**Créer un projet d’installation d’une solution WPF**

# Projet principal

## Créer le projet WPF (Net Core pour pouvoir utiliser Entity Framework)

Une image contenant texte

Description générée automatiquement

## Mettre en place le projet complet

# Projet Installer

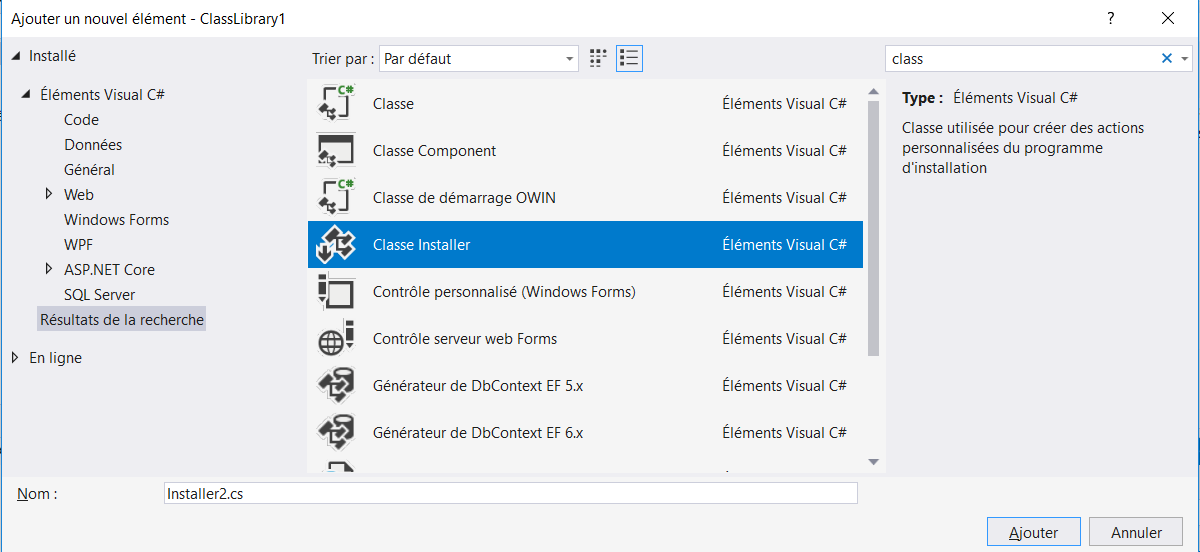
## Créer un nouveau projet de type bibliothèque de classes

Une image contenant texte

Description générée automatiquement

Choisir le .Net Framework

## Ajouter un class installer à la bibliothèque



## Ajouter le NuGet correspondant

Une image contenant texte

Description générée automatiquement

## Override les méthodes que l’on souhaite implémentée

Exemple Install

public override void Install(IDictionary stateSaver)

{

base.Install(stateSaver);

string param = this.Context.Parameters["base"];

File.WriteAllText("C:\\Users\\1701871\\Desktop\\test.txt", param);

}

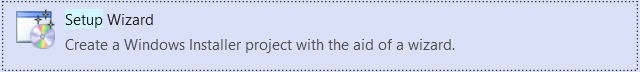
Base.install permet d’appeler la méthode mère qui fait l’installation

This.Context.Parameters : permet de récupérer les paramètres qui seront saisis dans l’interface Utilisateur que l’on mettra en place au moment du setup.

["base"] correspond au nom que l’on donnera à la récupération des zones de saisie dans la propriété CustomActionData de actions personnalisées

# Projet Setup

## Ajouter un nouveau projet à la solution de type Setup Wizard



S’il n’est pas disponible, ajouter l’extension Microsoft Visual Studio Installer Projects

## Choisir le type Windows Application

Une image contenant texte

Description générée automatiquement

## Sélectionner les sorties nécessaires

Sortie Principale et Publish Item pour le projet principal

Une image contenant texte

Description générée automatiquement

Et

Sortie Principal pour la bibliothèque de classe

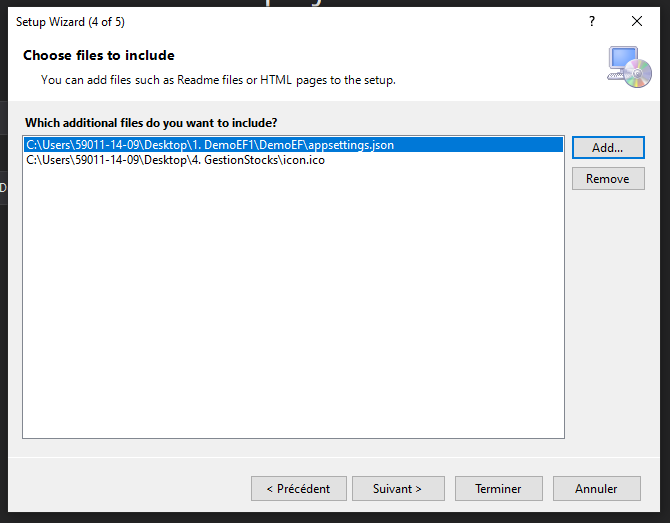


Puis cliquer sur « suivant »

## Ajouter des fichiers optionnels au projet

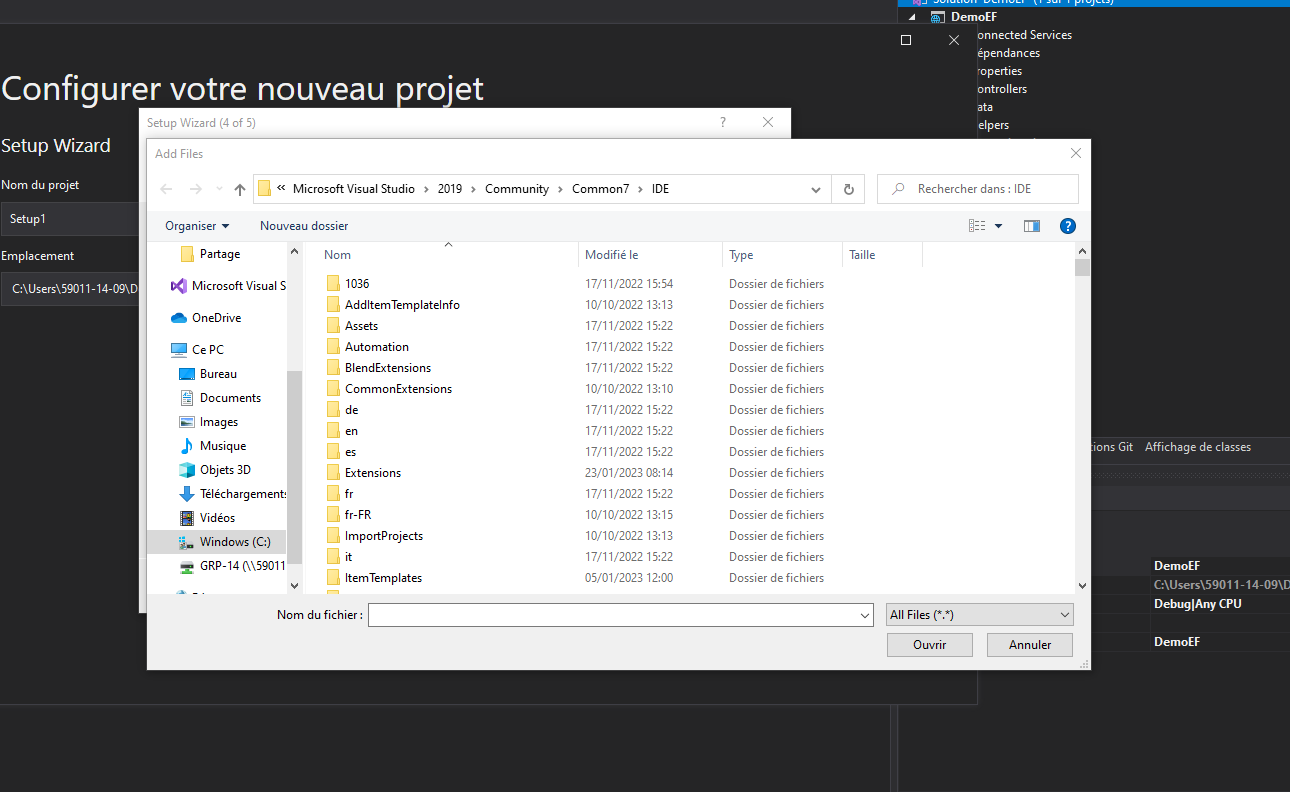
### Ajout de fichiers lors de la création du projet Setup Wizard

Ici on peut ajouter les fichiers utiles pour le projet :



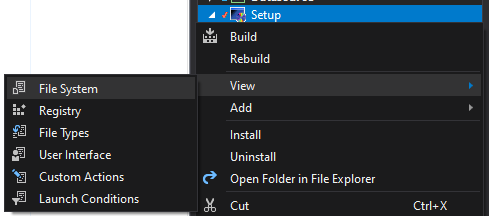
(Ici les 2 fichiers affichés ne sont donnés qu’à titre d’exemple)

Cliquer sur le Bouton « Add » ouvrant une nouvelle fenêtre permettant de sélectionner les fichiers voulus.

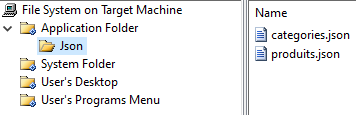


### Ajout de fichiers après la création du projet Setup Wizard

Il est aussi possible d’ajouter les fichiers après que le Setup soit créé



Ajouter les fichiers dans le dossier « Application Folder »

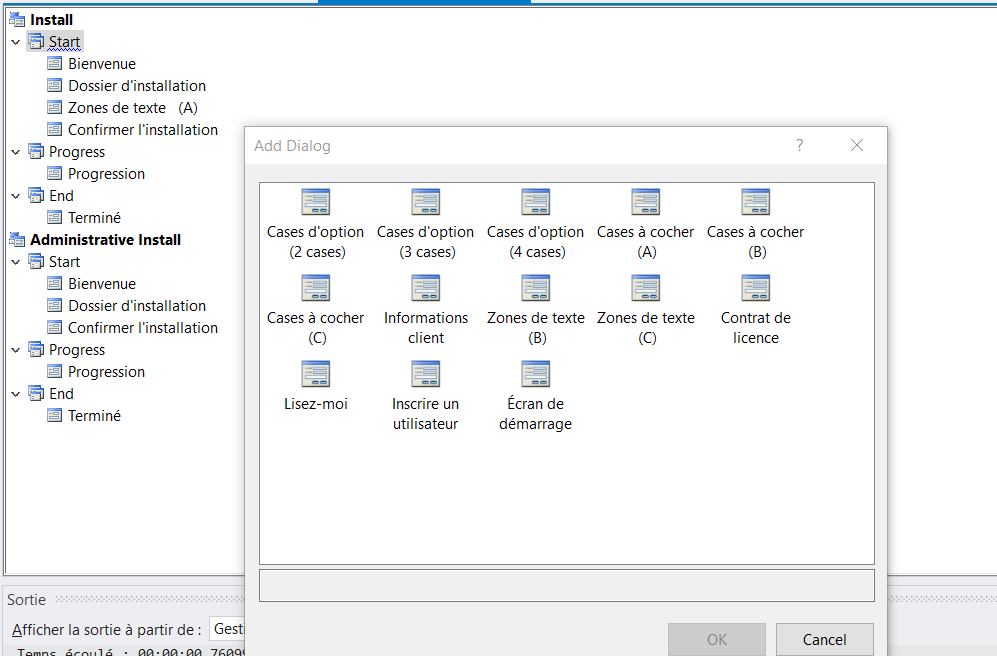


Dans chaque classe Services, changer le « PATH » (chemin) de votre fichier. Le dossier Application Folder sera la racine de votre application.

## Interface Utilisateur

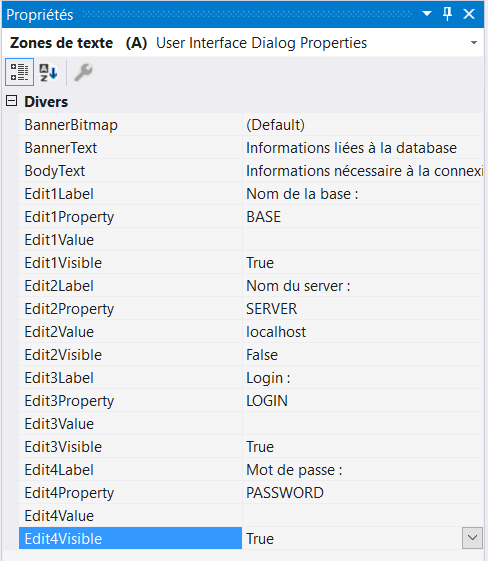
### Ouvrir la View interface Utilisateur

### Ajouter une /des boites de dialogues dans l’interface Utilisateur sur la partie Start



Utiliser les fonctions monter / descendre disponible en clic droit pour mettre les interfaces dans l’ordre

### Ajuster les propriétés des boites de dialogues



## Actions Personnalisées

### Ouvrir la View, actions personnalisées au projet

### Ajouter une action sur l’une des actions (Install par exemple)

### Sélectionner la sortie principale du projet bibliothèque de classe

### Mettre à jour les propriétés de cette action

Une image contenant texte

Description générée automatiquement

/base=[BASE]/server=[SERVER] : permet de faire correspondre les zones de saisie de l’interface (ex BASE) avec les entrées du tableau Parameters du Context (ex Context.Parameters["base"])

Celui-ci sera utilisé dans l’installer de la classe Installer dans le projet bibliothèque de classes

Condition : permet de mettre des conditions à l’exécution (ex : ChechBox1= 1)

# Ajustement du setup

## Installer dans un dossier

* Dans la View Système de fichiers, créer un dossier dans le dossier Application Folder
* Mettre à l’intérieur toutes les sorties.

## Créer les raccourcis

* Créer des raccourcis dans User’s Desktop et dans User’s Programs Menu.
* Ces raccourcis peuvent être conditionnés par l’utilisation d’une checkbox dans l’interface utilisateur