

## Méthodes de déploiement

---

### Sommaire

Méthodes de déploiement.....	1
1 Introduction.....	2
2 Gestion du ClickOnce.....	2
2.1 Qu'est ce que le ClickOnce.....	2
2.2 Les éléments du ClickOnce .....	4
2.2.1 Signature.....	4
2.2.2 Sécurité.....	5
2.2.3 Publication.....	7
2.2.4 Configurer l'installation .....	9
3 Gestion du Projet d'installation.....	10
3.1 Qu'est ce que le Projet d'installation.....	10
3.2 Créer un Projet d'installation.....	11
3.3 Les éditeurs du Projet d'installation .....	11
3.4 Utiliser un Projet d'installation .....	13
3.4.1 Propriétés du Projet d'installation.....	13
3.4.2 L'éditeur Fichier Système .....	16
3.4.3 L'éditeur Actions Personnalisées.....	23
4 Conclusion .....	25

## 1 Introduction

Le déploiement est un système permettant de transférer facilement, rapidement et en plus ou moins grande quantité une application. Pour cela Visual Studio propose deux méthodes adaptées à vos besoins : le ClickOnce et le Projet d'installation.

Le ClickOnce est un moyen de déploiement simple et court d'utilisation mais permettant moins de configuration que le Projet d'installation. Il est plus adapté à un petit déploiement dans un environnement maîtrisé.

Le Projet d'installation, à l'inverse, est très configurable et est plus approprié à un déploiement de grande échelle.

Dans ce chapitre, nous allons vous présenter comment utiliser ces deux outils.

Bon cours .NET !

L'équipe *Windows Form*.

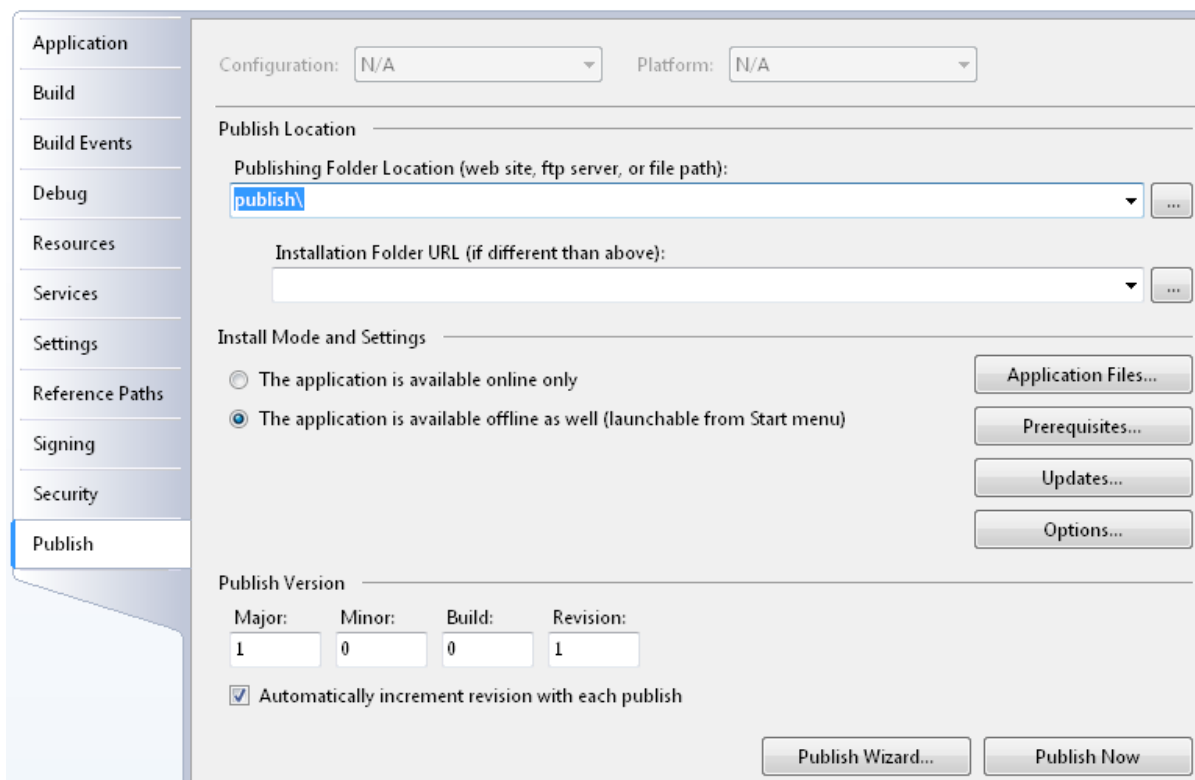
## 2 Gestion du ClickOnce

### 2.1 Qu'est ce que le ClickOnce

Le ClickOnce est une technologie Microsoft sortie avec le Framework 2.0. Il permet de déployer facilement vos applications avec une interface graphique agréable et facile d'utilisation. Toutes les applications Windows Form peuvent être déployées grâce au ClickOnce soit par un site Web, un CD-ROM ou un partage de fichier. Ces applications sont totalement isolées des autres applications du système, c'est-à-dire qu'elles possèdent leur propre composant et ne risque pas de créer des conflits.

De plus, cet outil permet de faire les mises à jour nécessaires au bon fonctionnement ou à l'amélioration d'une application. Ces mises à jour se feront automatiquement avant ou après le démarrage d'une application selon le mode que vous aurez choisi. Les mises à jour faites grâce au ClickOnce sont transactionnelles, c'est-à-dire qu'elles se feront entièrement ou ne se feront pas.

Vous pouvez accéder et gérer le ClickOnce en faisant clique droit dans le Solution Explorer puis en cliquant sur « Propriétés ».



Configuration: N/A Platform: N/A

**Publish Location**

Publishing Folder Location (web site, ftp server, or file path):  
publish\

Installation Folder URL (if different than above):

**Install Mode and Settings**

☐ The application is available online only

☒ The application is available offline as well (launchable from Start menu)

Application Files...

Prerequisites...

Updates...

Options...

**Publish Version**

Major: 1 Minor: 0 Build: 0 Revision: 1

☒ Automatically increment revision with each publish

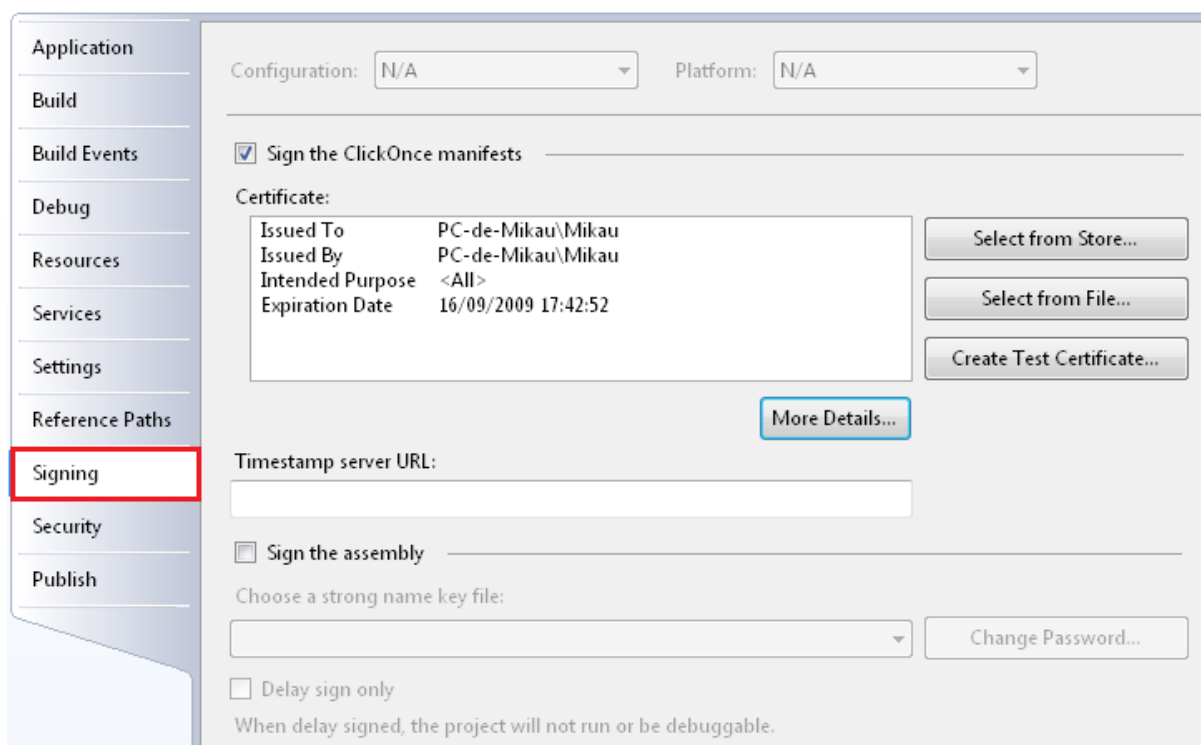
Publish Wizard... Publish Now

Nous utiliserons au cours de ce chapitre une application simple contenant seulement un formulaire « Form1 » vide et sans fonctionnalités.

## 2.2 Les éléments du ClickOnce

### 2.2.1 Signature

Cette partie permet de signer vos applications. C'est-à-dire indiquer l'origine de l'application grâce à une signature. Cela permet d'avoir un nom fort de disponible et une empreinte (signature) qui prouvera que vous êtes bien l'auteur de l'application. Elle est disponible dans la partie *Signature* :



Application

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

**Signing**

Security

Publish

Configuration: N/A Platform: N/A

☒ Sign the ClickOnce manifests

Certificate:

Issued To	PC-de-Mikau\Mikau
Issued By	PC-de-Mikau\Mikau
Intended Purpose	<All>
Expiration Date	16/09/2009 17:42:52

Select from Store...

Select from File...

Create Test Certificate...

More Details...

Timestamp server URL:

☐ Sign the assembly

Choose a strong name key file:

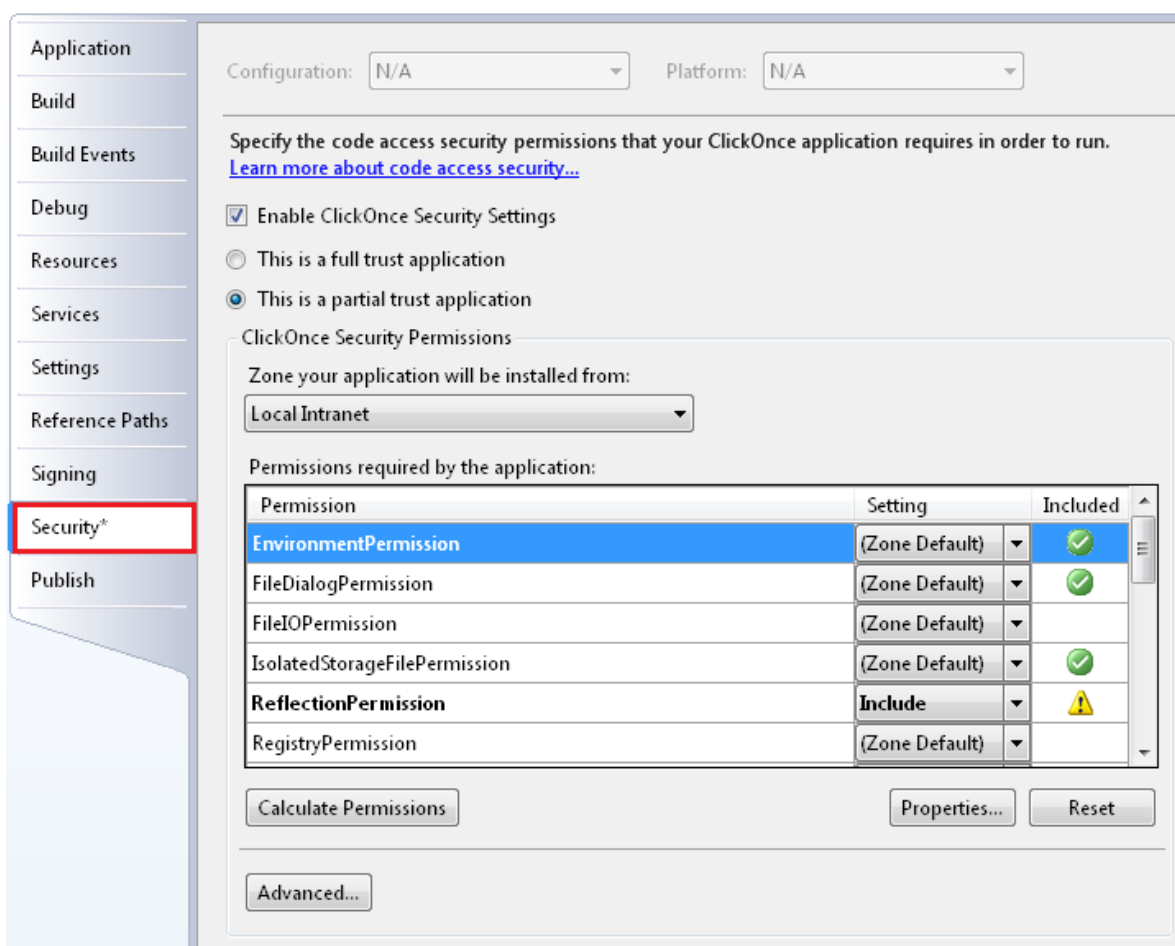
Change Password...

☐ Delay sign only

When delay signed, the project will not run or be debuggable.

## 2.2.2 Sécurité

Cette partie permet de configurer les sécurités de votre application, c'est-à-dire les autorisations minimales qu'il faut pour faire fonctionner correctement votre application. Ces autorisations sont personnalisables suivant les protections que vous souhaitez mettre. Sécuriser votre application est très important, ceci empêchera à un utilisateur malveillant d'effectuer des actions non autorisées. L'interface graphique est disponible depuis l'onglet *Sécurité* :



Configuration: N/A Platform: N/A

Specify the code access security permissions that your ClickOnce application requires in order to run.  
[Learn more about code access security...](#)

☒ Enable ClickOnce Security Settings

☐ This is a full trust application

☒ This is a partial trust application

ClickOnce Security Permissions

Zone your application will be installed from:  
Local Intranet

Permissions required by the application:

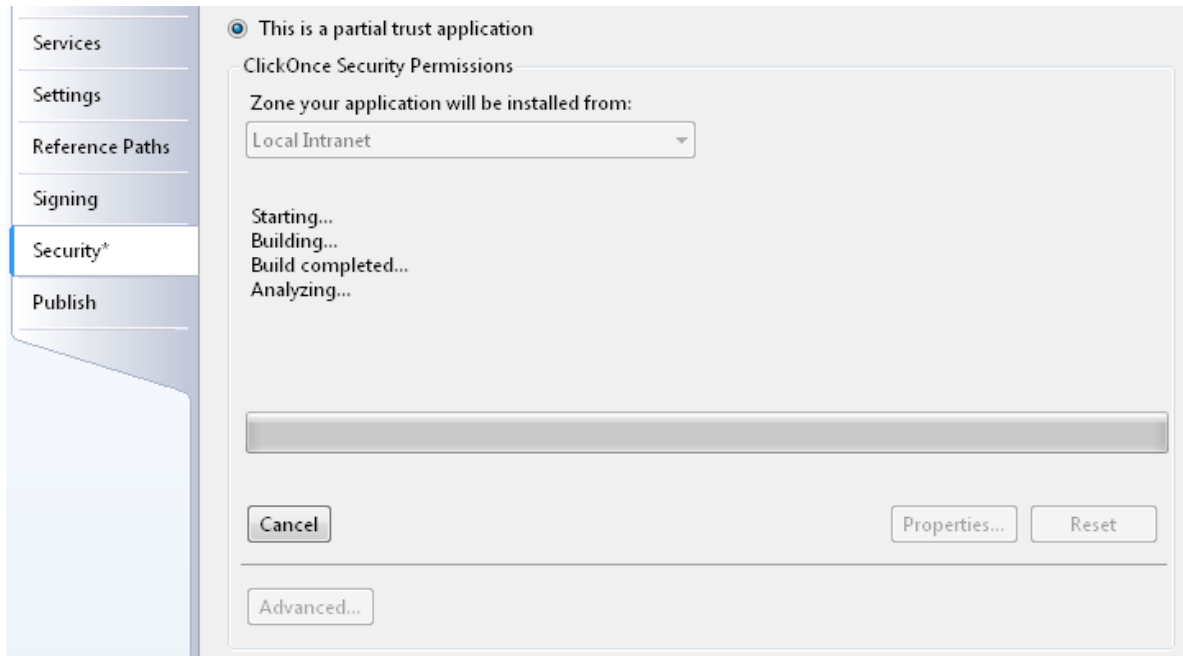
Permission	Setting	Included
EnvironmentPermission	(Zone Default)	✓
FileDialogPermission	(Zone Default)	✓
FileIOPermission	(Zone Default)	
IsolatedStorageFilePermission	(Zone Default)	✓
ReflectionPermission	Include	⚠
RegistryPermission	(Zone Default)	

Calculate Permissions Properties... Reset

Advanced...

Il existe deux options principales : une option permettant une protection personnalisée (« C'est une application à confiance partielle ») et une option permettant d'activer toutes les permissions (« C'est une application à confiance totale »).

Nous vous recommandons de choisir l'option avec autorisations personnalisées. De plus Visual Studio vous permet de définir les autorisations adaptées à votre application. Pour cela, il suffit de cliquer sur « Calculer les permissions », Visual Studio va analyser chaque ligne de votre code et donner les autorisations suffisantes pour le bon fonctionnement de l'application (les droits choisis par Visual Studio apparaissent en gras) :

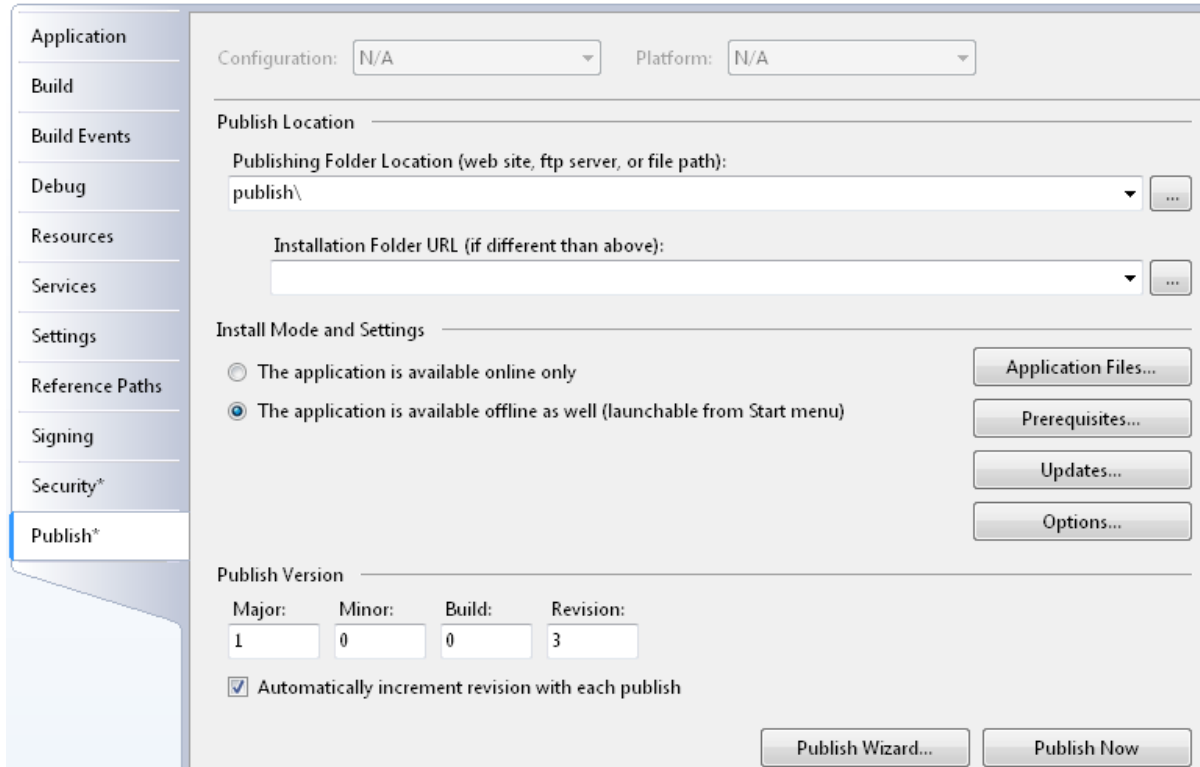


Ensuite vous pouvez soit laisser les droits que Visual Studio vous propose soit en rajouter. Si vous souhaitez tout de même rajouter des droits, il suffit de cocher « Inclure » sur les autorisations que vous souhaitez activer :

<b>FileIOPermission</b>	(Zone Default)	
IsolatedStorageFilePermission	(Zone Default)	✓
<b>ReflectionPermission</b>	<b>Include</b>	⚠
	Exclude	

### 2.2.3 Publication

La partie « Publication » permet, comme son nom l'indique, de configurer les options de publication de votre application mais aussi de configurer les options de mise à jour.

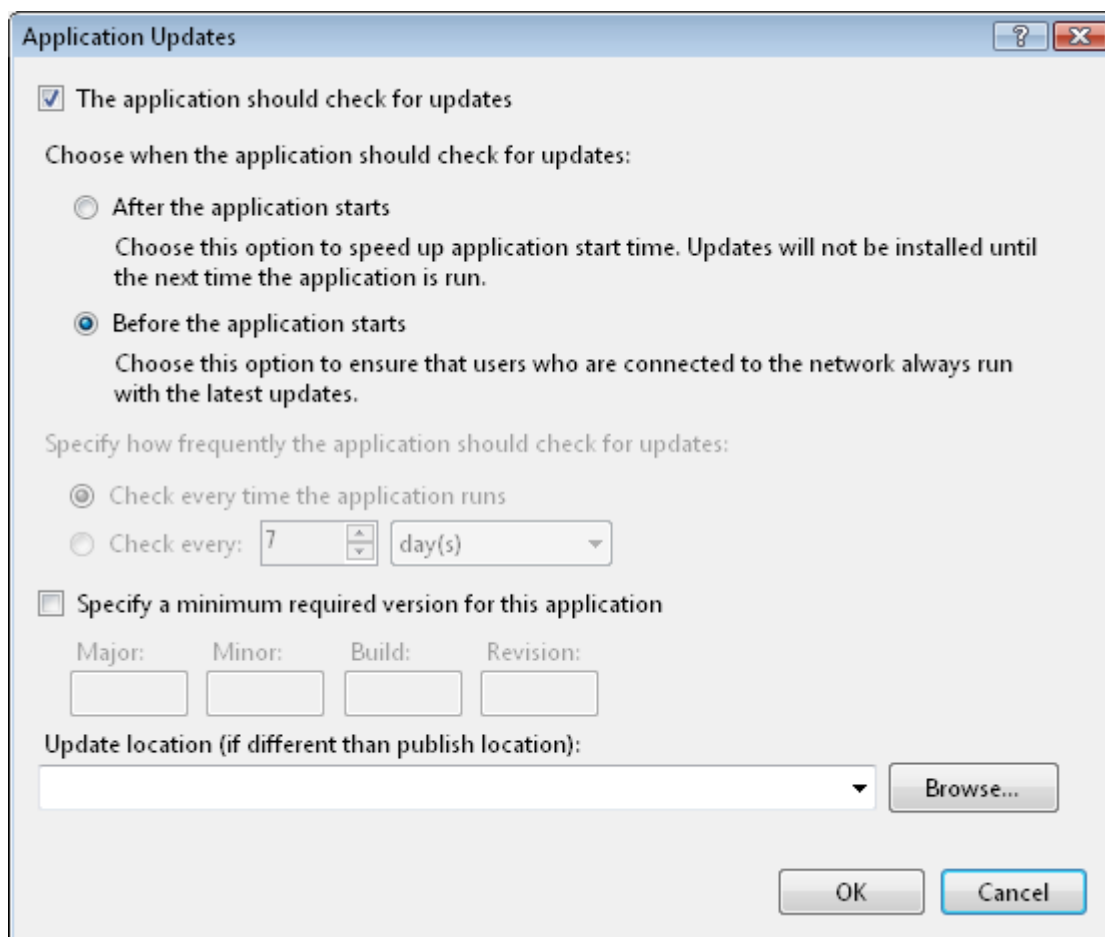


The screenshot shows the 'Publish' tab of the Visual Studio Publish Wizard. On the left is a sidebar with navigation links: Application, Build, Build Events, Debug, Resources, Services, Settings, Reference Paths, Signing, Security\*, and Publish\* (which is highlighted). The main area is divided into sections:

- Configuration:** Two dropdown menus, both set to 'N/A'.
- Publish Location:**
  - Publishing Folder Location (web site, ftp server, or file path):** A dropdown menu showing 'publish\' with a browse button (...).
  - Installation Folder URL (if different than above):** An empty text box with a browse button (...).
- Install Mode and Settings:**
  - Two radio buttons: 'The application is available online only' (unselected) and 'The application is available offline as well (launchable from Start menu)' (selected).
  - Four buttons on the right: 'Application Files...', 'Prerequisites...', 'Updates...', and 'Options...'.
- Publish Version:**
  - Four input fields: Major (1), Minor (0), Build (0), and Revision (3).
  - A checked checkbox: 'Automatically increment revision with each publish'.
  - Two buttons at the bottom right: 'Publish Wizard...' and 'Publish Now'.

Vous pouvez définir le lieu de publication grâce aux combos box « Lieu de publication ». Cela peut être un chemin d'accès, une adresse FTP ou HTTP, ou encore une adresse de partage réseau. C'est en général le lieu où l'application sera installée.

Ensuite, vous pouvez configurer l'accessibilité de votre application. C'est-à-dire de savoir si l'application ne sera disponible qu'en ligne (elle sera téléchargée et réinstallée à chaque fois) ou si elle sera disponible hors ligne (elle est installée et accessible via le menu « Démarrer »). Pour cela, il suffit de cocher l'option désirée via la partie « Mode et paramètre d'installation ». Si vous choisissez l'option hors ligne, vous devrez spécifier les options de mises à jour en cliquant sur le bouton « Mises à jour ». Vous aurez alors cette boîte de dialogue qui s'ouvrira :



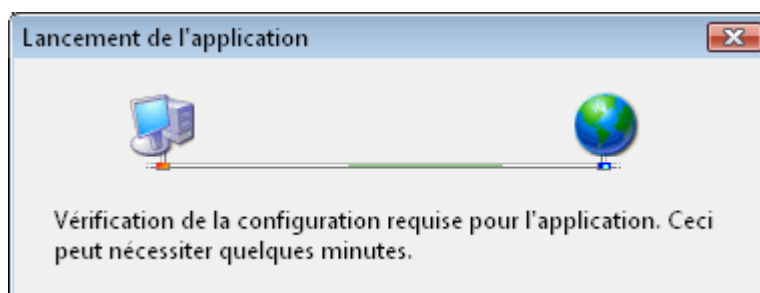
Ici vous pourrez définir si les mises à jour se font avant ou après l'exécution de l'application ainsi que la régularité des vérifications de versions.

Afin de faire une mise à jour, il suffit de refaire une publication avec la nouvelle version de l'application. Lorsqu'un utilisateur lancera l'application, la nouvelle version sera automatiquement détectée puis installée. Si un utilisateur utilise le fonctionnement en ligne, il aura directement accès à la nouvelle version.

Une fois tout de configuré, il suffit de cliquer sur le bouton « Publier maintenant » pour publier votre application.

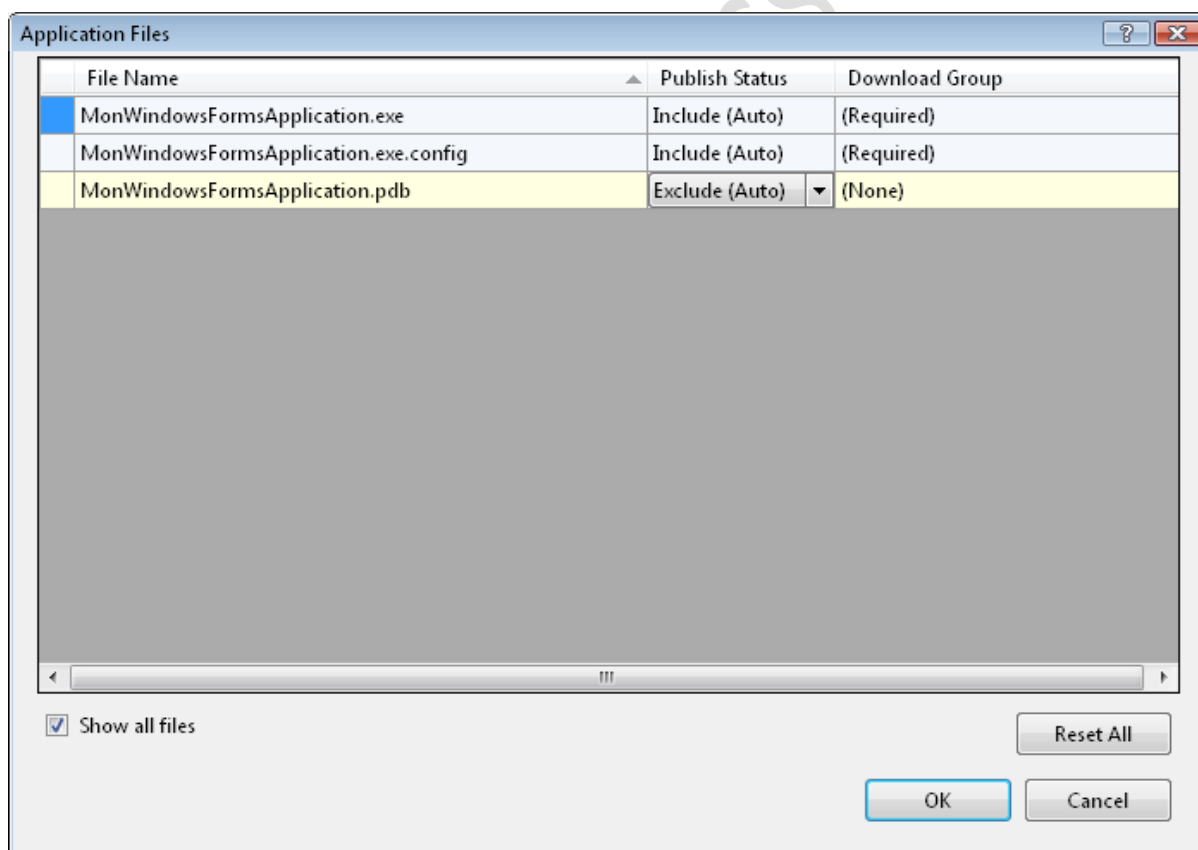


Une fois votre application publiée, il suffit de l'installer en double cliquant sur le fichier « setup » :

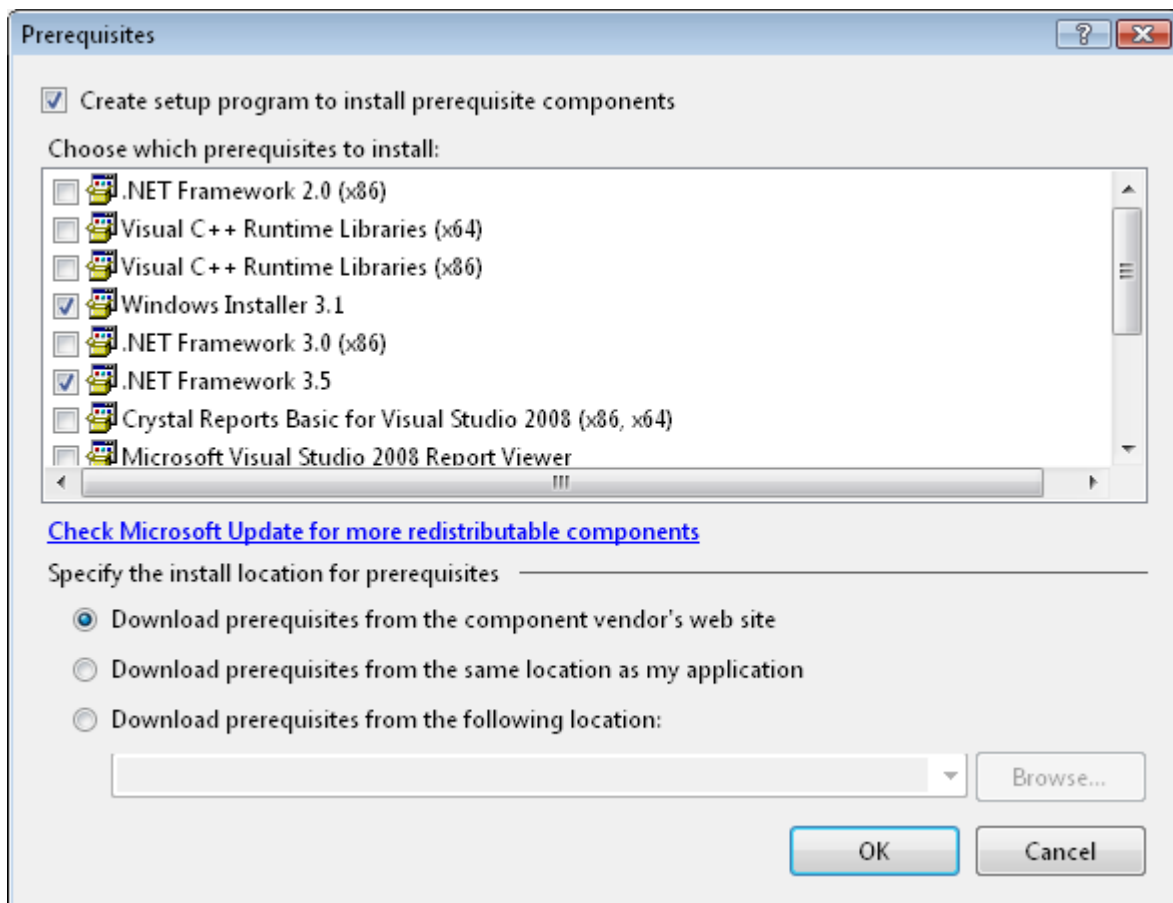


## 2.2.4 Configurer l'installation

Vous pouvez aussi définir quels fichiers de votre projet seront publiés. Cela peut être utile afin de faire tester une version beta par exemple ou encore de retirer certaines fonctionnalités. Pour cela, il suffit de cliquer sur le bouton « Fichier de l'application... » situé dans la partie « Publication » :



Vous pouvez aussi définir les composants pré-requis à l'installation. C'est-à-dire que votre application ne sera installée qu'à condition que l'utilisateur possède les composants définis. Ceci permet d'installer l'application et être sûr qu'il fonctionnera correctement. Ce menu est accessible en cliquant sur le bouton « Pré-requis... » situé dans la partie « Publication » :



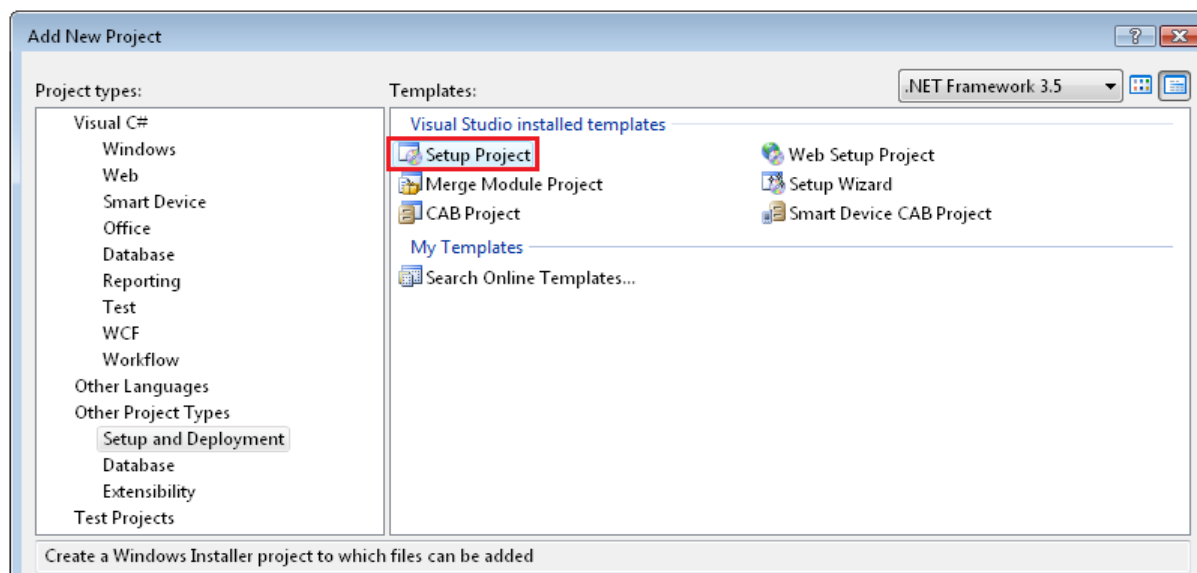
### 3 Gestion du Projet d'installation

#### 3.1 Qu'est ce que le Projet d'installation

Le Projet d'installation, comme le ClickOnce, est un outil permettant le déploiement. En revanche, il est plus configurable et par conséquent plus complexe d'utilisation. Cet outil est recommandé pour de grosses applications ou de grands déploiements hautement configurés. Il permet entre autre de créer de nouveaux dossiers ou d'en utiliser des anciens sur un ordinateur ciblé puis de copier et exécuter des actions sur ces derniers. Lorsque que vous compilerez un Projet d'installation, cela créera automatiquement un fichier .msi qui lors de son exécution démarrera un assistant d'installation qui permettra l'installation de votre application. Ce fichier msi peut être transféré facilement (CD-ROM, fichier de partage, ...).

### 3.2 Créer un Projet d'installation

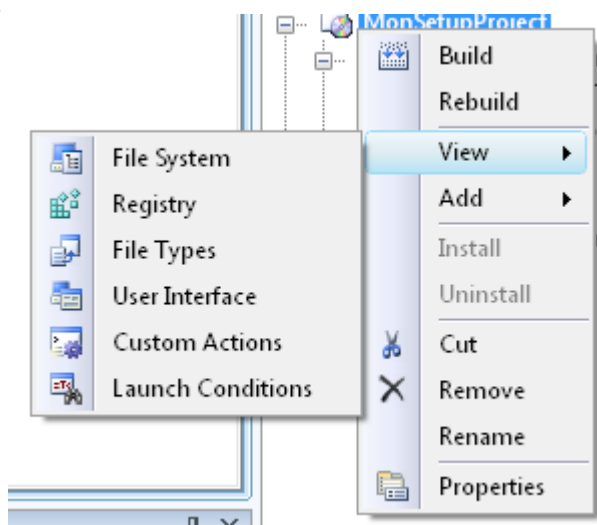
Créer un Projet d'installation est simple. Il suffit d'aller sur Fichier du menu de Visual Studio puis de choisir Ajouter puis Nouveau projet. Cela vous ouvrira une fenêtre dans laquelle vous choisirez Projet d'installation :



Remarque : Vous avez créé votre Projet d'installation, ceci ouvrira automatiquement sur la page centrale de Visual Studio l'éditeur Fichier Système décrit dans la partie 3.4.

### 3.3 Les éditeurs du Projet d'installation

Le Projet d'installation possède un total de six éditeurs permettant de le configurer. Ils sont accessibles en faisant clique droit sur votre Projet d'installation dans l'Explorateur de Solution puis sur Affichage :



On voit donc les six éditeurs dont chacun à une fonction précise :

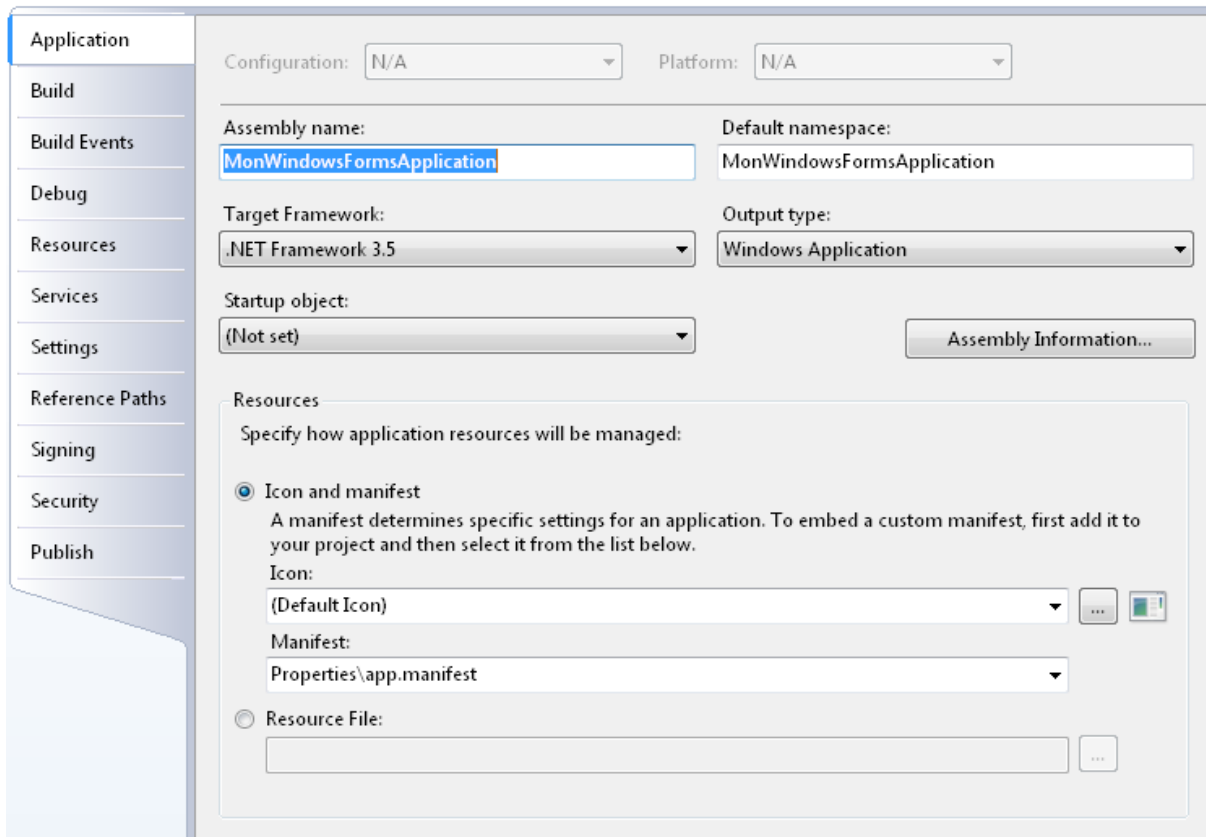
Editeur	Description
Fichier Système	Cet éditeur permet de configurer l'installation de votre application en gérant les fichiers systèmes sur un ordinateur ciblé.
Registre	Cet éditeur permet d'écrire des clés registres durant l'installation de votre application.
Types de Fichier	Cet éditeur permet de créer les associations entre votre application et les types de fichiers.
Interface Utilisateur	Cet éditeur permet de créer une interface utilisateur visible durant l'installation de votre application et permettant à un utilisateur de configurer son installation.
Actions Personnalisées	Cet éditeur permet de créer les actions personnalisées qui s'exécuteront pendant l'installation de votre application.
Conditions de Lancement	Cet éditeur permet de donner des conditions requises afin qu'un utilisateur puisse installer votre application.

## 3.4 Utiliser un Projet d'installation

### 3.4.1 Propriétés du Projet d'installation

Le projet d'installation possède une multitude de propriétés. Celles-ci sont accessibles de la même façon que le ClickOnce c'est-à-dire en faisant clique droit dans le Solution Explorer puis en cliquant sur « Propriétés ».

Vous arrivez donc à cette page :



The screenshot shows the 'Application' properties window in Visual Studio. On the left is a sidebar with tabs: Application, Build, Build Events, Debug, Resources, Services, Settings, Reference Paths, Signing, Security, and Publish. The 'Application' tab is selected. The main area contains the following settings:

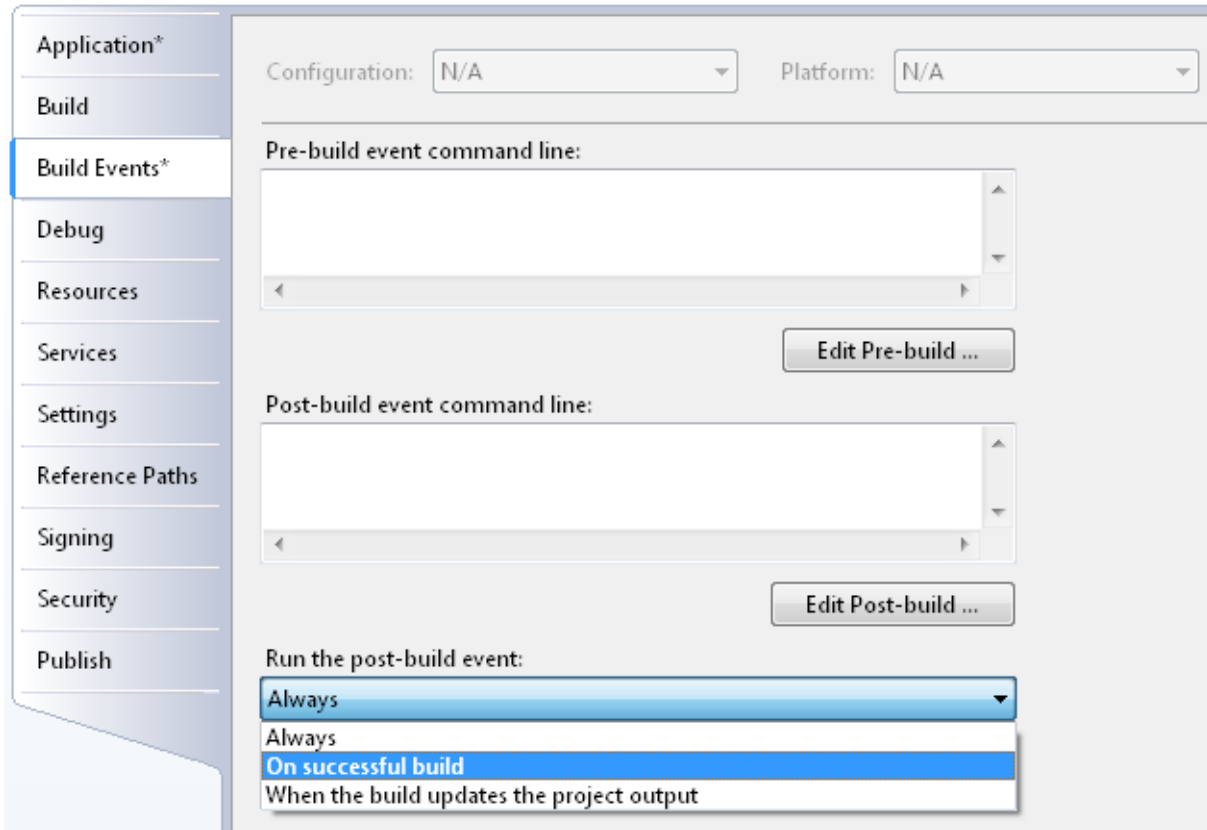
- Configuration: N/A (dropdown)
- Platform: N/A (dropdown)
- Assembly name: MonWindowsFormsApplication (text box)
- Default namespace: MonWindowsFormsApplication (text box)
- Target Framework: .NET Framework 3.5 (dropdown)
- Output type: Windows Application (dropdown)
- Startup object: (Not set) (dropdown)
- Assembly Information... (button)
- Resources section: 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) (dropdown) with browse and icon buttons
    - Manifest: Properties\app.manifest (dropdown)
  - ☐ Resource File: (empty text box) with browse button

Nous avons vu que le ClickOnce était configurable grâce aux parties Sécurité et Publication. La partie signature est commune aux deux outils.

→ **Dans la partie Application** : C'est la partie qui contient les informations à propos de votre application, c'est-à-dire sur quelle version du Framework elle fonctionne, le type de votre application, son nom, ...

Voir l'imprime écran précédent.

→ **Dans la partie Evènements de Construction** : C'est la partie qui vous permettra de donner des lignes de commandes qui seront exécutées soit avant soit après la construction. Dans le cas d'après la construction, vous pouvez aussi définir si la/les lignes de commandes seront exécutée(s) dans le cas d'une erreur.



Application\*

Build

**Build Events\***

Debug

Resources

Services

Settings

Reference Paths

Signing

Security

Publish

Configuration: N/A Platform: N/A

Pre-build event command line:

Edit Pre-build ...

Post-build event command line:

Edit Post-build ...

Run the post-build event:

- Always
- On successful build**
- When the build updates the project output

→ **Dans la partie Signature** : C'est la partie qui permet de signer vos applications. C'est-à-dire indiquer l'origine de l'application grâce à une signature. Cela permet d'avoir un nom fort de disponible et une empreinte (signature) qui prouvera que vous êtes bien l'auteur de l'application.

Application

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

**Signing**

Security

Publish

Configuration: N/A Platform: N/A

☒ Sign the ClickOnce manifests

Certificate:

Issued To	PC-de-Mikau\Mikau
Issued By	PC-de-Mikau\Mikau
Intended Purpose	<All>
Expiration Date	16/09/2009 17:42:52

Select from Store...

Select from File...

Create Test Certificate...

More Details...

Timestamp server URL:

☐ Sign the assembly

Choose a strong name key file:

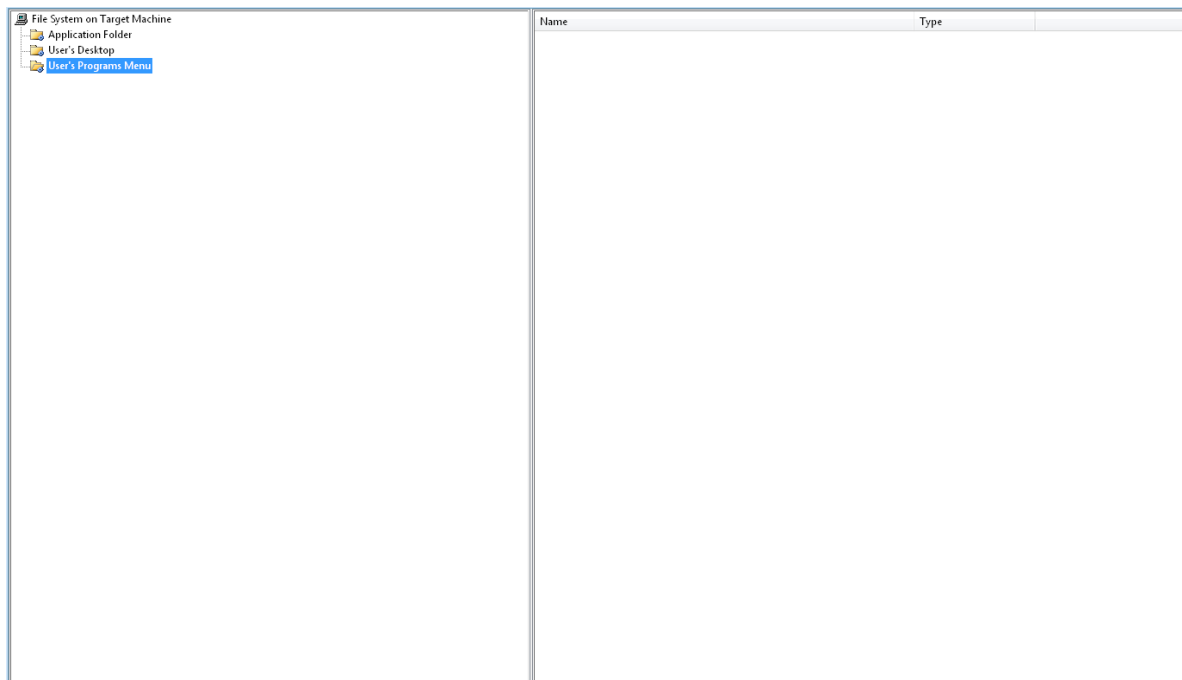
Change Password...

☐ Delay sign only

When delay signed, the project will not run or be debuggable.

### 3.4.2 L'éditeur Fichier Système

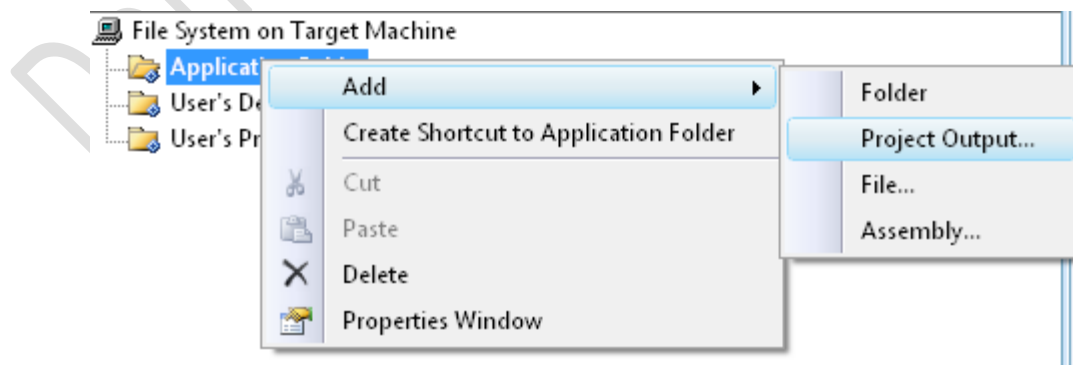
L'éditeur Fichier Système est automatiquement ouvert lors de la création sinon vous pouvez l'ouvrir en faisant clique droit sur votre Projet d'installation dans l'Explorateur de Solution puis sur Affichage. Vous remarquerez que l'éditeur est divisé en deux parties. La partie de gauche permet de voir les dossiers situés dans l'ordinateur ciblé et la partie de droite le détail des dossiers sélectionnés dans la partie gauche.



Remarque : De base, l'éditeur comprend trois fichiers : Bureau de l'utilisateur, Dossier d'application et Menu de Programmes de l'utilisateur.

A partir de là vous pouvez faire plusieurs actions :

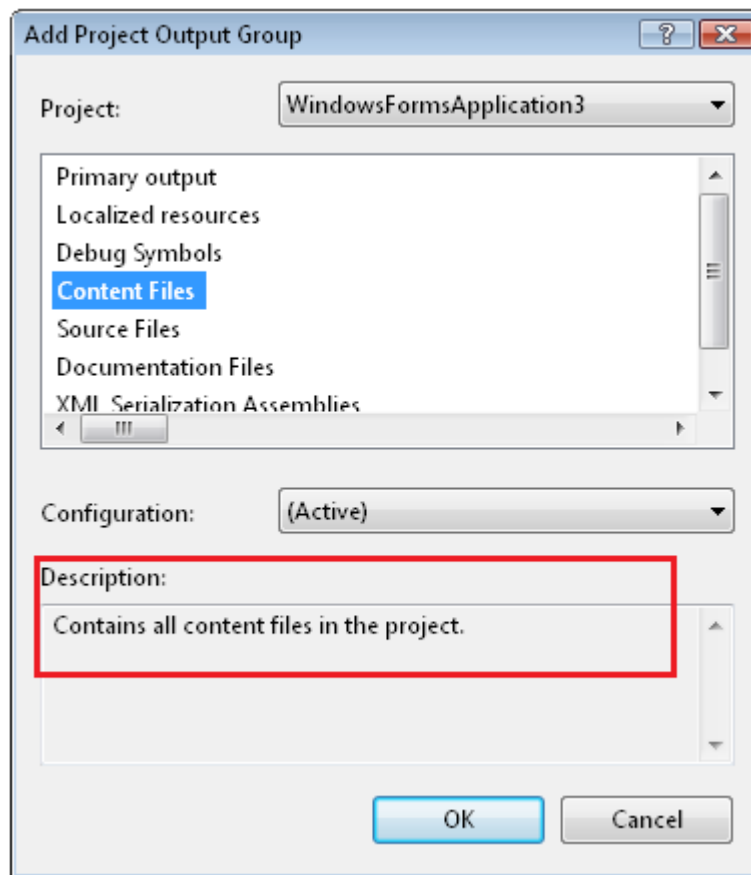
➔ **Ajouter un fichier de votre application dans un dossier** : Pour cela, il suffit de faire clique droit sur un des dossiers puis faire Ajouter et Sortie de Projet.



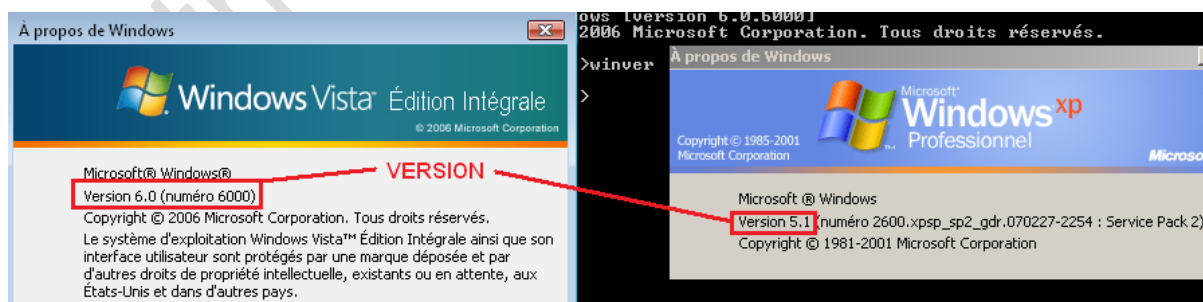
Cela ouvrira la boîte de dialogue « Ajouter le groupe de sorties du projet ».



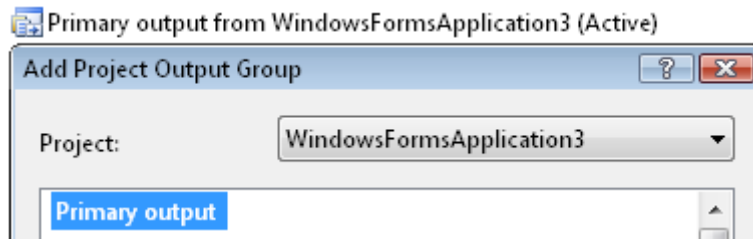
Vous trouverez tout ce qui concerne votre projet (les fichiers DLL, EXE, de débogage, etc....). Chaque fois que vous cliquerez sur un des dossiers vous verrez sa description en bas de la boîte de dialogue. Vous n'avez plus qu'à sélectionner celui que vous souhaitez et faire OK.



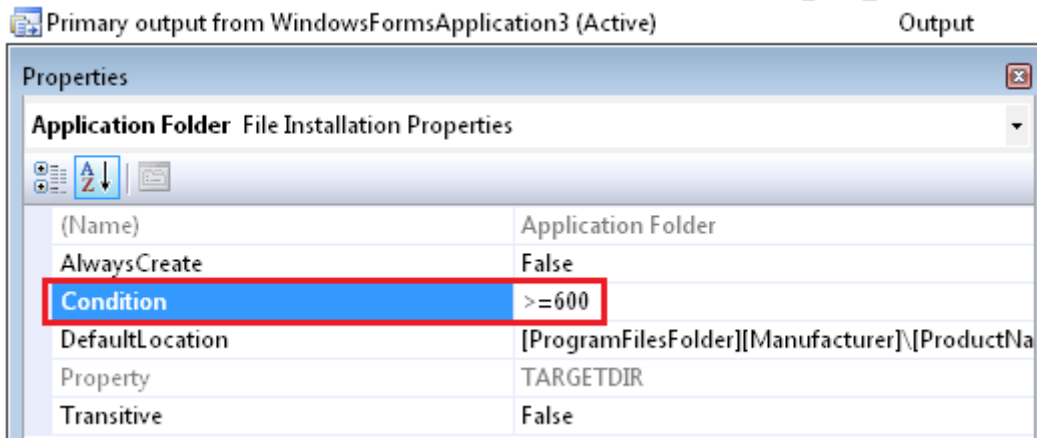
➔ **Empêcher l'installation de votre application sur certains systèmes d'exploitation** : Vous devez tout d'abord connaître les versions de votre Windows ou ceux visés par le déploiement. Pour cela, il suffit d'ouvrir le cmd à partir d'un menu Démarrer puis de taper la commande « winver ». Vous verrez s'ouvrir une fenêtre vous donnant la version de votre Windows :



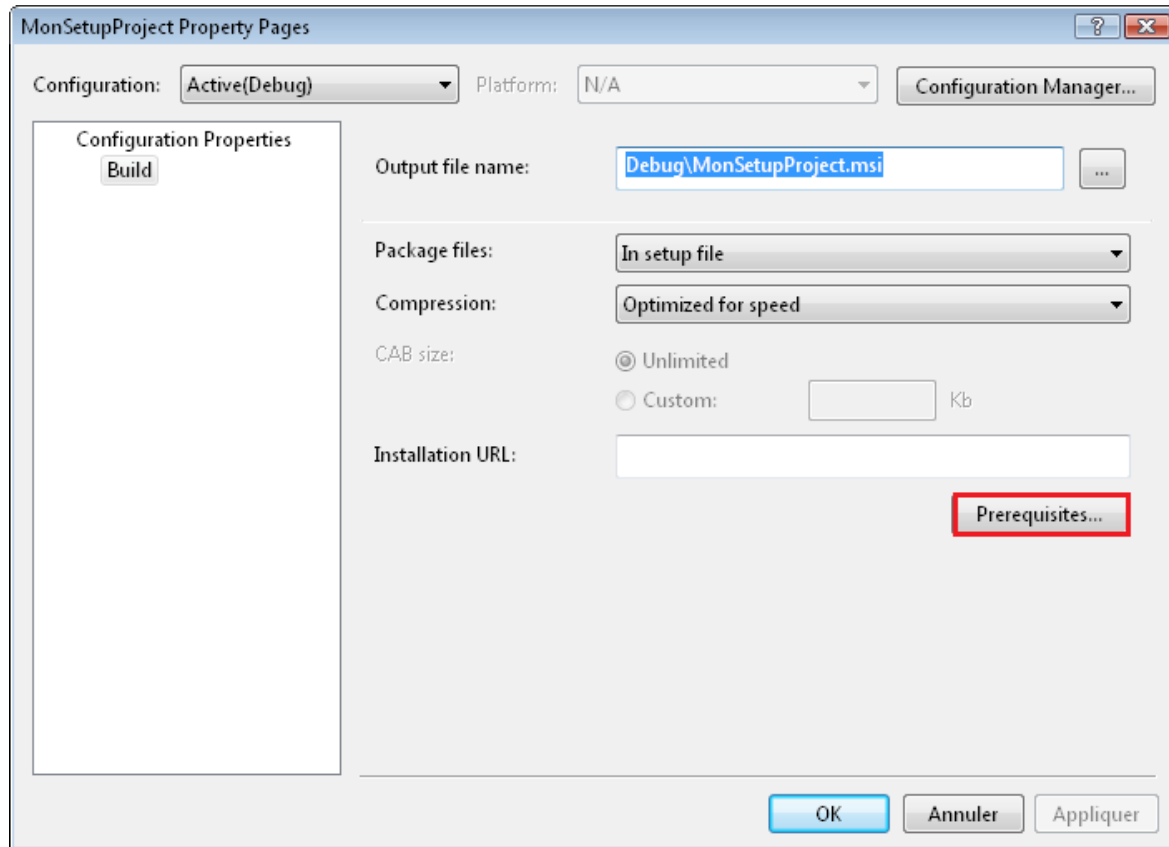
Ensuite vous devez faire clique droit sur la première sortie de votre application. Vous pouvez la créer en faisant clique droit sur un des dossiers puis faire Ajouter et Sortie de Projet. La boîte de dialogue « Ajouter le groupe de sorties du projet » ouverte, sélectionner « Première Sortie » :



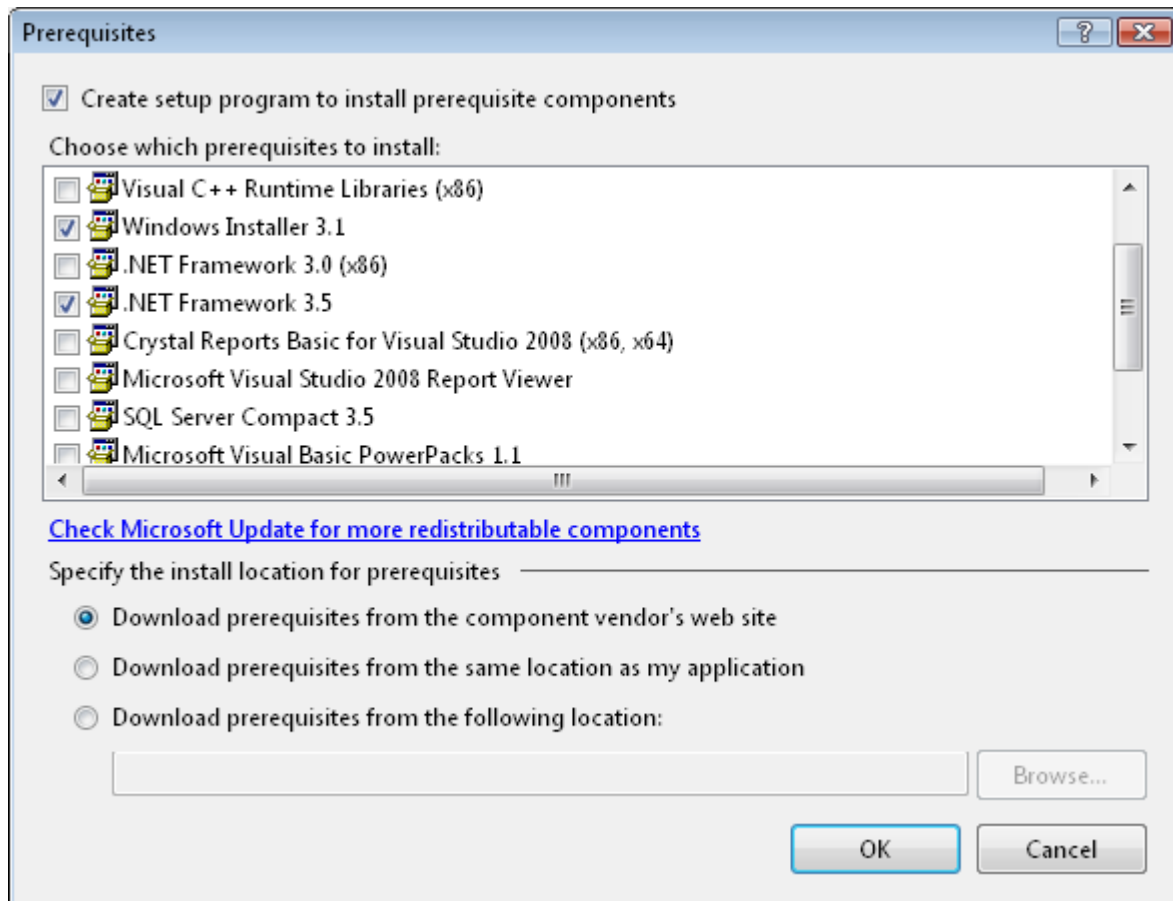
Une fois cela créé, sélectionnez le et faites « Fenêtre Propriété ». Dans celle-ci vous remarquerez la propriété *Condition* qui vous permettra de définir à partir de quel système d'exploitation votre application peut être installée. Il suffit de faire un calcul simple : Version Majeure x 100 + Version mineur. Par exemple avec le Windows Vista de l'imprime écran précédent on a, à la fin du calcul, une valeur de 600. Si vous souhaitez que votre application ne soit disponible qu'à partir de cette version de Windows, il suffira de mettre  $\geq 600$  sur votre propriété *Condition* :



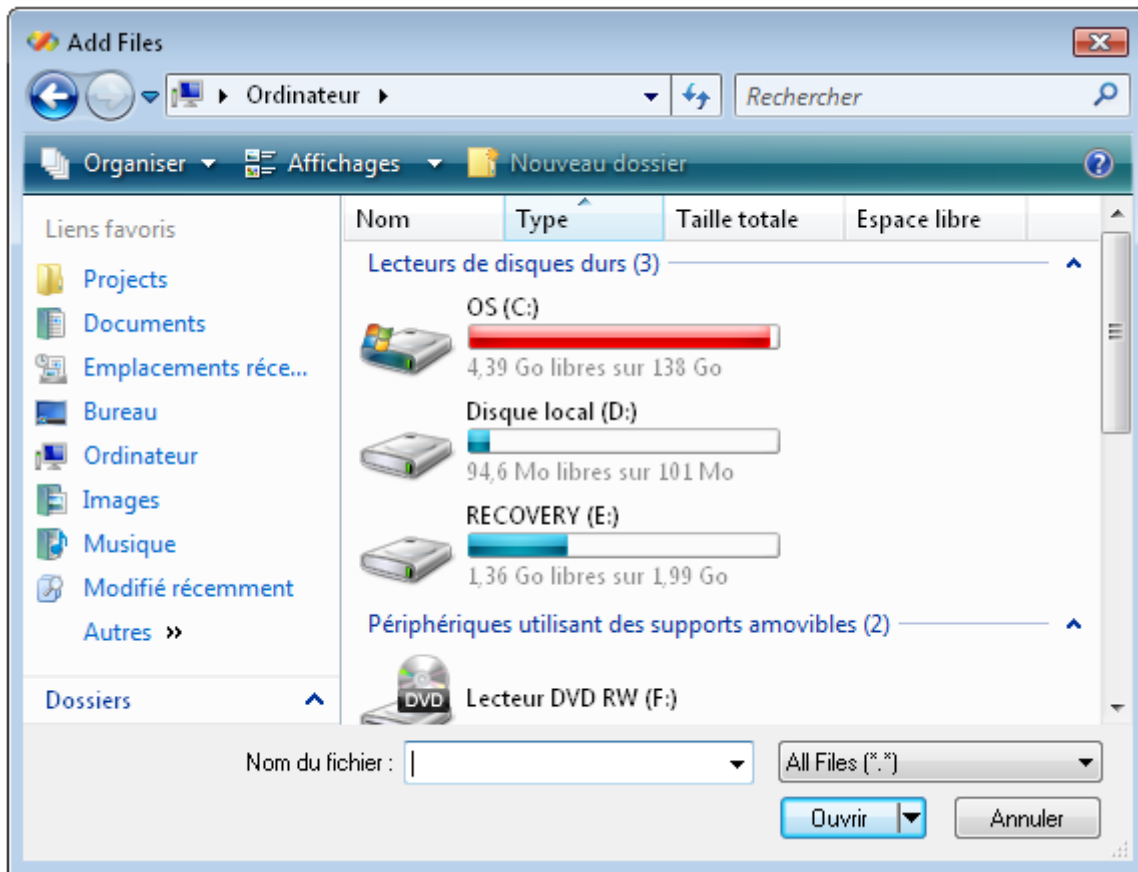
➔ **Empêcher l'installation de votre application suivant les composants d'un ordinateur** : Le système est le même que pour le ClickOnce. Dans votre Exploreur de solution, faites un clic droit sur votre Projet d'installation, cela ouvrira la boîte de dialogue « Page de Propriétés » « NomDeVotreProjetd'Installation » :



Cliquez ensuite sur le bouton « Pré-requis » qui ouvrira la boîte de dialogue dans laquelle vous pourrez sélectionner les composants que vous jugerez nécessaires au bon fonctionnement de votre application :




➔ **Associer une icône à un dossier de votre éditeur** : Vous devez tout d'abord sélectionner un fichier de votre éditeur dont vous voulez associer l'icône. Ensuite faites clique droit dessus puis Ajouter et Fichier. Cela vous ouvrira la boîte de dialogue « Ajouter un Fichier » :



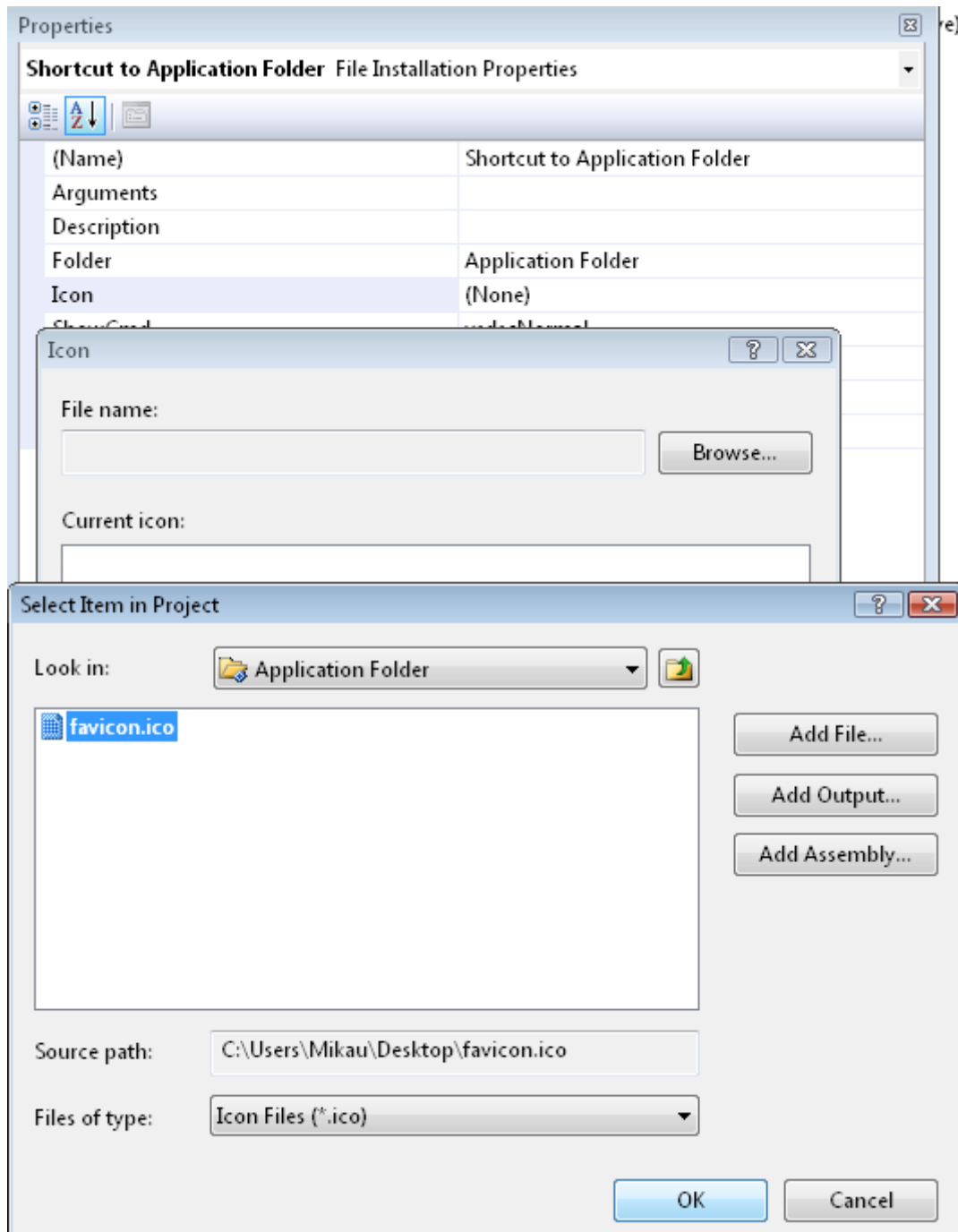
Une fois la boîte de dialogue ouverte, cherchez votre icône et faites Ouvrir.

Vous devriez voir votre icône sur la partie droite de votre éditeur comme ceci (« favicon.ico » est une icône créée juste pour l'exemple) :

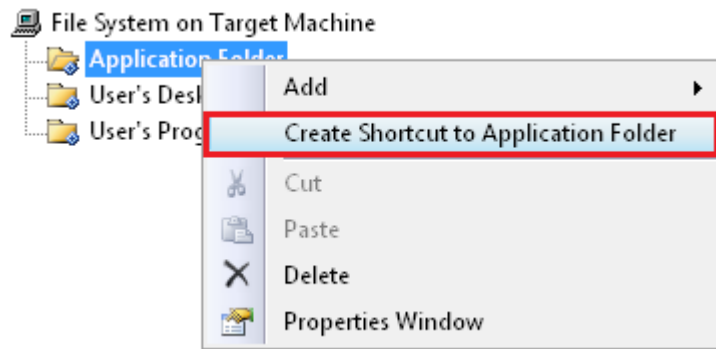
Name	Type
 favicon.ico	File

Maintenant, il faut associer l'image. Pour cela vous devez créer un raccourci de votre fichier en faisant clique droit sur votre fichier puis « Créer un raccourci vers « Nom de votre dossier » ».

Une fois le raccourci créé, sélectionnez votre raccourci sur la partie droite de votre éditeur puis faites « Fenêtre Propriété ». La boîte de dialogue des propriétés s'ouvre, vous n'avez plus qu'à associer l'icône dans la propriété *Icon* :



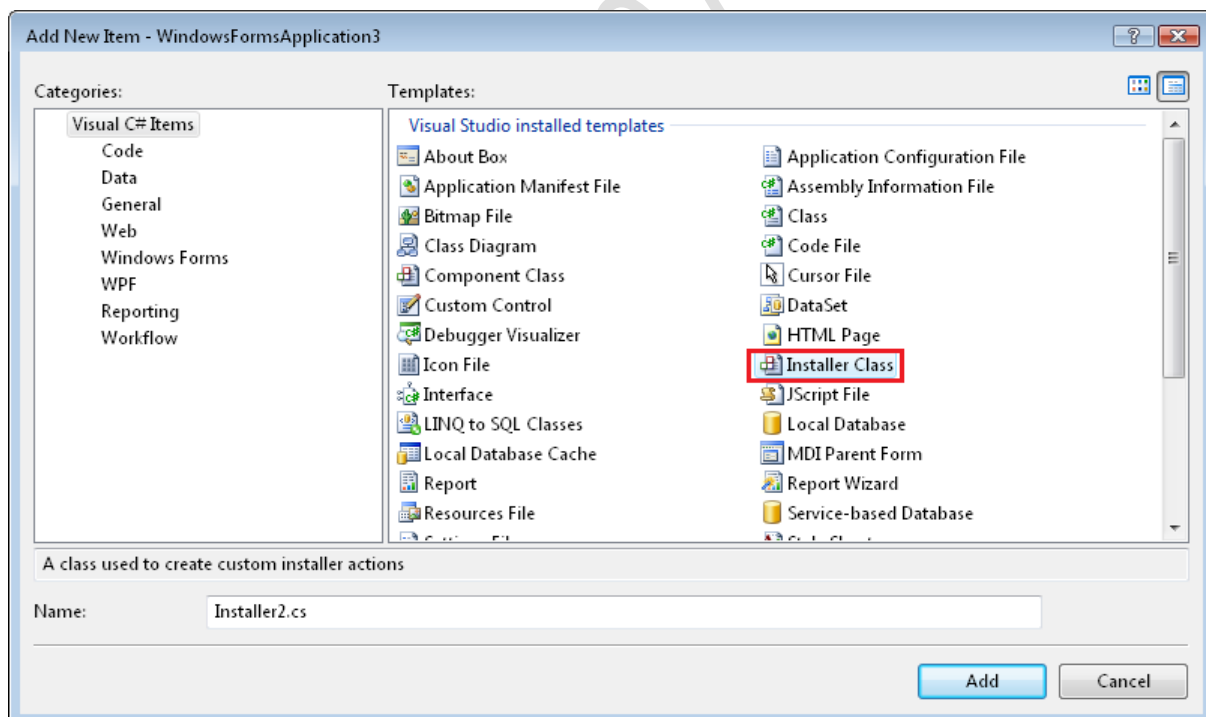
➔ **Créer un raccourci d'un fichier de l'éditeur** : Pour cela, il suffit de faire clique droit sur votre fichier puis de faire « Créer un raccourci vers « Nom de votre dossier » ».



### 3.4.3 L'éditeur Actions Personnalisées

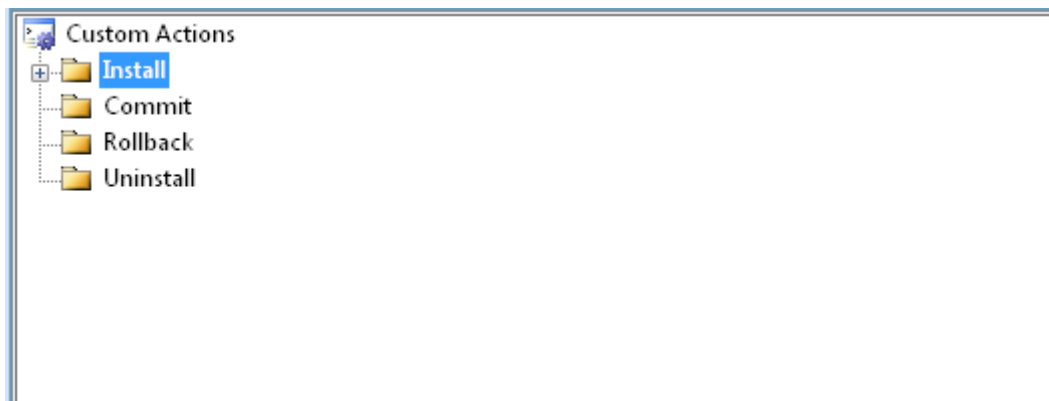
L'éditeur Actions Personnalisées permet de créer des installations de niveau plus avancé avec plus de configuration : il va vous permettre d'exécuter du code pendant l'installation. En revanche pour mettre le code permettant d'utiliser cet outil, il vous faudra créer une classe *Installer*.

→ **Créer une classe *Installer*** : Pour créer une classe *Installer*, il suffit de faire clique droit sur votre projet puis faire Ajouter et Ajouter un Nouvel Objet. Cela ouvrira la boîte de dialogue « Ajouter un Nouvel Objet », ensuite sélectionner la classe *Installer* :



Cette classe possède les méthodes utiles pour l'utilisation de l'outil : *Install*, *Commit*, *Rollback* et *Uninstall*.

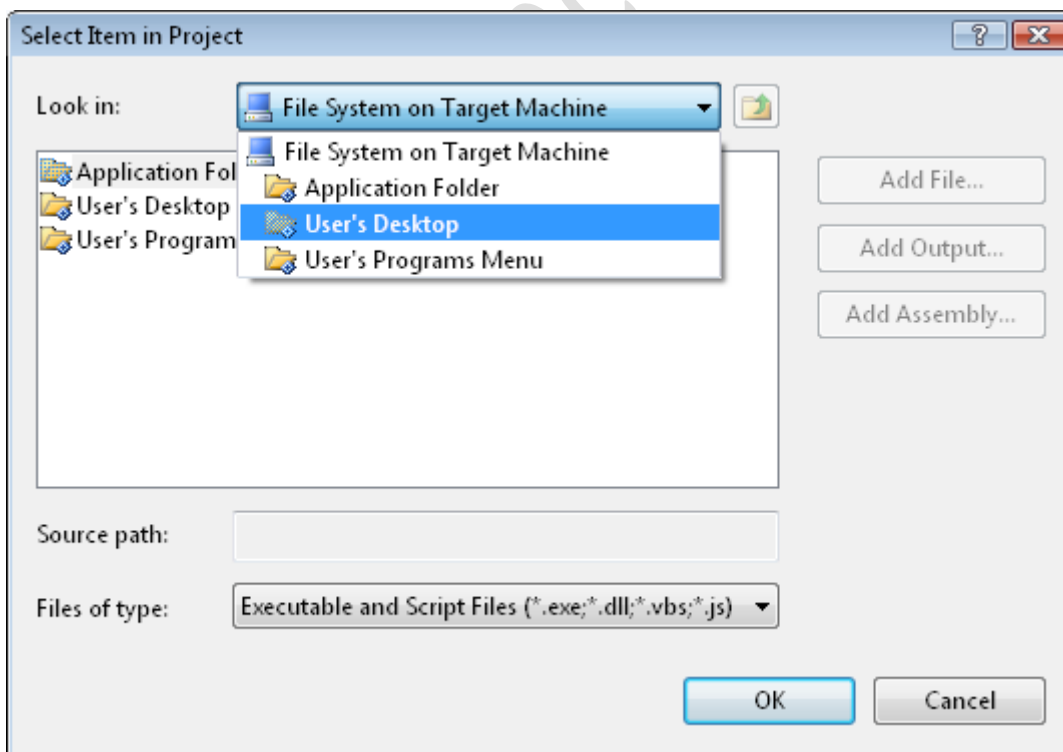
➔ **Afficher l'éditeur Actions Personnalisées** : Pour afficher l'éditeur, faites cliquer droit sur votre Projet d'installation et faites Affichage. Sélectionnez Actions Personnalisées, vous devriez arriver sur cette page :



Remarque : Vous pouvez donc voir les fichiers nommés précédemment : *Installer*, *Valider*, *Restaurer* et *Désinstaller*.

➔ **Créer une action personnalisée** : Une fois votre application créée, la classe *Installer* ajoutée et l'éditeur ouvert, écrivez l'action que vous souhaitez dans la classe *Installer* en utilisant les méthodes *Install*, *Commit*, *Rollback* ou *Uninstall*.

Ensuite faites cliquer droit sur l'évènement (un des quatre visible du dernier imprime écran) et faites « Ajouter une action personnalisée ». Cela vous ouvrira une boîte de dialogue dans laquelle vous pourrez ajouter un élément réalisant l'action de votre choix :



Vous n'avez plus qu'à configurer les propriétés de l'élément que vous avez ajouté.



## 4 Conclusion

Vous avez fini ce chapitre sur les méthodes de déploiement en *Windows Form*. Nous vous conseillons de bien adapter votre méthode de déploiement (ClickOnce et Projet d'installation) par rapport au besoin de déploiement, ainsi que d'être vigilant des configurations que vous définissez.

L'équipe *Windows Form*.

Dotnet-France Association