Projet

Compléments au tutoriel « Apprendre à créer un installeur avec Visual Studio Installer Projects »

Contenu

Le tutoriel initial

Il décrit la réalisation pas à pas d'un setup basique. Un wizard dédié étant le fil rouge. L'objet du présent complément est de rajouter des personnalisations supplémentaires simples ou d'effectuer d'autres actions, avec les éditeurs de Visual Studio Project. Les éditeurs ne sont pas évoqués, dans le tutoriel initial, car certains sont gérés automatiquement par le wizard et sont de ce fait

transparents (par exemple l'interface utilisateur par défaut ou les conditions de lancement minimales). Et d'autres ne sont pas nécessaires pour un setup basique (modification du registre, actions personnalisées etc…).

Ce complément vous apprendra à personnaliser davantage les installeurs Windows (setup.msi). Par exemple :

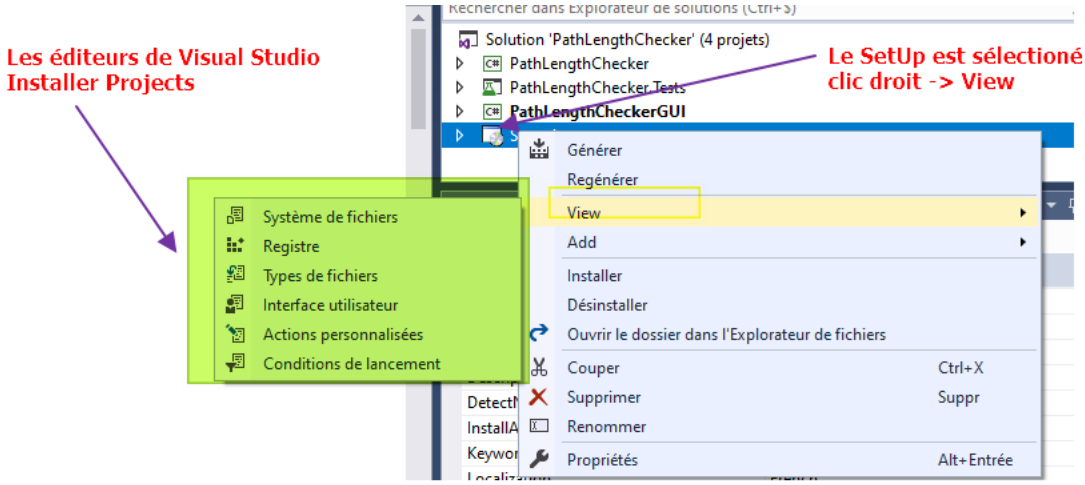
- Modifier / compléter l'interface utilisateur :
  - Visualiser un fichier lisezmoi ;
  - Visualiser un fichier licence ;
  - Visualiser un splash bitmap au lancement de l'installeur;
  - Modifier les messages par défaut (ou tout simplement les traduire).
- Ajouter des clés dans la base de registre ;
- Associer un type de fichier au programme qui va être installé ;
- Compléter les prérequis pour l'installation ;

Mais aussi, il indique sommairement comment aller plus loin, avec Visual Studio Installer Projects, en utilisant les ***actions personnalisées*** (custom Actions)

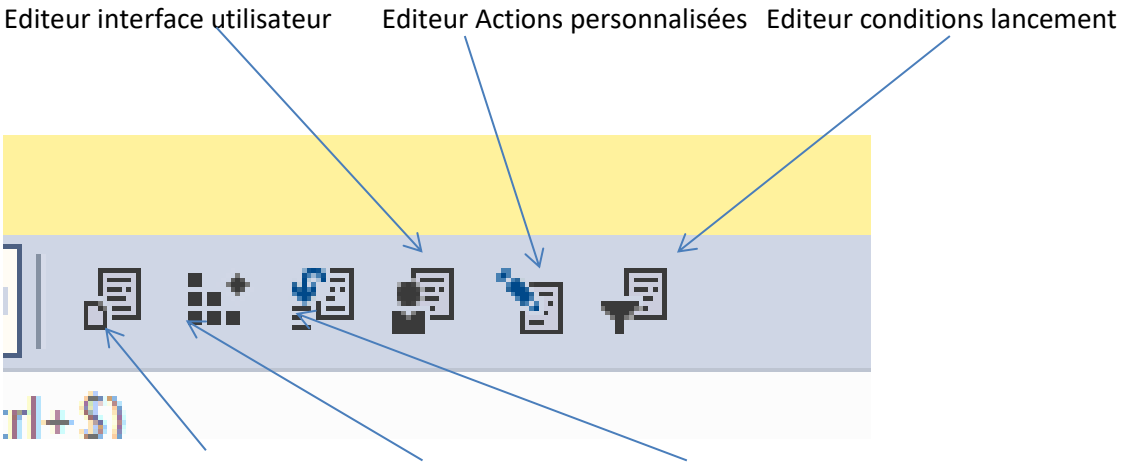
L'utilisation de la plupart des éditeurs est décrite ci-dessous d'une façon suffisamment détaillée pour pouvoir les utiliser facilement, à l'exception de l'éditeur d' « Actions personnalisées » qui est simplement survolé car plus complexe à mettre en œuvre et pas du niveau débutant. Pour ce dernier l'objectif est de « savoir que ça existe ».

Les éditeurs de Visual Studio Project Installer :

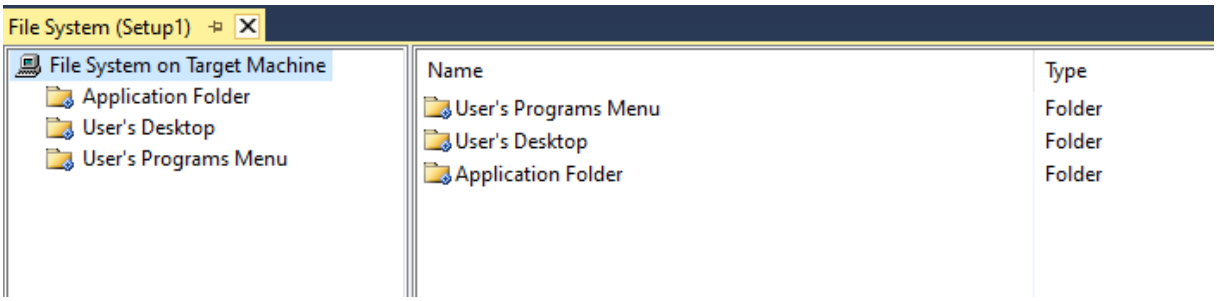
On y accède par un clic droit sur le projet, dans l'explorateur de solution ou par les icones qui sont disponibles, toujours dans l'explorateur de solutions, lorsque le setup est sélectionné.



Les icônes :



Editeur du Système de fichiers



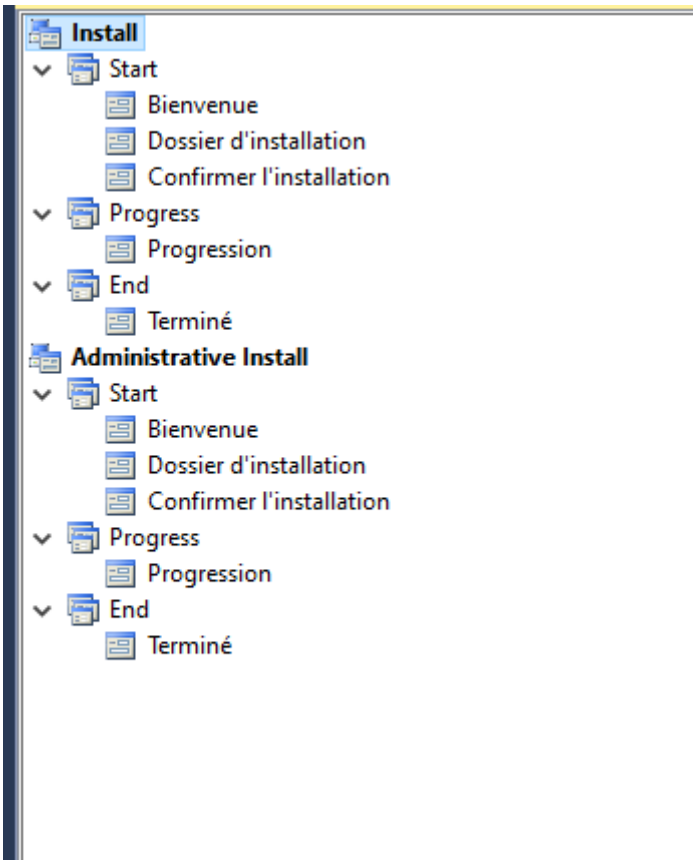
On a utilisé cet onglet dans le tutoriel. Pour les personnalisations décrites ci-après il faudra ajouter (dans ***Application Folder***) les fichiers nécessaires aux personnalisations (lisezmoi, licence, splashBitmat etc…)

Rappel : pour ajouter un fichier, clic droit sur Application Folder -> Add -> fichier

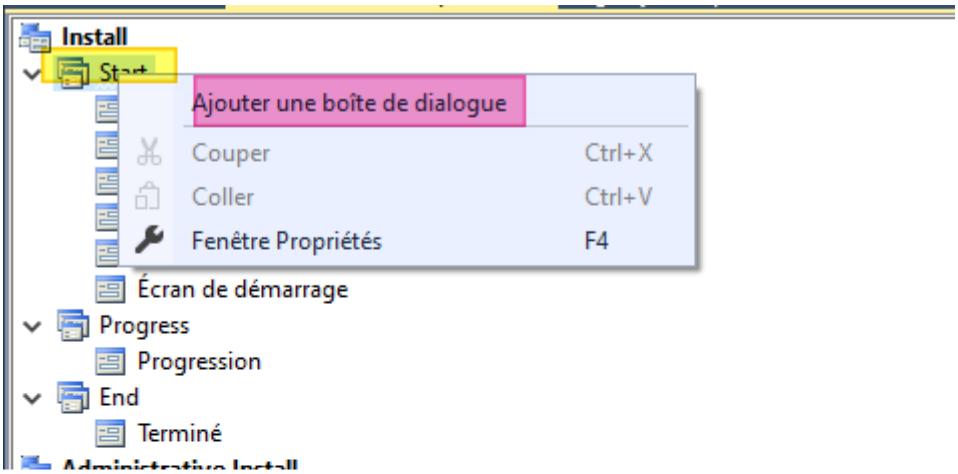
Editeur de l'interface utilisateur

C'est elle qui génère ce qui va être affiché lors de l'installation. Dans le tutoriel cet éditeur n'ait jamais évoqué puisqu'on a utilisé l'interface utilisateur par défaut générée par le Wizard. Mais on peut à l'aide de cet éditeur modifier ou compléter les messages. Par exemple afficher sur chaque écran le logo de l'éditeur, indiquer le nom du correspondant informatique à contacter en cas de difficultés avec l'installation, visualiser un fichier lisezmoi, un fichier licence.

Interface utilisateur après exécution du wizard :

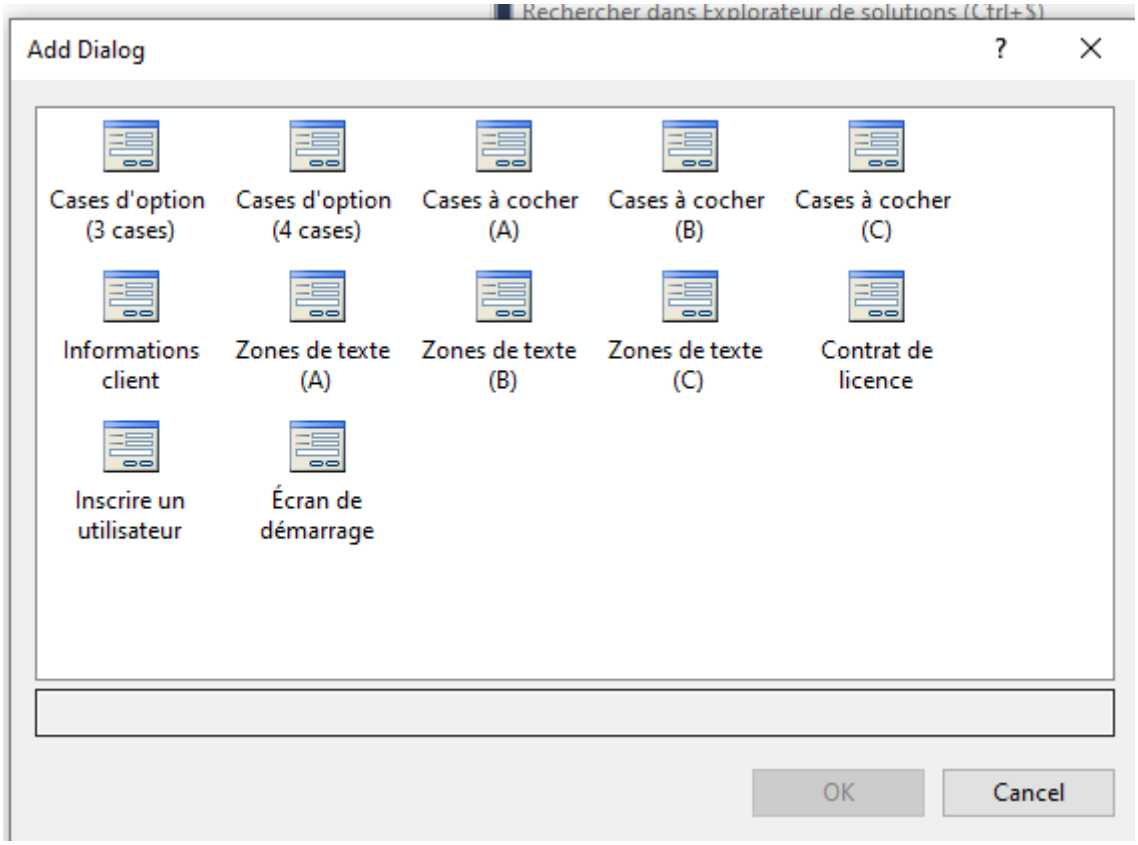


Ajouts d’écrans supplémentaires :



Pour ajouter des personalisations : cli droit sur Start, ou Progress ou End etc...

Choisir ensuite la(es) boite(s) de dialogue

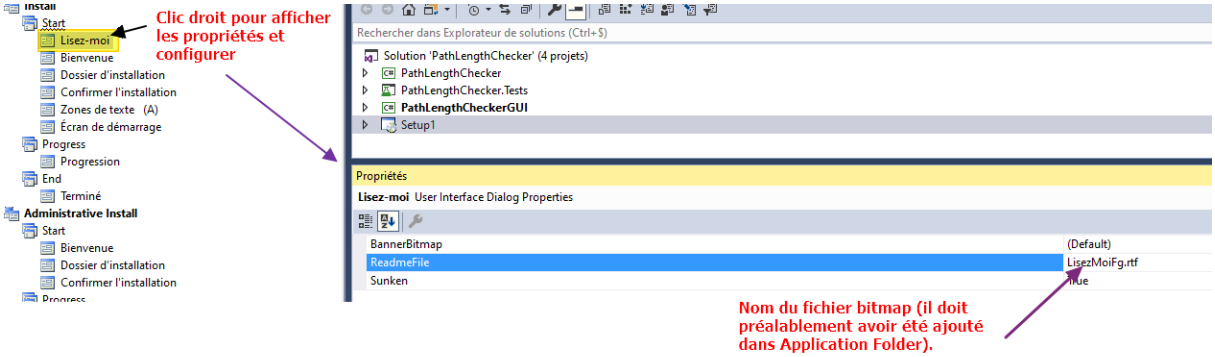


Nota 1 : Ecran de démarrage = splash bitmap

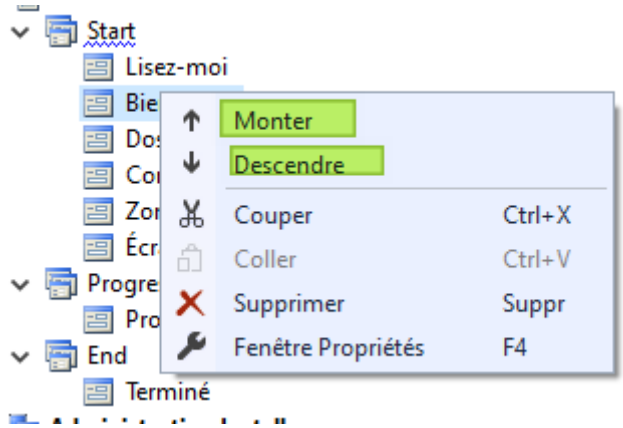
Nota 2 : chaque boite de dialogue n'est disponible qu'une fois. Quand elle a été utilisée elle disparait du choix proposé.

Apres avoir ajouté une boite de dialogue, il faut la configurer dans sa fenêtre propriétés (nom de fichier à visualiser, texte(s) à afficher etc...).

On peut également modifier le texte des boites de dialogue crées par le setup, par exemple le texte de la boite de dialogue « Bienvenue » est par défaut *« The installer will guide you through the steps required to install [ProductName] on your computer. »* on peut simplement traduire.



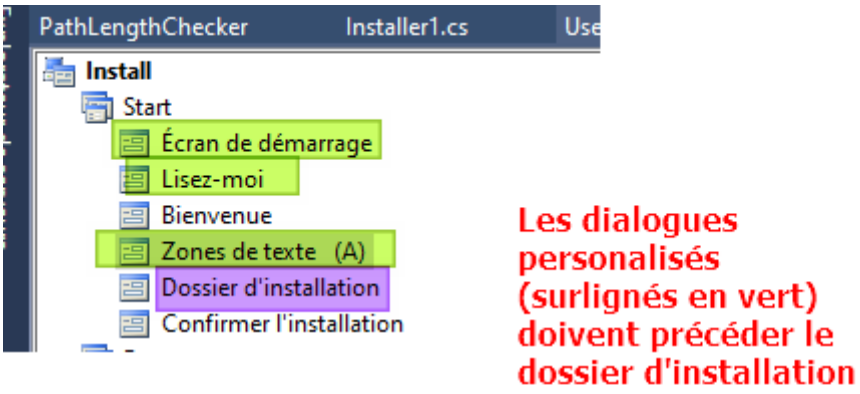
On peut aussi modifier l'ordre des boites de dialogue :



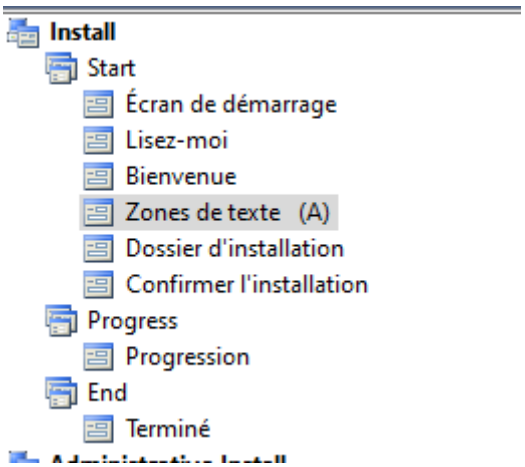
On pourrait également créer un installeur « silencieux »

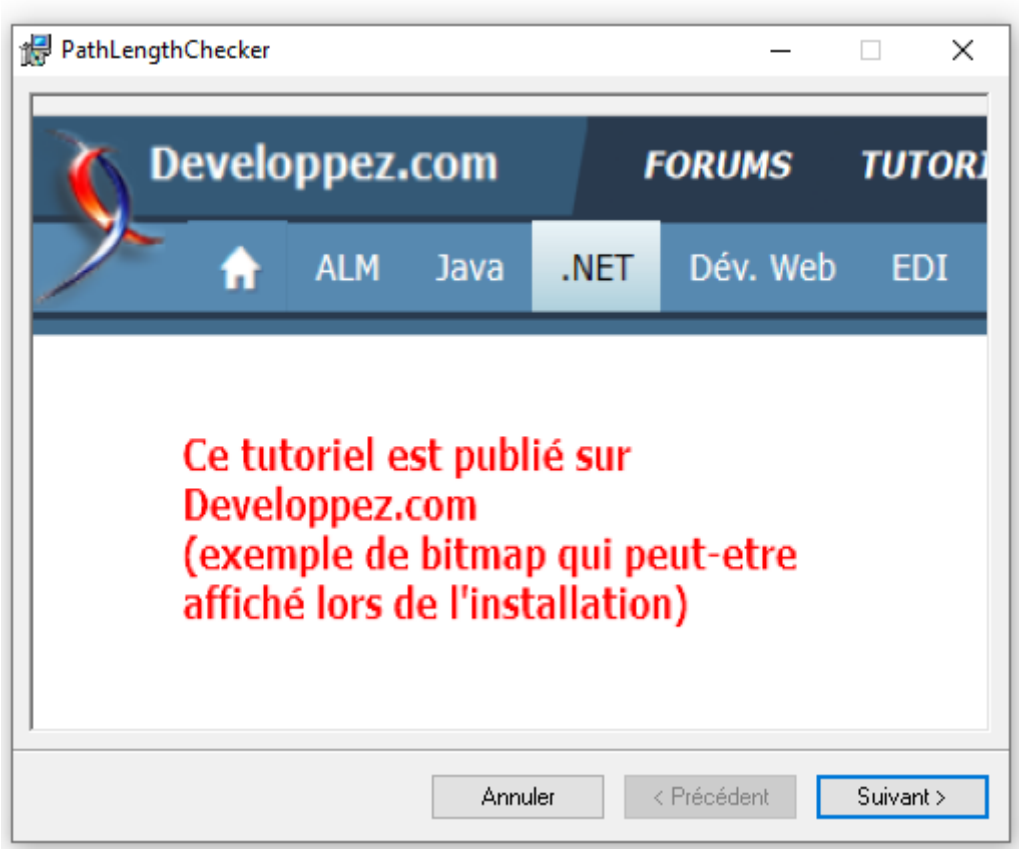
> **Tip**
>
> If you do not want to display a user interface, delete all dialog boxes from the **User Interface Editor**.

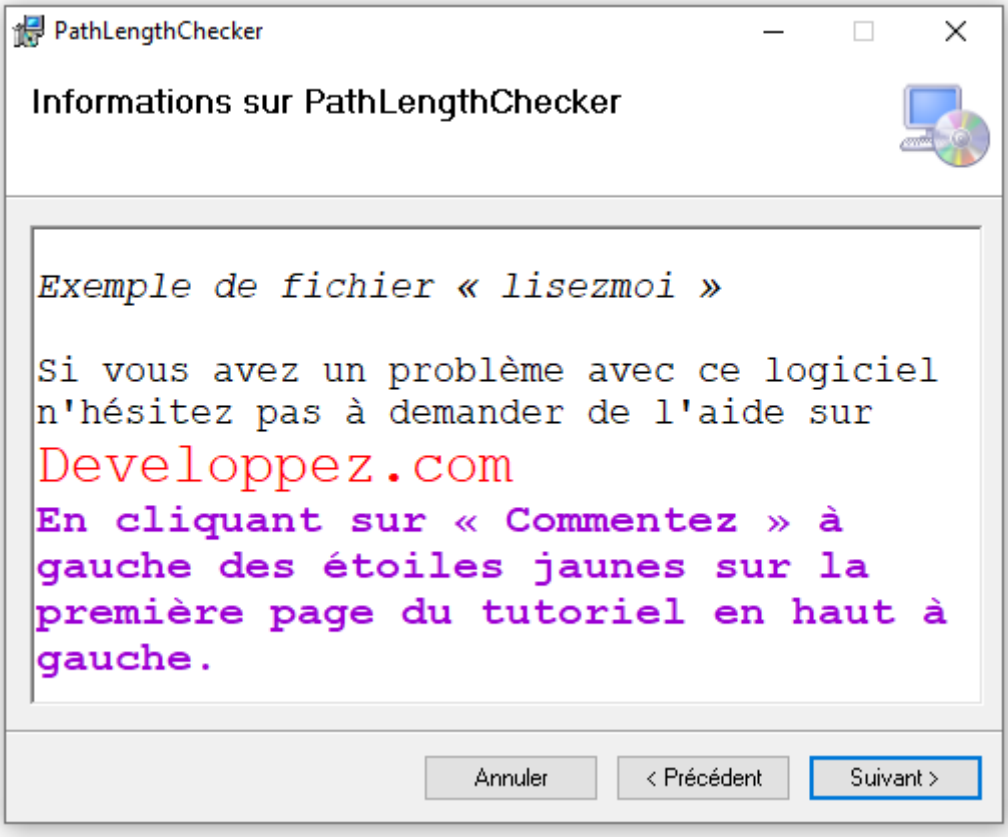
Nota :

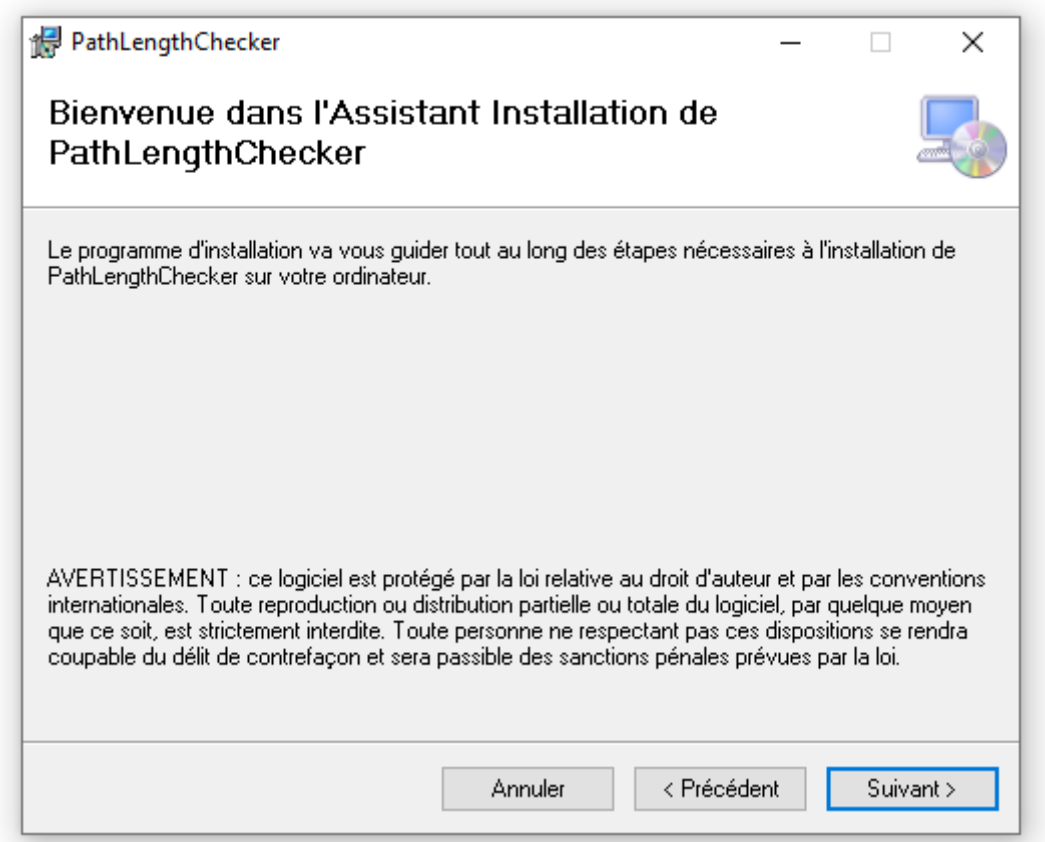


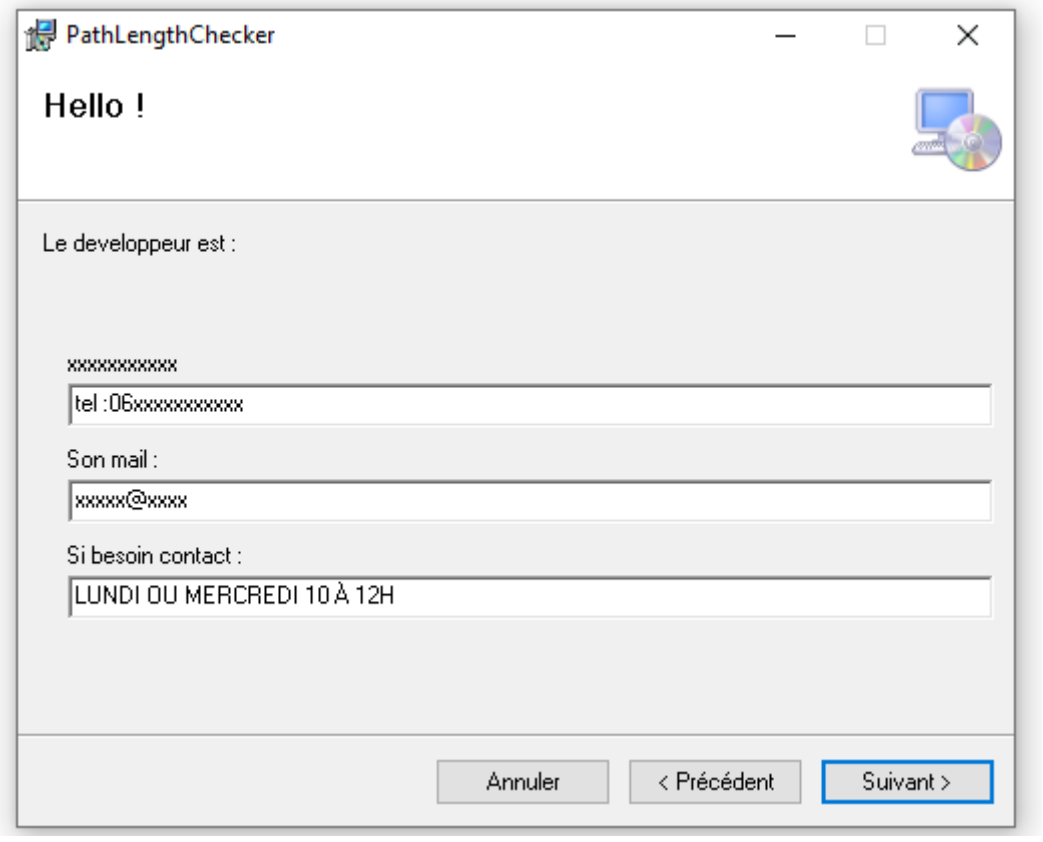
En final à l'installation l'utilisateur verra :

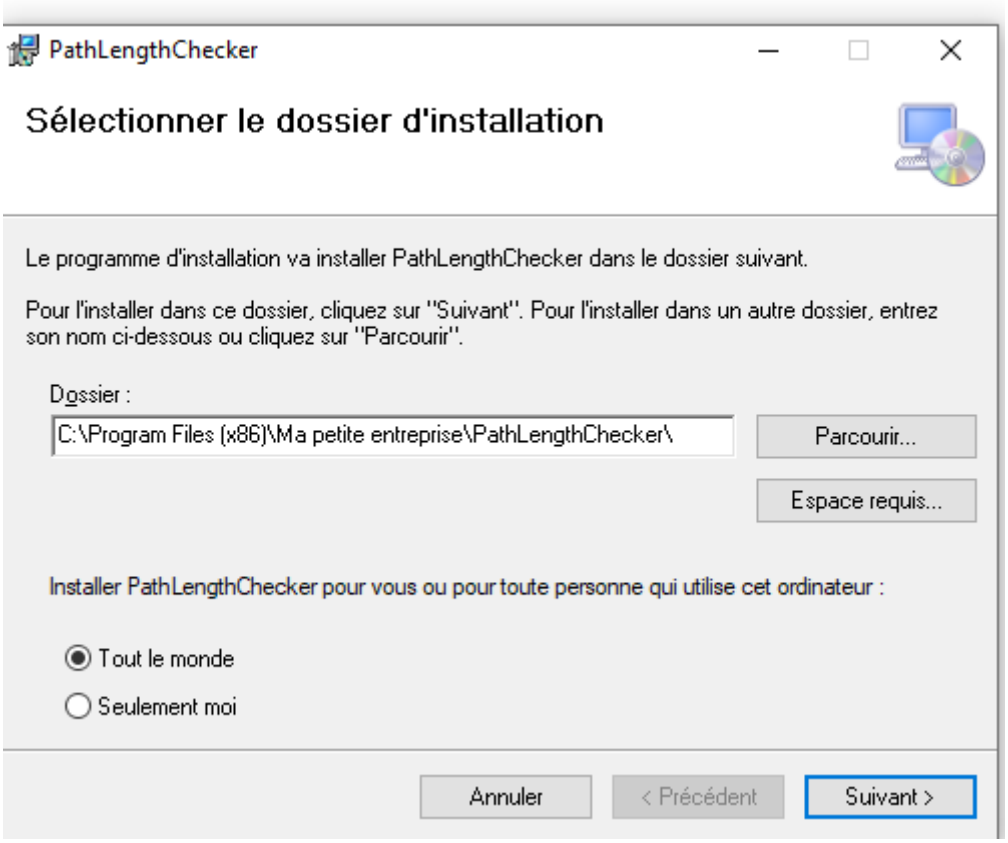



PathLengthChecker
Developpez.com
FORUMS
ALM
Java
.NET
Dév. Web
EDI
Ce tutoriel est publié sur
Developpez.com
(exemple de bitmap qui peut-etre
affiché lors de l'installation)
Annuler
< Précédent
Suivant >


PathLengthChecker
Informations sur PathLengthChecker
Exemple de fichier « lisezmoi »
Si vous avez un problème avec ce logiciel
n'hésitez pas à demander de l'aide sur
Developpez.com
En cliquant sur « Commentez » à
gauche des étoiles jaunes sur la
première page du tutoriel en haut à
gauche.
Annuler
< Précédent
Suivant >


PathLengthChecker
Bienvenue dans l'Assistant Installation de PathLengthChecker
Le programme d'installation va vous guider tout au long des étapes nécessaires à l'installation de PathLengthChecker sur votre ordinateur.
AVERTISSEMENT : ce logiciel est protégé par la loi relative au droit d'auteur et par les conventions internationales. Toute reproduction ou distribution partielle ou totale du logiciel, par quelque moyen que ce soit, est strictement interdite. Toute personne ne respectant pas ces dispositions se rendra coupable du délit de contrefaçon et sera passible des sanctions pénales prévues par la loi.
Annuler
< Précédent
Suivant >

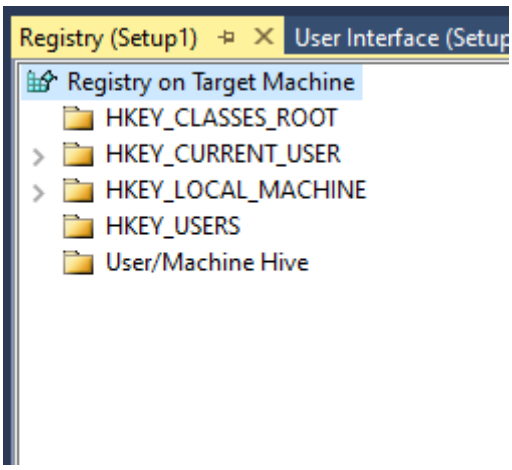

PathLengthChecker
Hello !
Le developpeur est :
xxxxxxxxxx
tel :06xxxxxxxxxx
Son mail :
xxxxx@xxxx
Si besoin contact :
LUNDI OU MERCREDI 10 À 12H
Annuler
< Précédent
Suivant >



Puis l'UAC puis D° dans le tutoriel

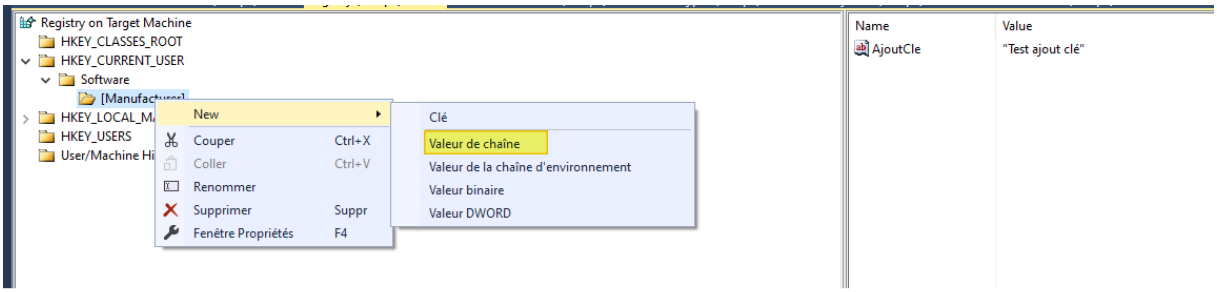
Editeur du Registre

Permet de créer très simplement une(des) clé(s) de registre à l'installation. On peut choisir de les effacer à la désinstallation du programme ou pas (pour savoir à la prochaine installation que le programme a déjà été installé sur le PC).



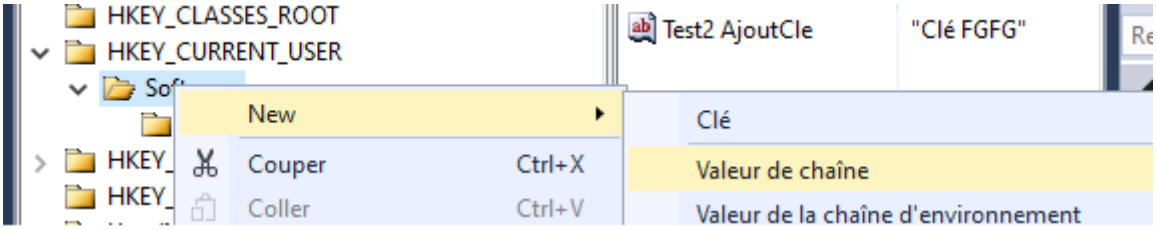
Exemple

Ajout d'une clé dans [Manufacturer]. Cette valeur correspond à celle qui a été indiquée au paragraphe 4.2.1 du tutoriel en l'occurrence « Ma petite entreprise »

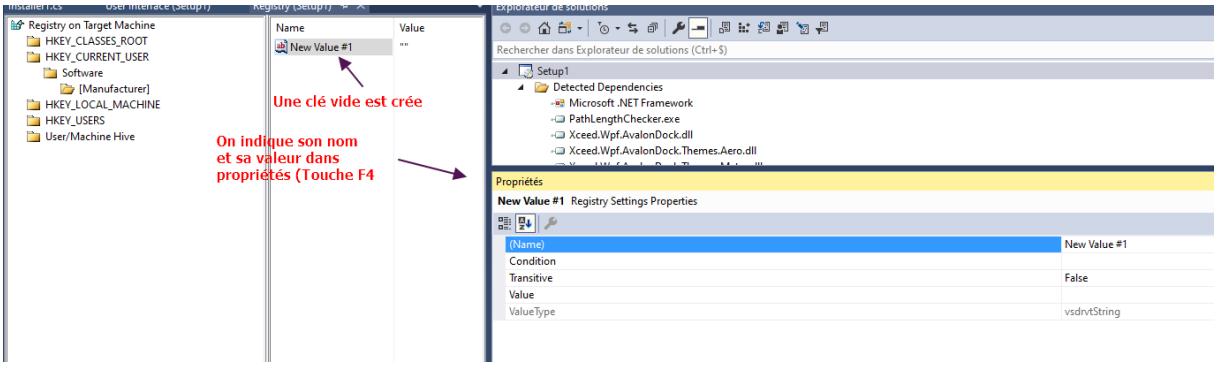


Ajout d'une clé dans HCKU Software

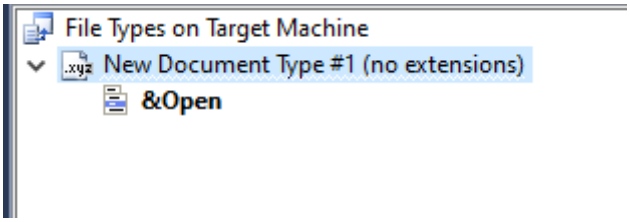
Etape 1



Ajout d'une clé dans HKCU



Editeur du Type de fichier

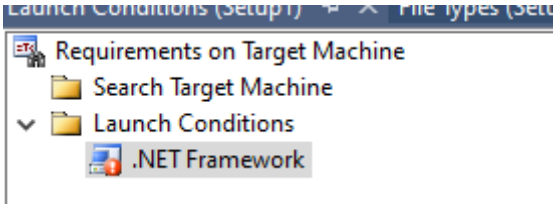


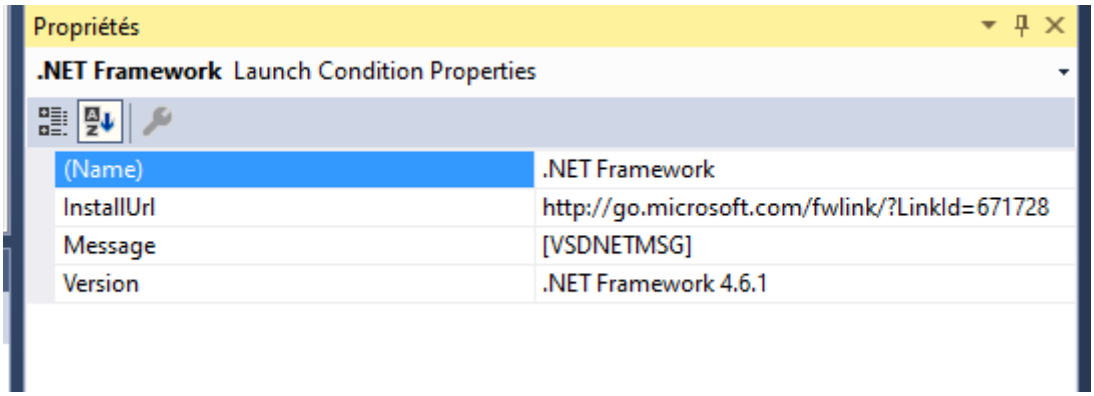
Permet d'associer un type de fichier à l'exécutable qui est installé (de la même façon qu'un fichier .docx si on double clic s'ouvrira ace Word) . Par exemple pour PathLengthChecker un fichier de configuration qui lorsqu'on double clique dessus lance PathLengthChecker , configure le root directory, indique si on recherche ou pas dans les subdirectory (bien sûr il faudrait rajouter du code dans PathLengthChecker pour gérer le fichier de configuration) .

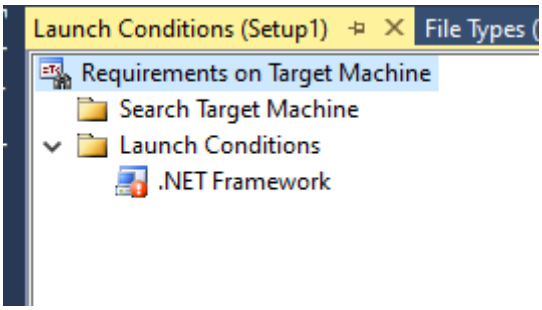
Editeur des Conditions de lancement (Launch conditions)

Dans le tuto on n'a pas utilisé cet onglet puisque le Wizard a automatiquement rajouté ce qui est à coup sûr obligatoire le .netFramework (et sa version). On pourrait rajouter une version minimum de Windows, la présence de clés de registre, de fichiers etc…

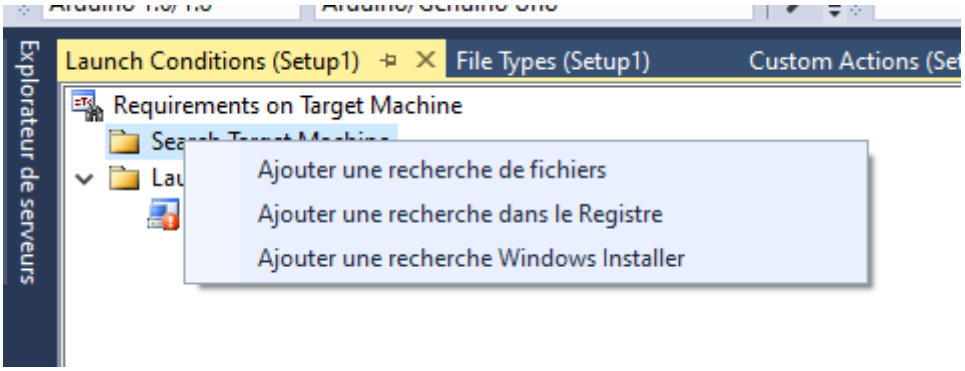
La condition de lancement crée par le wizard :







Les conditions de lancement que l'on peut rajouter.



Editeur d'actions personnalisées ou comment aller plus loin avec Visual Studio Installer Projects

Comme déjà indiqué, ce paragraphe n'est qu'un aperçu généraliste.

Source : message privé de Goldbergg (repris ici avec son accord).

Les **actions personnalisées (custom action)** permettent de répondre à pratiquement tous les besoins lors d'une installation / désinstallation.

Principe :

Il faut rajouter dans les projet de type Winform, WPF, Service, Library un fichiers de type "Classe Installer" (https://docs.microsoft.com/fr-fr/dot...tframework-4.8) qui permet de créer toute sorte d'action a différent moment du processus d'installation :

Via des event :

- BeforeInstall

- Committing
- AfterUninstall
- AfterRollback
- AfterInstall
- Committed
- BeforeRollback
- BeforeUninstall

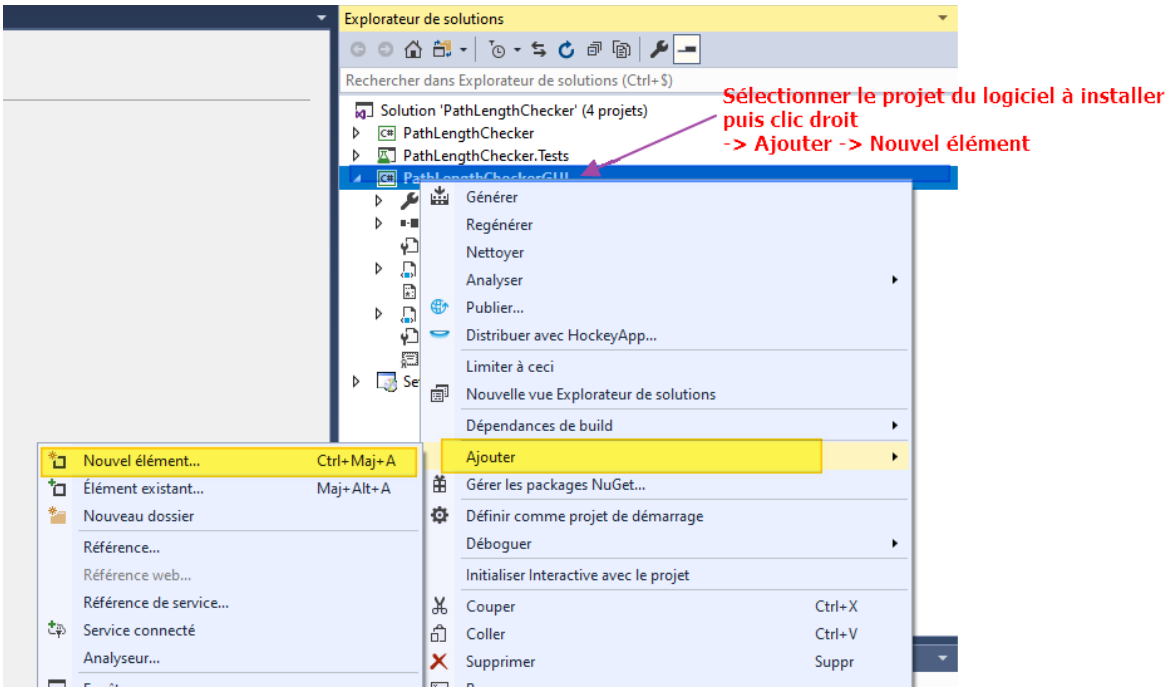
Ou via réimplémentation de méthode héritée :

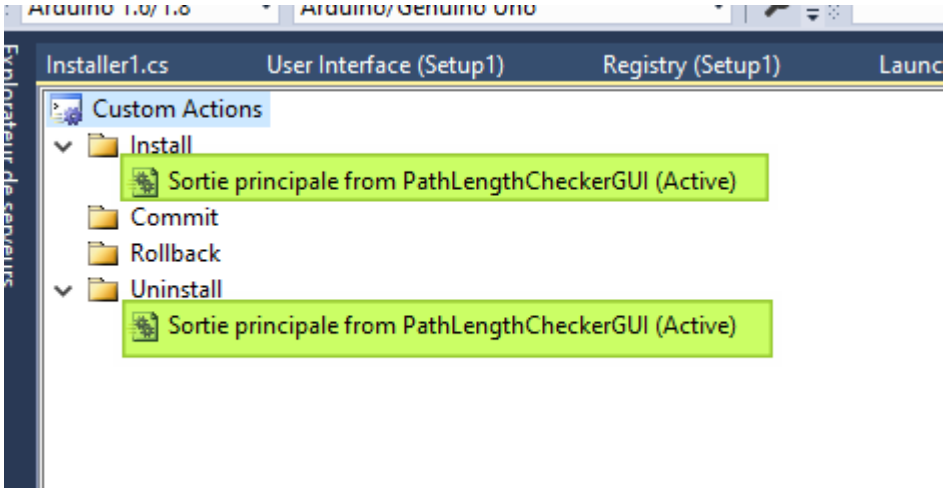
- Commit
- Install
- Rollback
- Uninstall

Une fois ces classes créé dans chaque projet il suffit de les référencer dans le projet Installer via "les actions personnalisées" (clic droit sur le nom du projet => view => actions personnalisées).

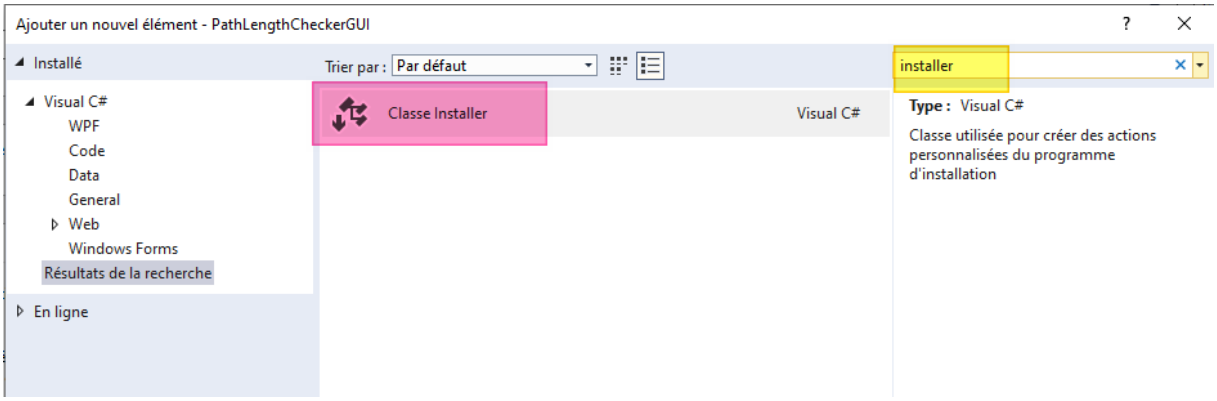
De cette façon il est par exemple possible de créer un projet d'installation qui contient deux Service Windows et une GUI Winform, l'installateur install/désinstall bien les deux services, les lances, puis lance la GUI.

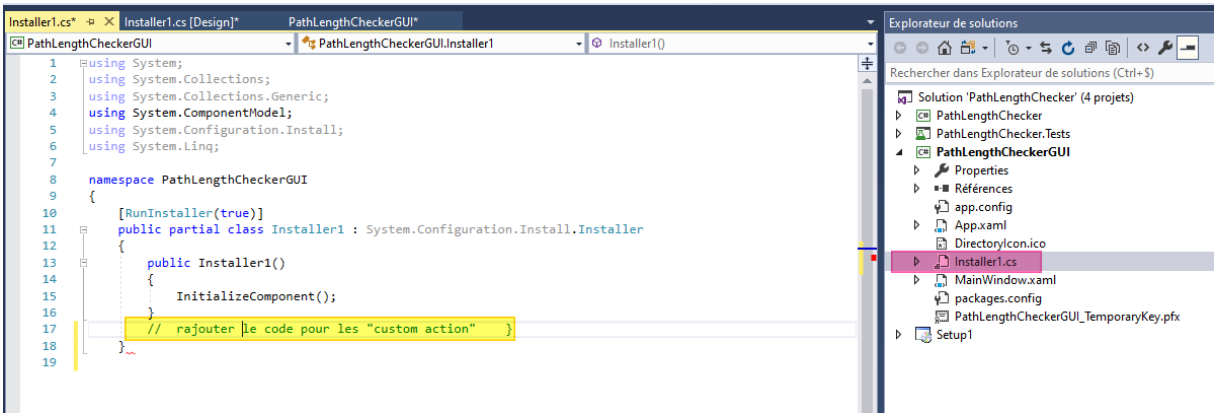
Comment rajouter « Class installer » à un projet




Installer1.cs
User Interface (Setup1)
Registry (Setup1)
Custom Actions
Install
Sortie principale from PathLengthCheckerGUI (Active)
Commit
Rollback
Uninstall
Sortie principale from PathLengthCheckerGUI (Active)

Comment


Ajouter un nouvel élément - PathLengthCheckerGUI
Installé
Trier par : Par défaut
installer
Visual C#
WPF
Code
Data
General
Web
Windows Forms
Résultats de la recherche
En ligne
Classe Installer
Visual C#
Type : Visual C#
Classe utilisée pour créer des actions personnalisées du programme d'installation


Installer1.cs*
Installer1.cs [Design]*
PathLengthCheckerGUI*
PathLengthCheckerGUI
PathLengthCheckerGUI.Installer1
Installer1()
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration.Install;
using System.Linq;
namespace PathLengthCheckerGUI
{
[RunInstaller(true)]
public partial class Installer1 : System.Configuration.Install.Installer
{
public Installer1()
{
InitializeComponent();
}
// rajouter le code pour les "custom action" }
}
Explorateur de solutions
Rechercher dans Explorateur de solutions (Ctrl+$)
Solution 'PathLengthChecker' (4 projets)
PathLengthChecker
PathLengthChecker.Tests
PathLengthCheckerGUI
Properties
Références
app.config
App.xaml
DirectoryIcon.ico
Installer1.cs
MainWindow.xaml
packages.config
PathLengthCheckerGUI_TemporaryKey.pfx
Setup1