est-ce que les Classes et les Namespaces?

Une classe est essentiellement un plan d'action qui définit les caractéristiques d'une entité

Un espace de noms représente une collection logique des classes



## **System.IO namespace**



**File** class



**FileInfo** class



**Path** class



**DirectoryInfo** class



**Directory** class

# Structure d'une Application Console

```
using System;
```

```
namespace MyFirstApplication
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

Ajout du namespace **System** au Scope du fichier

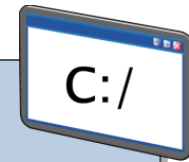
Déclaration du Namespace

Déclaration de la classe **Program**

Déclaration de la méthode **Main**

# Réaliser des opérations d'Input et Output dans une Application Console

## Méthode présentes dans System.Console :



Clear()

Read()

ReadKey()

ReadLine()

Write()

WriteLine()

```
using System;  
...  
Console.WriteLine("Hello there!");
```

# Bonnes pratiques pour les Commentaires

Débuter les procédures en utilisant un bloc de commentaire

Dans les procédures les plus longues, utiliser les commentaires pour briser les unités de travail

Lorsque vous déclarez des variables, utiliser un commentaire pour indiquer comment la variable sera utilisée

Lorsque vous écrivez une structure de décision, utiliser un commentaire pour indiquer la façon dont la décision est prise et ce que cela implique

```
// This is a comment on a separate line.  
string message = "Hello there!"; // This is an inline comment.
```

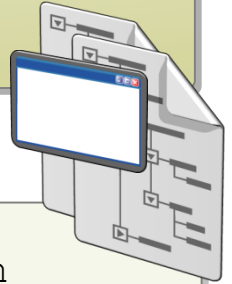


## Leçon 3: Documenter une Application

- Qu'est-ce que les commentaires XML?
- Tags de commentaires XML communs
- Générer la documentation à partir des commentaires XML

# Qu'est-ce que les commentaires XML?

**Utilisez les commentaires XML pour générer la documentation d'aide de votre application**



```
/// <summary> The Hello class prints a greeting on the screen
/// </summary>
public class Hello
{
    /// <summary> We use console-based I/O. For more information
    /// about
    /// WriteLine, see <seealso cref="System.Console.WriteLine"/>
    /// </summary>
    public static void Main( )
    {
        Console.WriteLine("Hello World");
    }
}
```

# Tags de commentaires XML communs

## On peut citer:

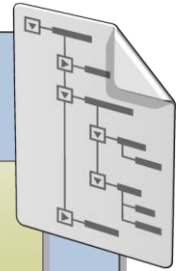
`<summary> ... </summary>`

`<remarks> ... </remarks>`

`<example> ... </example>`

`<code> ... </code>`

`<returns> ... </returns>`



# Générer la documentation à partir des commentaires XML

Générer un fichier XML à partir de Visual Studio

Générer un fichier XML à partir csc.exe

```
<?xml version="1.0"?>
<doc>
  <assembly> <name>MyProject</name></assembly>
  <members>
    <member name="T:Hello">
      <summary> The Hello class prints a greeting on the screen
      </summary>
    </member>
    <member name="M:Hello.Main">
      ...
    </member>
  </members>
</doc>
```

Consommer le fichier XML dans Sandcastle Help File Builder

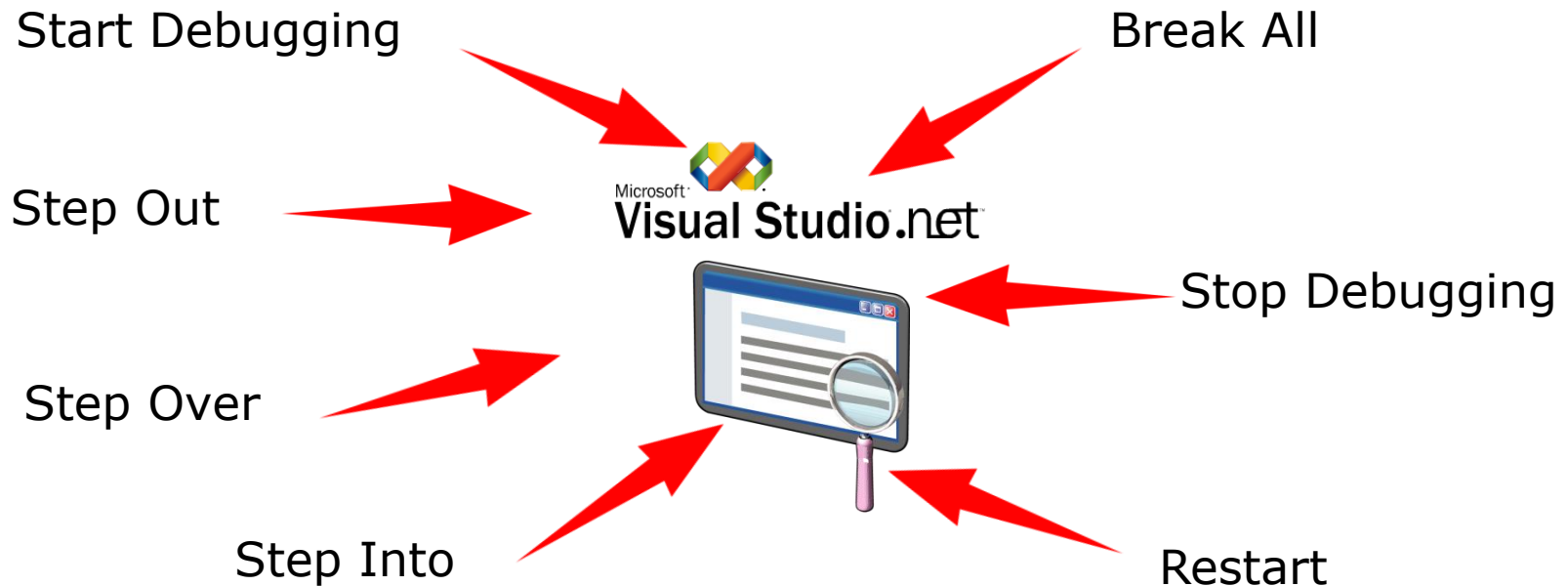
# Leçon 6: Déboguer des Applications avec Visual Studio

- Debogage dans Visual Studio
- Les Points d'Arrêts
- Mode Pas à pas
- Les fenêtres de Debug

# Debuggage dans Visual Studio

Les débogage est une étape essentielle du développement




Visual Studio fournit plusieurs outils pour déboguer du code



# Les Points d'Arrêts

Quand vous exécutez une application en mode Debug mode, vous mettez en pause l'exécution et rentrez en mode break




## **Visual Studio vous permet de :**

-  Localiser une ligne de code spécifique et y insérer un point d'arrêt
-  Localiser un point d'arrêt et le désactiver
-  Localiser un point d'arrêt et le supprimer

# Mode Pas à pas

Vous pouvez executer une instruction à la fois pour voir exactement ce qui se passe dans votre application

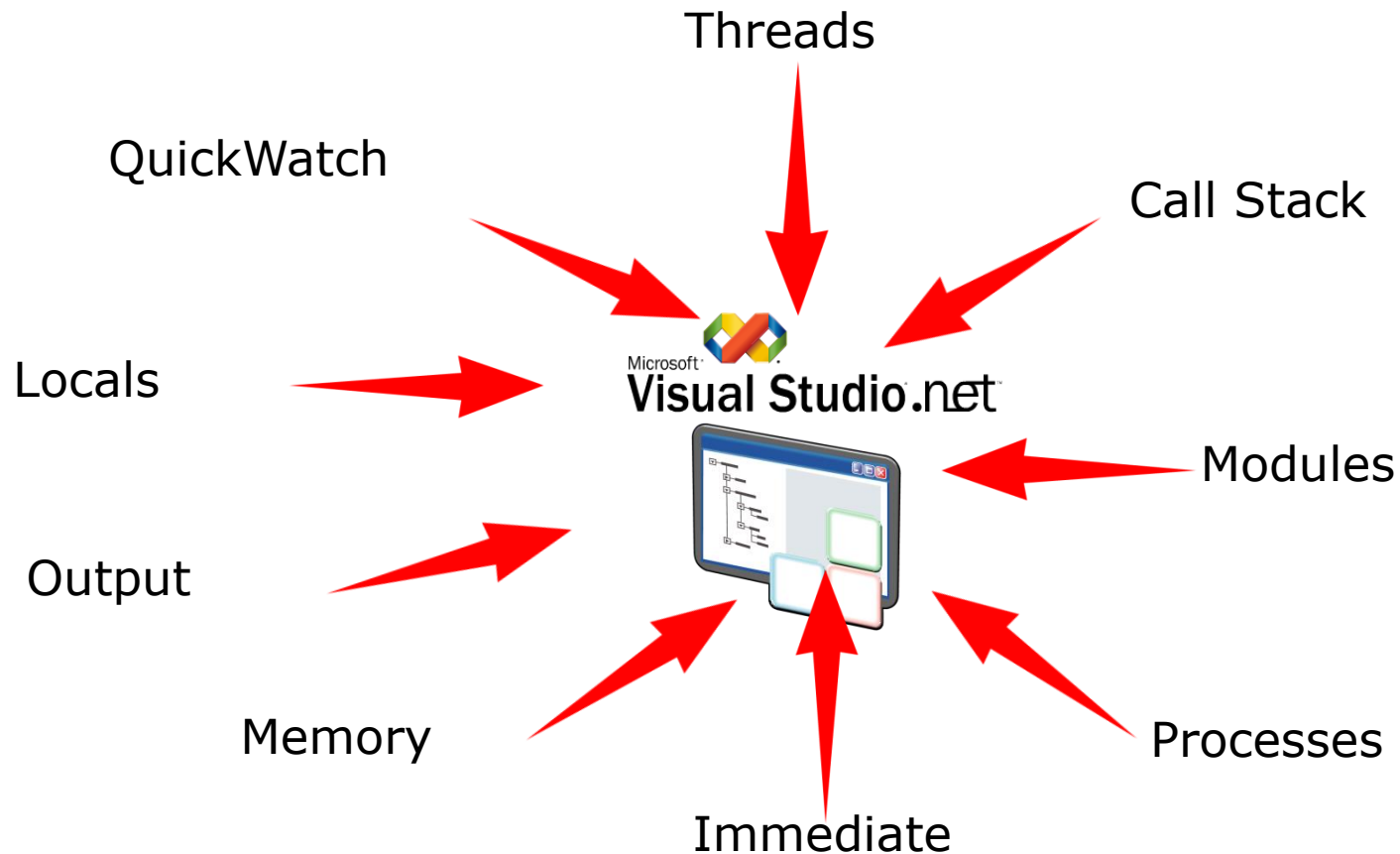
## **Visual Studio vous permet de:**

-  Pénétrez dans l'instruction en cours
-  Enjamper l'instruction en cours
-  Sortir de l'instruction en cours



# Les fenêtres de Debug

Visual Studio comprend plusieurs fenêtres que vous pouvez utiliser pour vous aider dans le débogage de vos applications



# Lab: Introducing C# and the .NET Framework

- Exercice 1: Créer une application console simple
- Exercice 2: Vérifier l'Application
- Exercice 3: Générer la Documentation d'une Application