

Visual Studio

Programmation .Net avec C#

Présentation

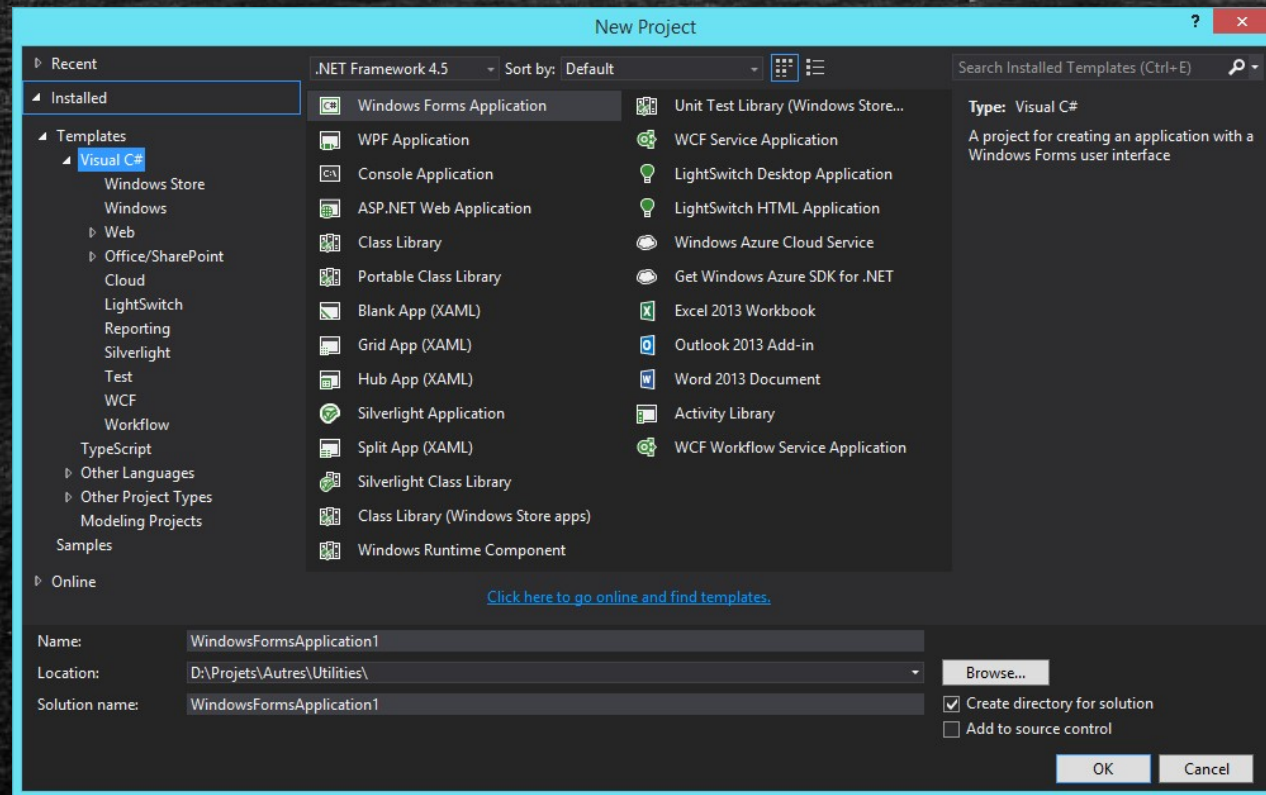
- Environnement de développement privilégié pour les solutions .Net.
- Offre un environnement tout en un :
 - Intègre les différents langages : C#, VB.Net, ...
 - Intègre les différents projets possibles : WinForm, WPF, ASP.Net, WCF ...
 - Choix de la version du Framework cible.



Modèles de projets

- Clients lourds :
 - Console.
 - Windows Forms.
 - WPF.
- Web :
 - ASP.Net.
 - Silverlight.
- Services :
 - Bibliothèque (DLL).
 - WCF.
 - Service Web.

Modèles de projets



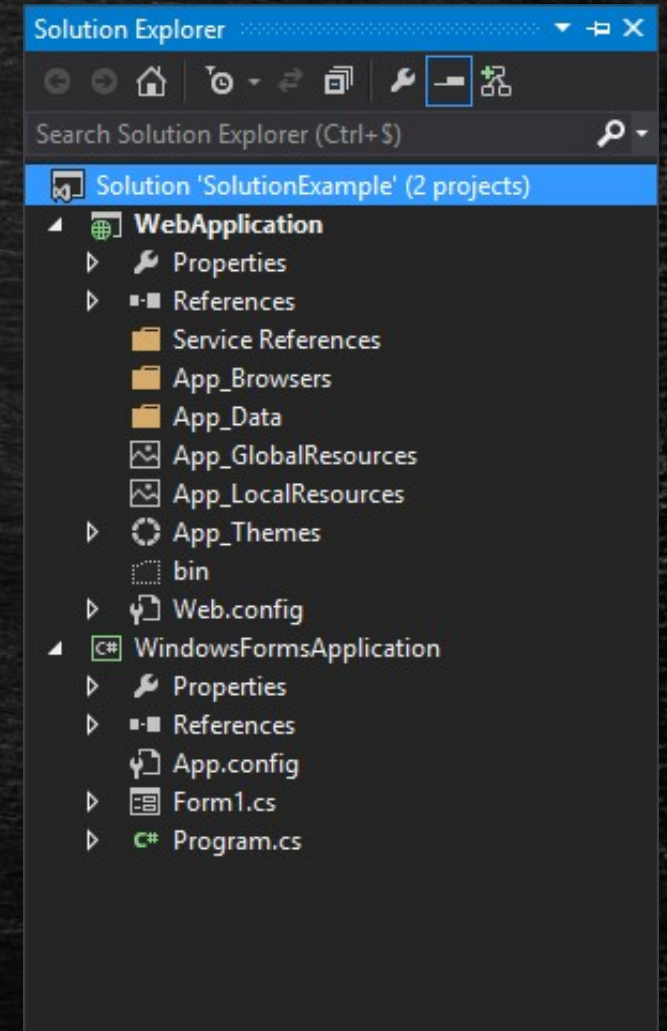
Fenêtre de sélection du type du nouveau projet.

Analyse de la structure d'un projet C#

- Fichiers Solution (.sln, .suo) :
 - Les fichiers projets à inclure (.sln)
 - Métadonnées (exemple tâches) (.suo)
- Fichier Projet (.csproj).
- Éléments d'un projet Windows :
 - Classes, formulaires, modules : (.cs)
 - Designer : (.Designer.cs)
- Éléments d'un projet Web :
 - Services Web XML (.asmx)
 - Formulaires Web (.aspx)
 - Classes globales d'applications (.asax)

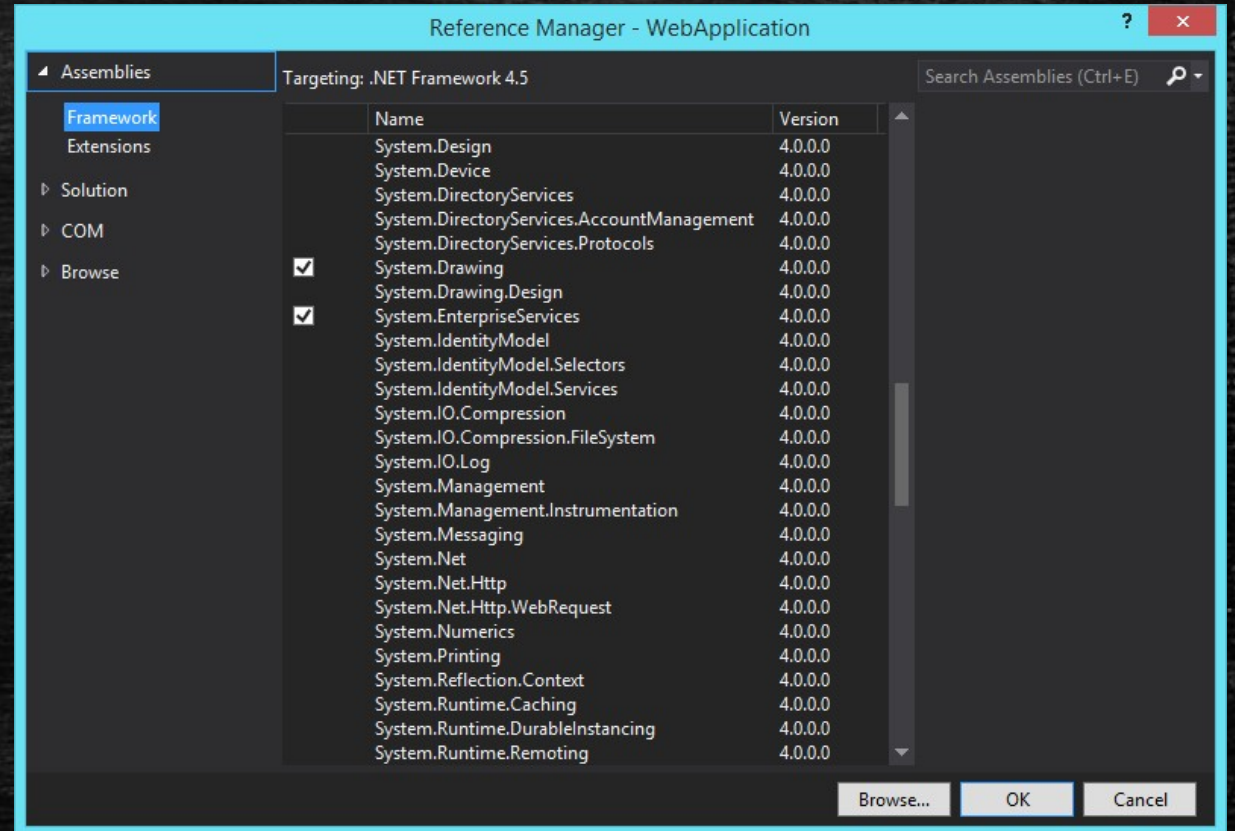
Analyse de la structure d'un projet C#

- Répertoires – Site Web :
 - App_Browsers : fichiers de définition de navigateurs.
 - App_Data : MDB ou XML de fonctionnement.
 - App_GlobalResources : fichiers ressources globaux.
 - App_LocalResources : fichiers ressources page.
 - App_Themes : fichiers d'apparence globale.
 - App_WebReferences : références Web (services).
 - Bin : fichiers de code compilés.
- Répertoires – Projet Windows :
 - Properties : propriétés projet
 - References : fichiers de code compilés.
 - Web References : références Web (services).



Les références du projet

- Le résultat de la compilation produit des assemblies.
 - Menu Build / Générer
- Le projet peut référencer :
 - Les assemblies .Net.
 - Les composants COM.
 - Les projets de la solution.



Notion de namespace

- Un classement des classes.
 - Toute classe définie dans un assembly possède un namespace.
- Une Zone de déclaration qui permet de délimiter la recherche de classe par le compilateur.
- Organisent les objets dans un assemblage.
 - Structuration et regroupement logique des objets.
- Créent les noms des objets.
 - Prévention des ambiguïtés.
 - Prévention des conflits entre les noms des classes.
 - Si deux types ont le même nom, il est nécessaire de préfixer ces types par le nom complet du type : Nom de l'espace de nom + nom du type.

Propriétés générales du projet

WebApplication* [icon] [X]

Application

Configuration: N/A Platform: N/A

Assembly name: WebApplication Default namespace: WebApplication

Target framework: .NET Framework 4.5 Output type: Class Library

Startup object: (Not set) Assembly Information...

Resources

Specify how application resources will be managed:

☒ Icon and manifest

A manifest determines specific settings for an application. To embed a custom manifest, first add it to your project and then select it from the list below.

Icon: (Default Icon) ... [icon]

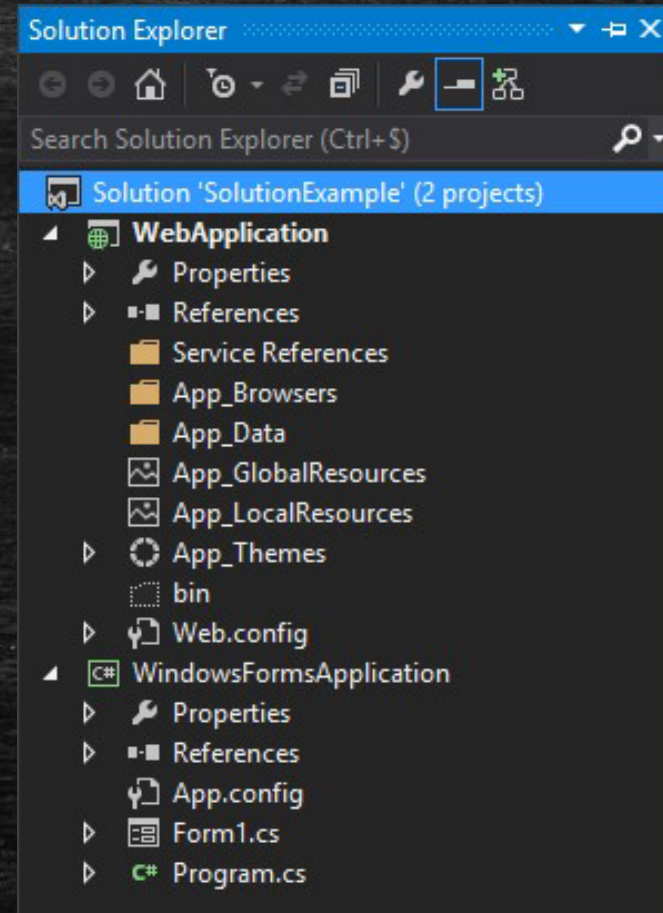
Manifest: Embed manifest with default settings

☐ Resource file: ...

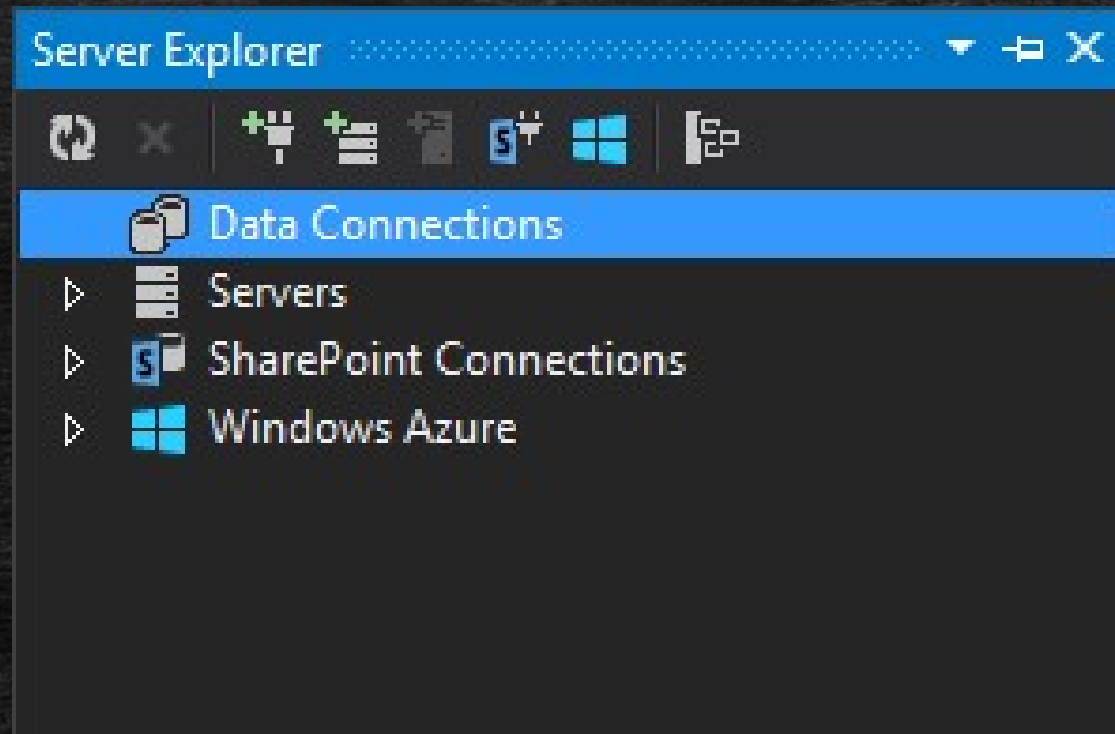
Utilisation des explorateurs et fenêtres

- Explorateur de projets.
- Explorateur de serveurs.
- Explorateur TFS.
- Boîte d'outils.
- Fenêtre de propriétés.
- Fenêtre des erreurs.
- Fenêtre des tâches.

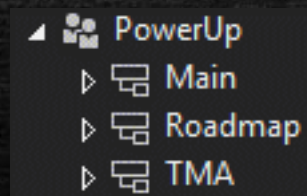
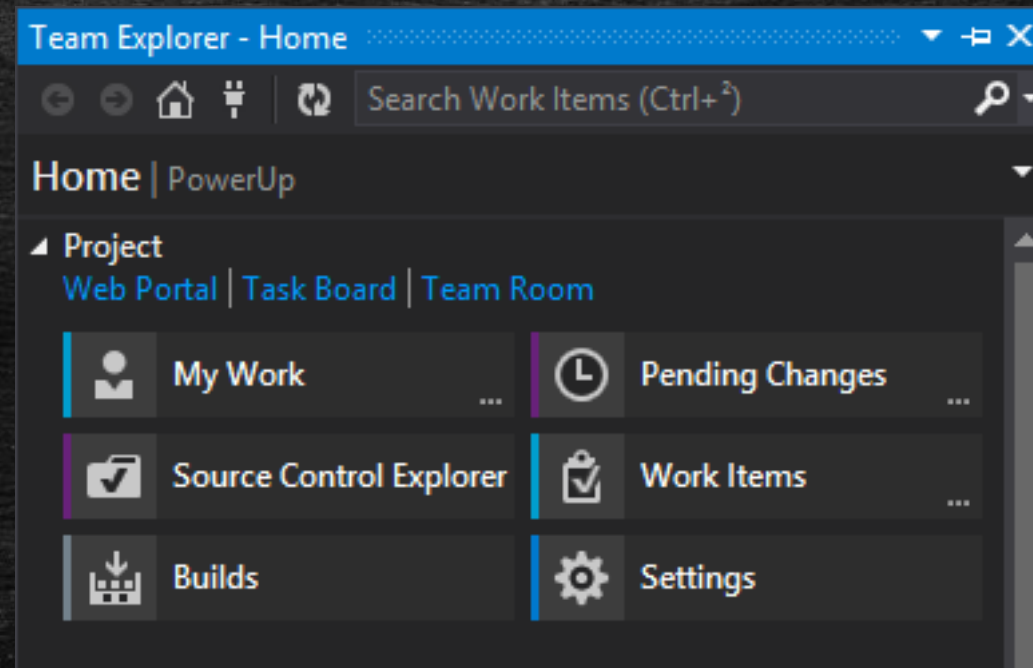
Explorateur de solutions



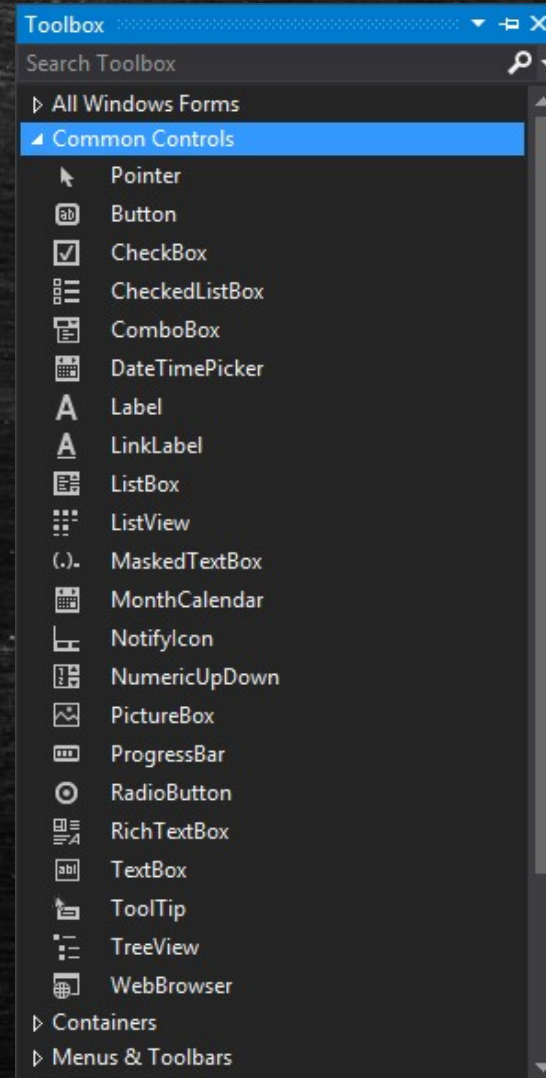
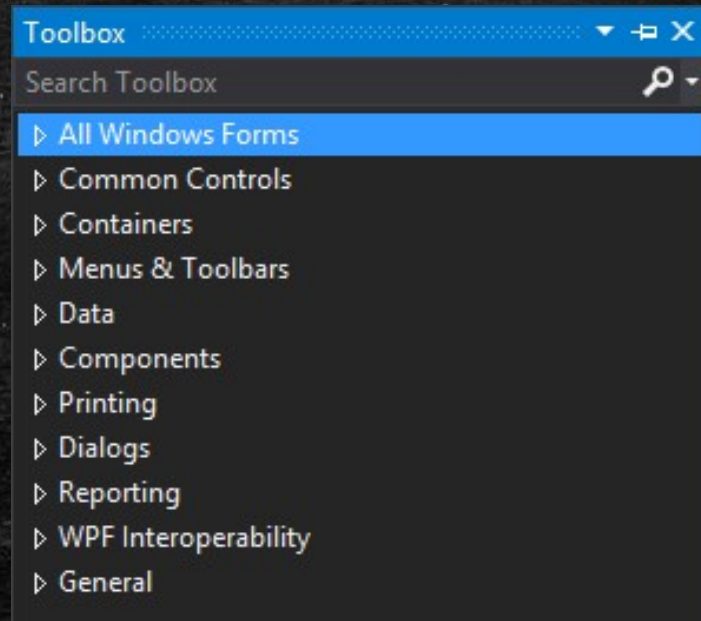
Explorateur de serveurs



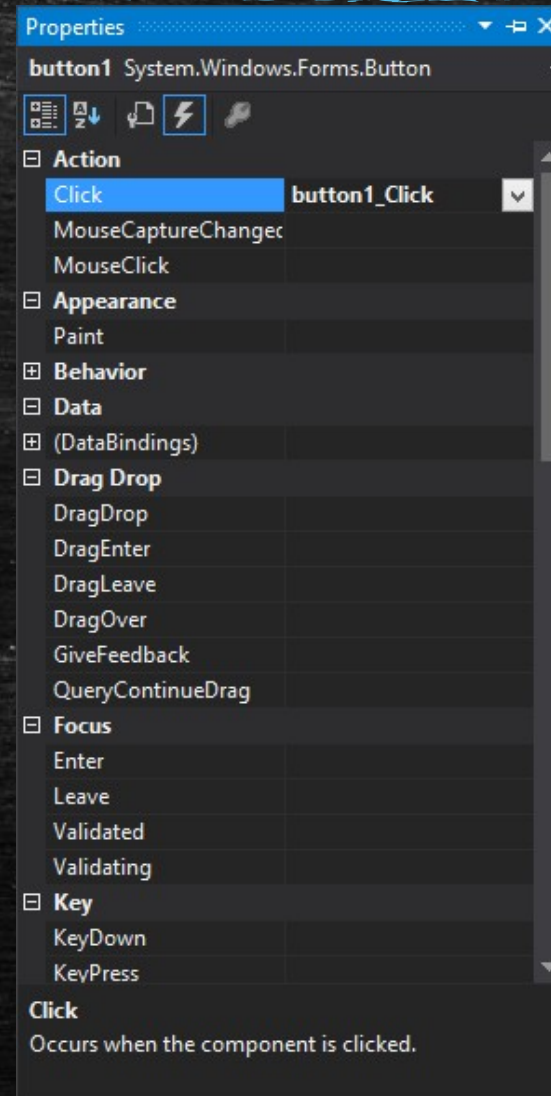
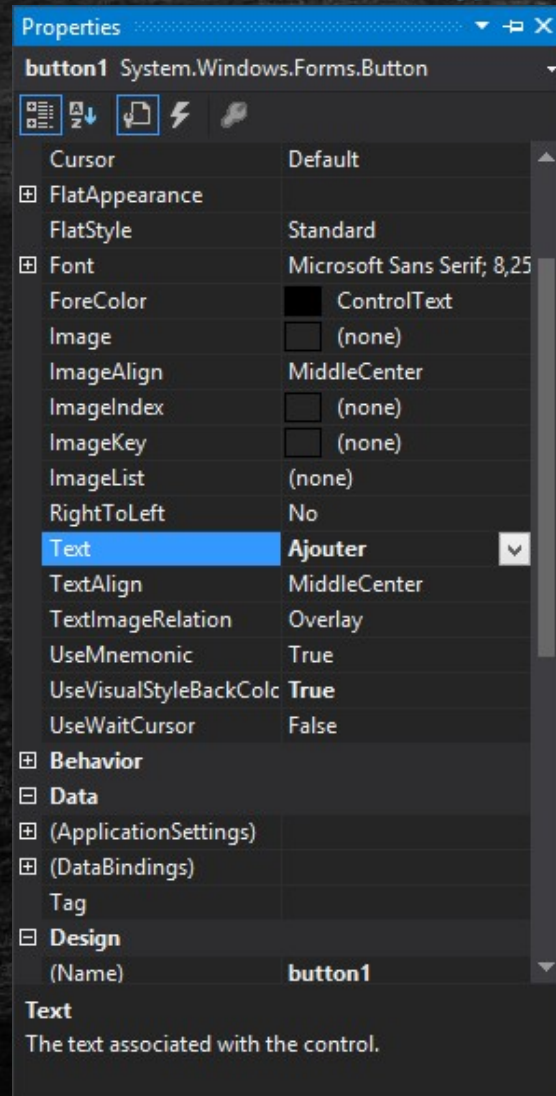
Explorateur TFS



Boite d'outils



Fenêtre de propriétés



Fenêtre des erreurs

Error List					▼	⌵	✕
▼					7 Errors		
					0 Warnings		
					0 Messages		
					Search Error List		
	Description	File ▲	Line ▲	Column ▲	Project ▲		
✕ 1	Invalid expression term 'class'	Form1.cs	22	22	WindowsFormsApplication		
✕ 2	; expected	Form1.cs	22	22	WindowsFormsApplication		
✕ 3	} expected	Form1.cs	22	22	WindowsFormsApplication		
✕ 4	} expected	Form1.cs	22	28	WindowsFormsApplication		
✕ 5	Identifier expected	Form1.cs	22	28	WindowsFormsApplication		
✕ 6	Method must have a return type	Form1.cs	22	34	WindowsFormsApplication		
✕ 7	Type or namespace definition, or end-of-file expected	Form1.cs	27	1	WindowsFormsApplication		

Fenêtre des tâches

0 references

```
private void SupprimerEtudiant(int ID)
{
    //TODO : Créer la méthode de suppression d'un étudiant.
}
```

Task List - 1 task

Comments

!	Description ▲	File ▲	Line ▲
	TODO : Créer la méthode de suppression d'un étudiant.	Form1.cs	31

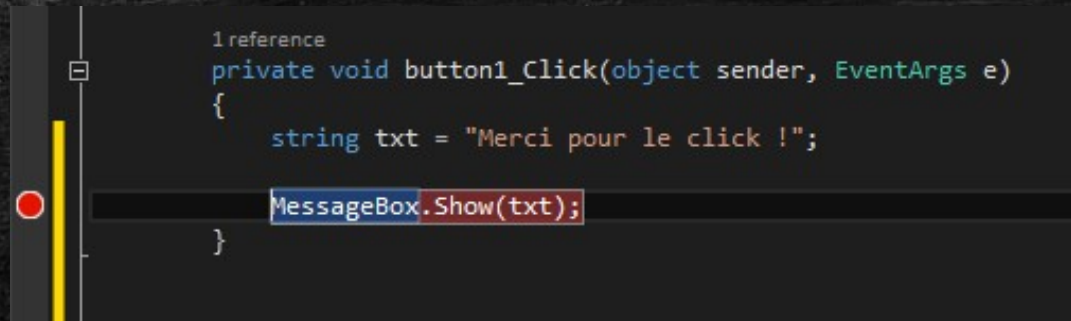
Options de compilations

- Configuration de l'exécution :
 - Mode « Debug » : fournit les informations de débogage.
 - Mode « Release » : optimise le code et la taille de l'exécutable, et ne fournit pas des informations de débogage.
- Options de l'exécution :
 - Build : ne construit que les projets qui ont changé.
 - Rebuild : reconstruit tout le projet sans vérifier les changements.
 - Clean : supprime les fichiers et répertoires intermédiaires.

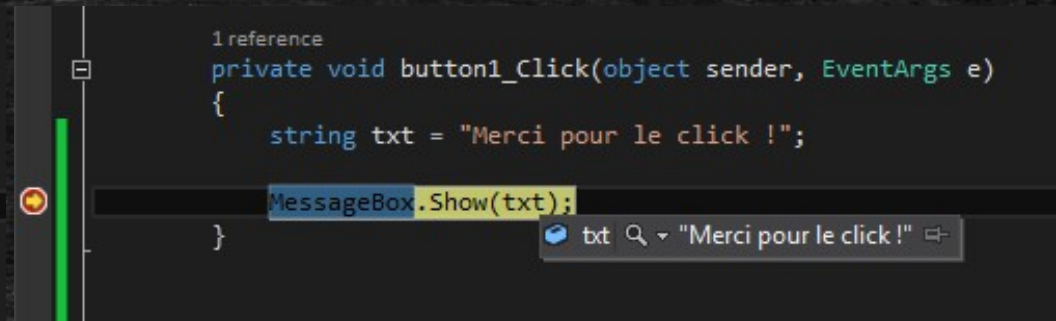
Débugage et points d'arrêts

- Positionner des points d'arrêts.
 - Débugger le code.
 - Avancer dans le code :
 - F10 : pas à pas sans rentrer à l'intérieur des méthodes.
 - F11 : pas à pas en rentrant à l'intérieur de la méthode courante.
 - Visualiseur de débogue.
- Utiliser les fenêtres des débogueur :
 - Espions.
 - Espion Express.
 - Sortie.
 - Pile d'appels.
 - Points d'arrêts.

Débugage et points d'arrêts



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string txt = "Merci pour le click !";
    MessageBox.Show(txt);
}
```



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string txt = "Merci pour le click !";
    MessageBox.Show(txt);
}
```

txt "Merci pour le click !" ⇄