



Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

Déployer une application cliente avec le Framework .NET 3.5 Client Profile

Version 1.0



James RAVAILLE

<http://blogs.dotnet-france.com/jamesr>



Sommaire

1	Introduction.....	3
1.1	Présentation	3
1.2	Pré-requis	3
1.3	Présentation de l'application à déployer	3
2	Présentation du Framework .NET Client	5
2.1	Les avantages apportés	5
2.2	Architecture.....	5
2.3	Les packages proposés par Microsoft	5
3	Activation du sous-ensemble du Framework .NET client.....	7
3.1	Dans l'application C#	7
3.2	Dans l'application VB .NET	7
3.3	Modifications apportées au projet.....	8
3.4	Attention aux assemblies utilisés dans votre projet	9
4	Déployer une application avec un sous-ensemble du Framework .NET Client et Windows Installer.....	10
4.1	Création du projet Windows Installer	10
4.2	Installation.....	15
5	Déployer une application avec un sous-ensemble du Framework .NET client et un ClickOnce ...	16
5.1	Scénario de déploiement	16
5.2	Configuration du projet de déploiement	16
5.3	Installation.....	18
6	Conclusion	23

1 Introduction

1.1 Présentation

A partir du Service Pack 1 du Framework .NET 3.5, Microsoft propose le Framework .NET 3.5 Client Profile. Il s'agit d'un sous ensemble du Framework .NET, qui présente une version simplifiée des composants, permettant d'exécuter (il inclut donc la CLR) des applications clientes conçues avec WPF (Windows Presentation Foundation) ou Windows Forms, qui peuvent utiliser WCF (Windows Communication Foundation) et pouvant être déployées via ClickOnce (déploiement au travers d'un serveur Web).

Ce cours présente l'utilisation du Framework .NET 3.5 Client Profile, au travers de deux scénarii de déploiement : via *Windows Installer* et *ClickOnce*. Pour des raisons de simplification, nous l'appellerons *Framework .NET Client* dans la suite de ce cours.

1.2 Pré-requis

Pour réaliser l'application que nous allons déployer, nous vous recommandons d'avoir des connaissances basiques dans WPF, vous permettant de créer un projet et un simple formulaire.

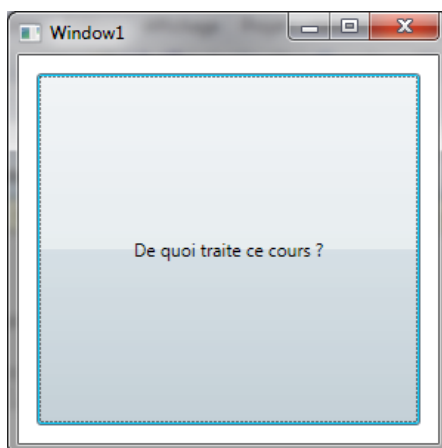
Pour le déploiement d'application, nous vous recommandons de savoir déployer une application :

- Via la création d'un projet de configuration et de déploiement (*Windows Installer*).
- Via *ClickOnce*.

1.3 Présentation de l'application à déployer

Dans ce cours, nous allons déployer une simple application WPF. Cette application ne contient qu'un formulaire, permettant d'afficher le message suivant « Framework .NET 3.5 Client Profile par James RAVAILLE », au travers d'un simple bouton. Elle se nomme *AppliWPF_CS* pour l'application créée avec le langage C#, et *AppliWPF_VB* pour l'application créée avec le langage VB.

Cette application ne contient qu'un seul formulaire, dont le design est le suivant :



Le code XAML de ce formulaire est le suivant :



```
// C# et VB .NET

<Window x:Class="AppliWPF_CS.Window1"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Window1" Height="300" Width="300">
    <Grid>
        <Button Margin="12" Name="CmdAfficherMessage"
Click="CmdAfficherMessage_Click">De quoi traite ce cours ?</Button>
    </Grid>
</Window>
```

Et voici la classe code-behind du formulaire :

```
// C#

public partial class Window1 : Window
{
    public Window1 ()
    {
        InitializeComponent ();
    }

    private void CmdAfficherMessage_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Framework .NET 3.5 Client Profile par James
RAVAILLE");
    }
}
```

```
// VB .NET

Class Window1
    Private Sub CmdAfficherMessage_Click(ByVal sender As System.Object,
ByVal e As System.Windows.RoutedEventArgs)
        MessageBox.Show("Framework .NET 3.5 Client Profile par James
RAVAILLE")
    End Sub
End Class
```

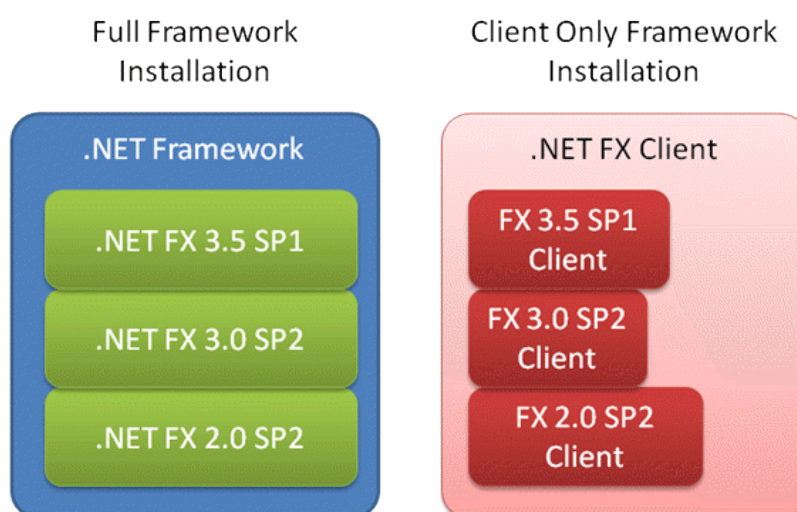
2 Présentation du Framework .NET Client

2.1 Les avantages apportés

Le principal avantage de l'utilisation du Framework .NET Client dans le déploiement d'applications .NET, est de diminuer la durée d'installation des applications, sur les postes de travail sur lesquels aucune version du Framework .NET n'a été utilisée, ou celle n'ayant pas la version 3.5 du Framework .NET. Ainsi, le déploiement d'applications clients via Internet est facilité.

2.2 Architecture

Voici l'architecture du Framework .NET 3.5 Client, comparée à la version complète du Framework .NET 3.5 :



On peut remarquer que le Framework .NET 3.5 Client agrège l'ensemble des composants permettant le développement d'applications clientes depuis la version 2.0 du Framework .NET. Il permet d'exécuter les fonctionnalités de :

- La CLR (environnement d'exécution).
- Déploiement au travers de ClickOnce (déploiement d'applications Web au travers d'un serveur Web).
- Développement d'applications Windows Forms.
- Développement d'applications WPF (Windows Presentation Foundation).
- Développement d'applications WCF (Windows Communication Foundation).

Si aucune version du Framework .NET n'est installée sur le poste de travail des utilisateurs, alors tous les composants du Framework .NET Client sont installés. Mais dans le cas où une version du Framework .NET est déjà installée, par exemple la version 3.0 SP2, alors seuls les composants clients du Framework .NET 3.5 SP1 sont installés.

2.3 Les packages proposés par Microsoft

Au travers de son centre de téléchargement, Microsoft propose :



- Microsoft .NET Framework Client Profile Offline Installer. Il s'agit de la version complète du Framework .NET 3.5 Service Pack 1, auquel s'ajoute le Framework.NET Client. Sa taille est de 257 Mo. Il est disponible en téléchargement ici : <http://www.microsoft.com/downloads/details.aspx?FamilyID=992CFFCB-F8CE-41D9-8BD6-31F3E216285C&displaylang=en>
- Microsoft .NET Framework Client Profile Online Installer. Il s'agit d'un package de 280 Ko, qui détectera, téléchargera et installera uniquement les composants requis pour le bon fonctionnement de l'application (contenus dans le package présenté ci-dessus). Il est disponible ici : <http://www.microsoft.com/downloads/details.aspx?FamilyID=8CEA6CD1-15BC-4664-B27D-8CEBA808B28B&displaylang=fr>

Pour comparer les tailles des packages énoncées ci-dessus, voici quelques autres chiffres :

- Le Framework .NET 3.5 Full Package est de 197 Mo.
- Le Framework .NET 3.5 Full Package Service Pack1 est de 232 Mo.

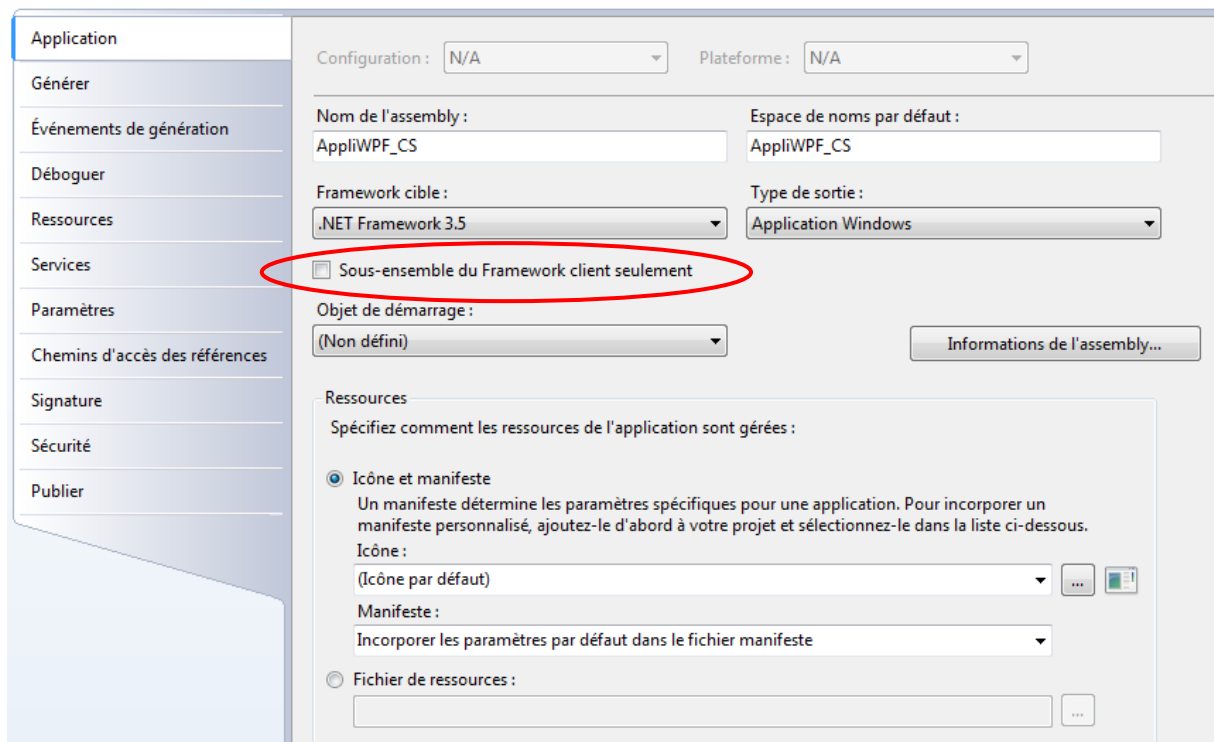


3 Activation du sous-ensemble du Framework .NET client

Lorsque vous créez une application .NET avec le Framework .NET 3.5, cette option n'est pas activée. L'activation est différente est dans les projets C# et VB .NET. Dans ce chapitre, nous exposerons les deux, sur le projet présenté dans le premier chapitre.

3.1 Dans l'application C#

Dans notre application *AppliWPF_CS*, positionnons-nous dans la fenêtre de propriétés du projet. Sur l'onglet Application, cochons la checkbox « Sous-ensemble du Framework Client seulement » :



3.2 Dans l'application VB .NET

Dans notre application *AppliWPF_VB*, positionnons-nous dans les propriétés du projet. Sur l'onglet Compiler, cliquons sur le bouton « Options avancées de compilation... ». Une fenêtre apparaît, sur laquelle nous cochons la checkbox « Sous-ensemble du Framework Client seulement » :



Paramètres avancés du compilateur

Optimisations

Supprimer les contrôles de dépassement sur les entiers Activer les optimisations

Adresse de base de la DLL : &H00400000

Générer des infos de débogage : Full

Constantes de compilation

Définir la constante `DEBUG` Définir la constante `TRACE`

Constantes personnalisées :

Exemple : `Name1="Value1",Name2="Value2",Name3="Value3"`

Générer des assemblys de sérialisation : Auto

Unité centrale cible : AnyCPU

Framework cible (toutes les configurations) : .NET Framework 3.5

Sous-ensemble du Framework client seulement

OK Annuler

3.3 Modifications apportées au projet

Une fois l'option « Sous-ensemble du Framework client seulement » est activée, le fichier de configuration est modifié, dans lequel nous pouvons (entre autre) observer le contenu suivant :

```
// C# et VB
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v2.0.50727" sku="Client"/>
  </startup>
</configuration>
```

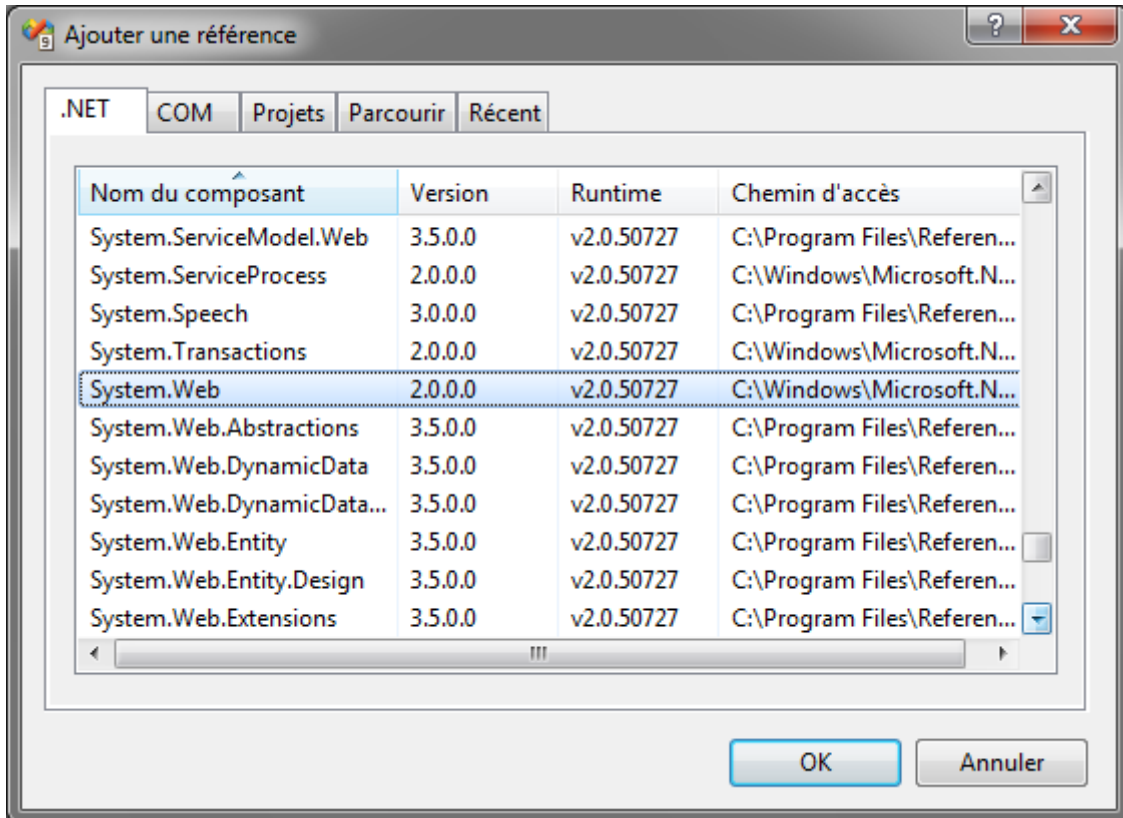
L'élément qui doit attirer notre attention est l'élément *supportedRuntime* :

- Au travers de son attribut *version*, il permet de spécifier la version du Framework .NET, sur laquelle l'application doit s'exécuter. La version spécifiée doit correspondre au nom du répertoire se trouvant dans le répertoire d'installation des différentes versions du Framework .NET (c:\Windows\Microsoft.NET\Framework). Si aucune version n'est spécifiée, alors la version du Framework .NET ciblée est celle utilisée pour générer l'application.
- La seule valeur possible pour la propriété *sku* est *Client*, qui permet de spécifier que le sous-ensemble du Framework .NET client est activé pour cette application. Si aucune valeur n'est spécifiée ou qu'une autre valeur est spécifiée, alors l'application s'exécute avec la version complète du Framework .NET spécifiée.

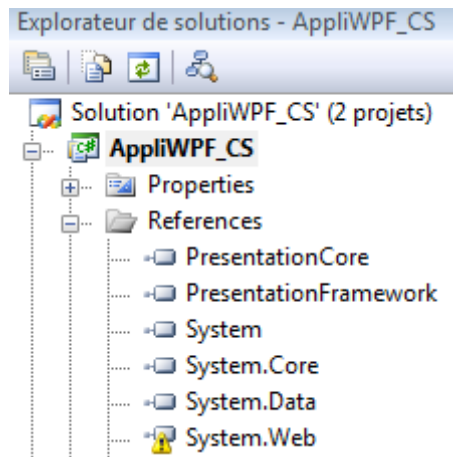


3.4 Attention aux assemblés utilisés dans votre projet

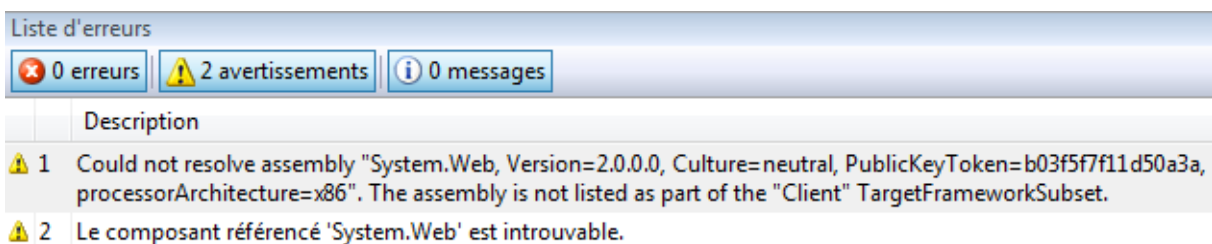
Supposons que nous souhaitons référencer l'assembly *System.Web.dll* du Framework .NET :



L'ajout de cette référence vers un composant du Framework .NET qui ne sera pas déployé provoque l'apparition d'avertissement dans les références du projet :



Cet avertissement est aussi visible lors de la compilation du projet :

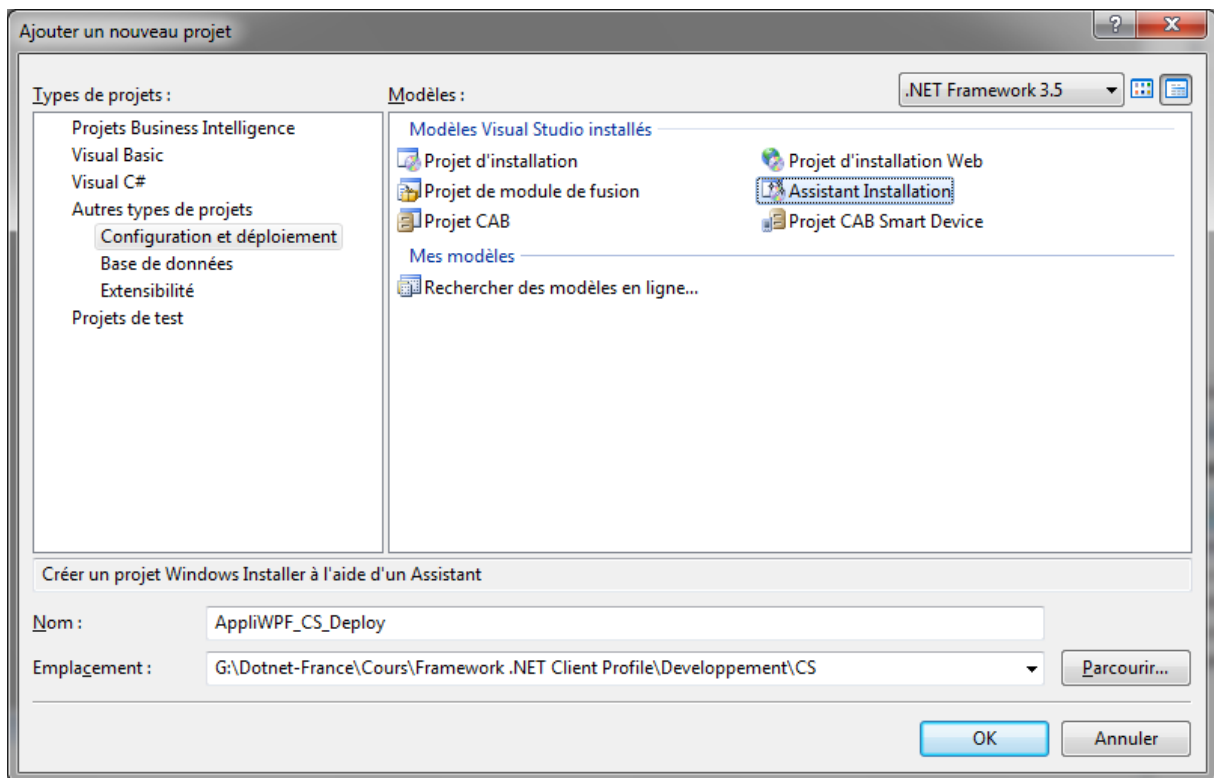




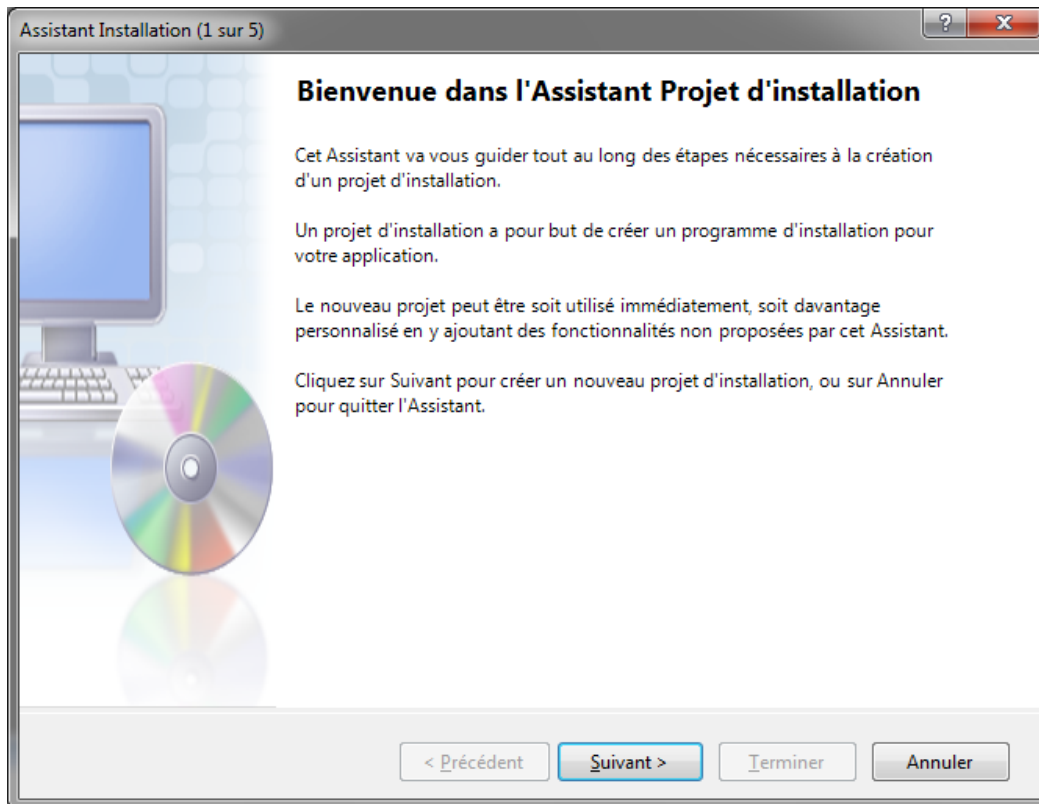
4 Déployer une application avec un sous-ensemble du Framework .NET Client et Windows Installer

4.1 Création du projet Windows Installer

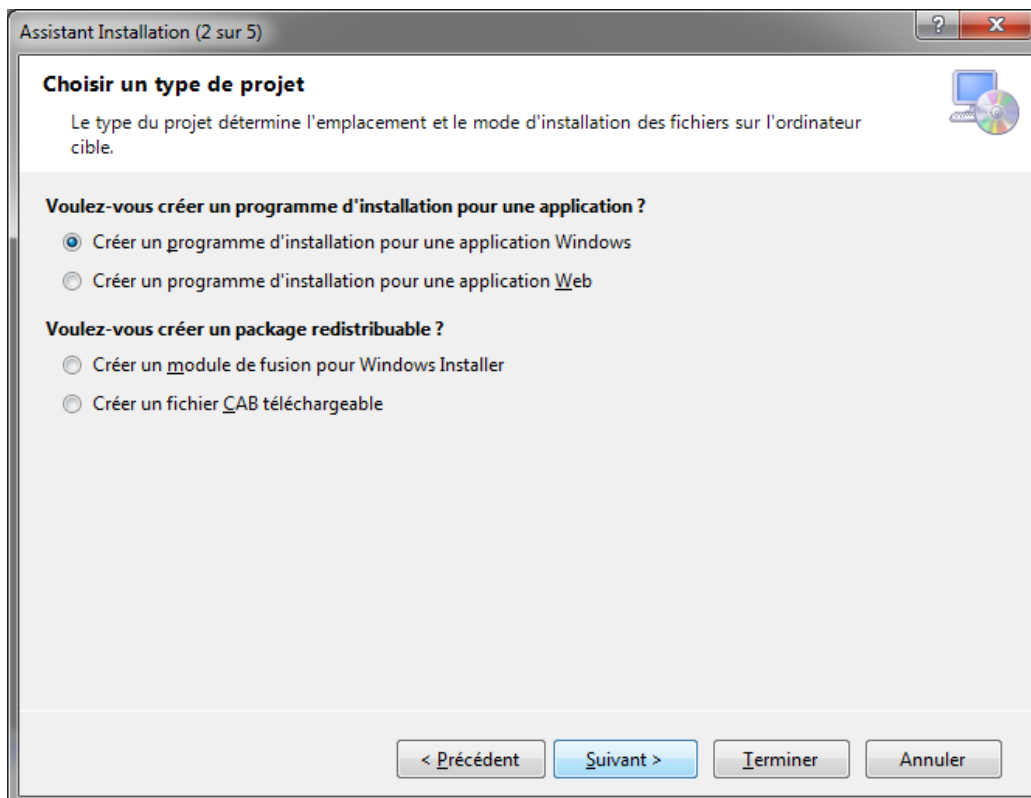
Nous allons créer un projet permettant de créer un assistant d’empaquetage et de déploiement. Seules les étapes nécessaires à la création de ce projet et à l’utilisation du Framework .NET côté client seront mises en œuvre. Pour ce faire, dans la solution, ajoutons un projet de type Assistant Installation nommé *AppliWPF_CS_Deploy* ou *AppliWPF_VB_Deploy* :



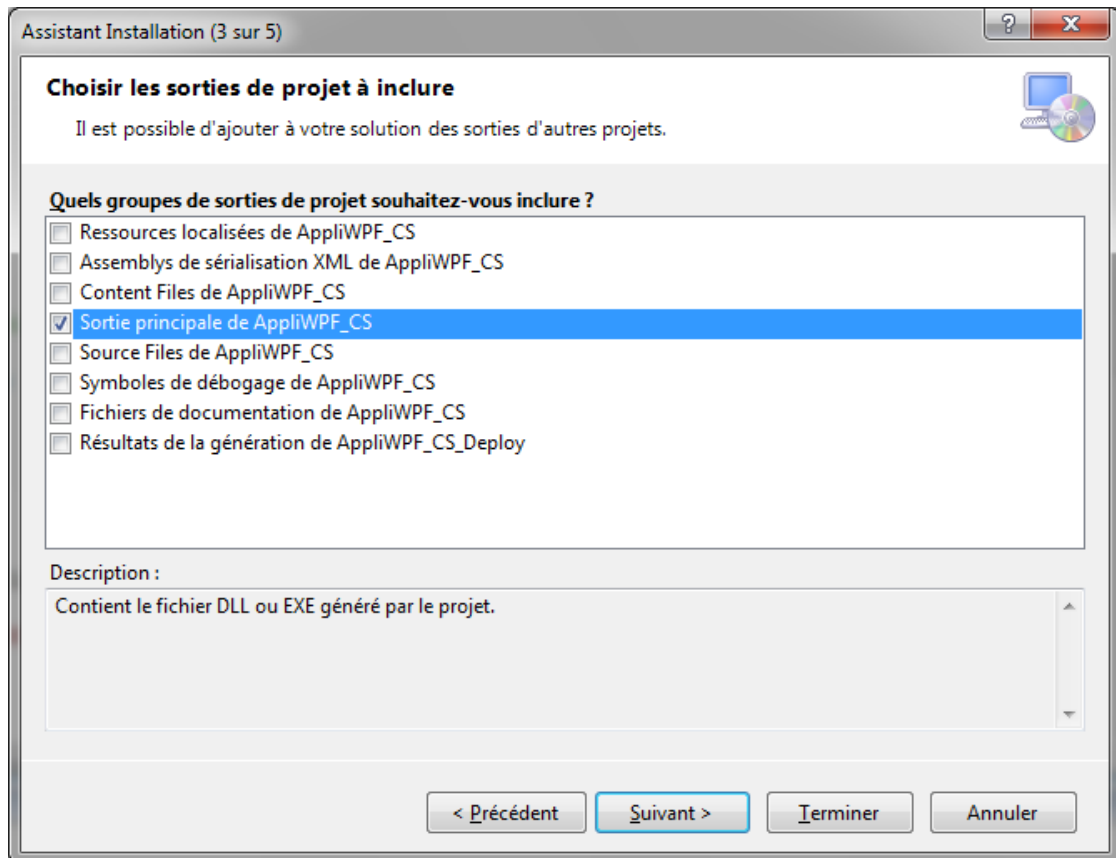
Puis cliquons sur le bouton OK. La fenêtre suivante apparaît :



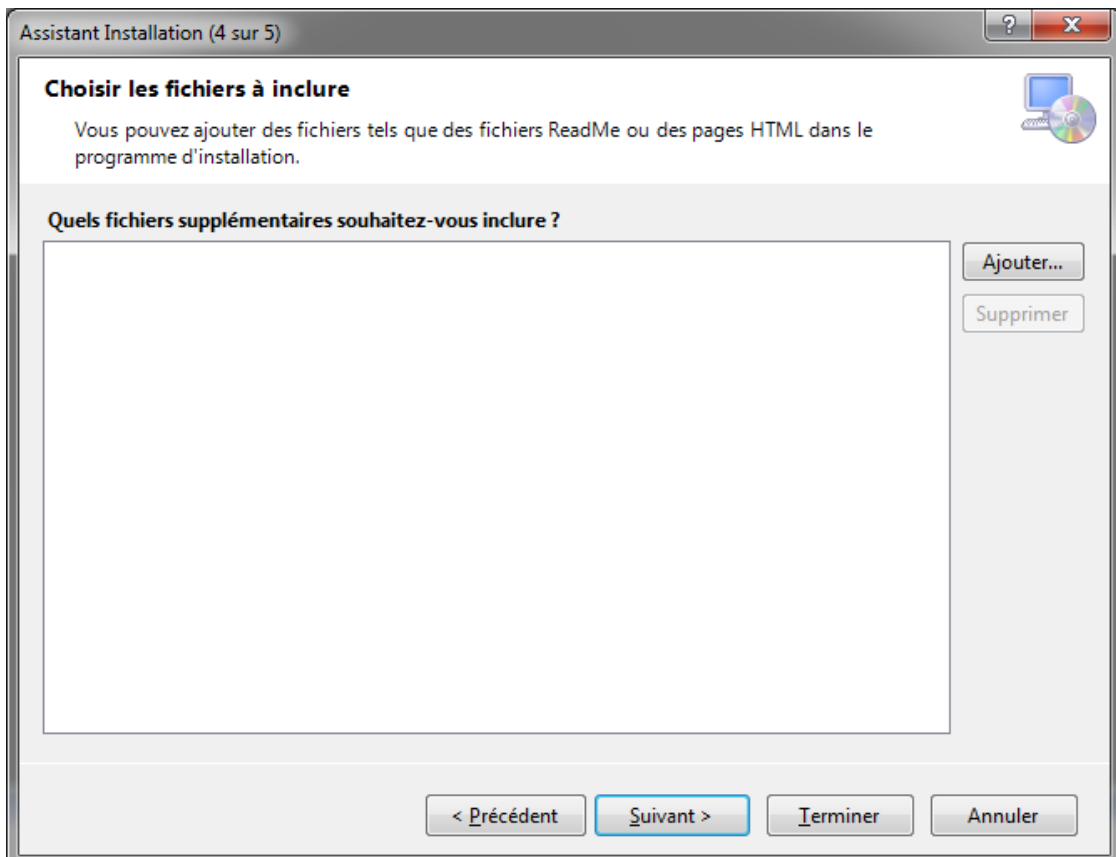
Cliquons sur bouton « Suivant ». La fenêtre suivante apparaît :



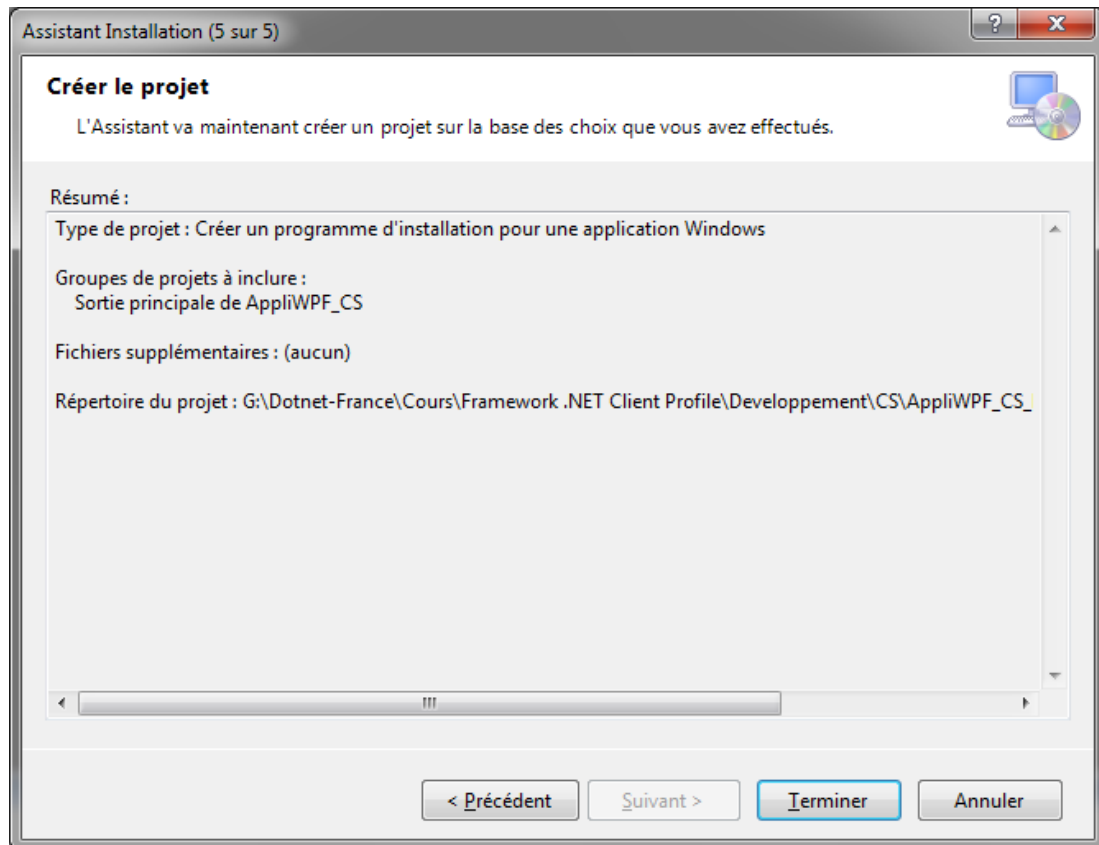
Choisissons de créer un programme d'installation pour une application Windows. Cliquons sur bouton « Suivant » :



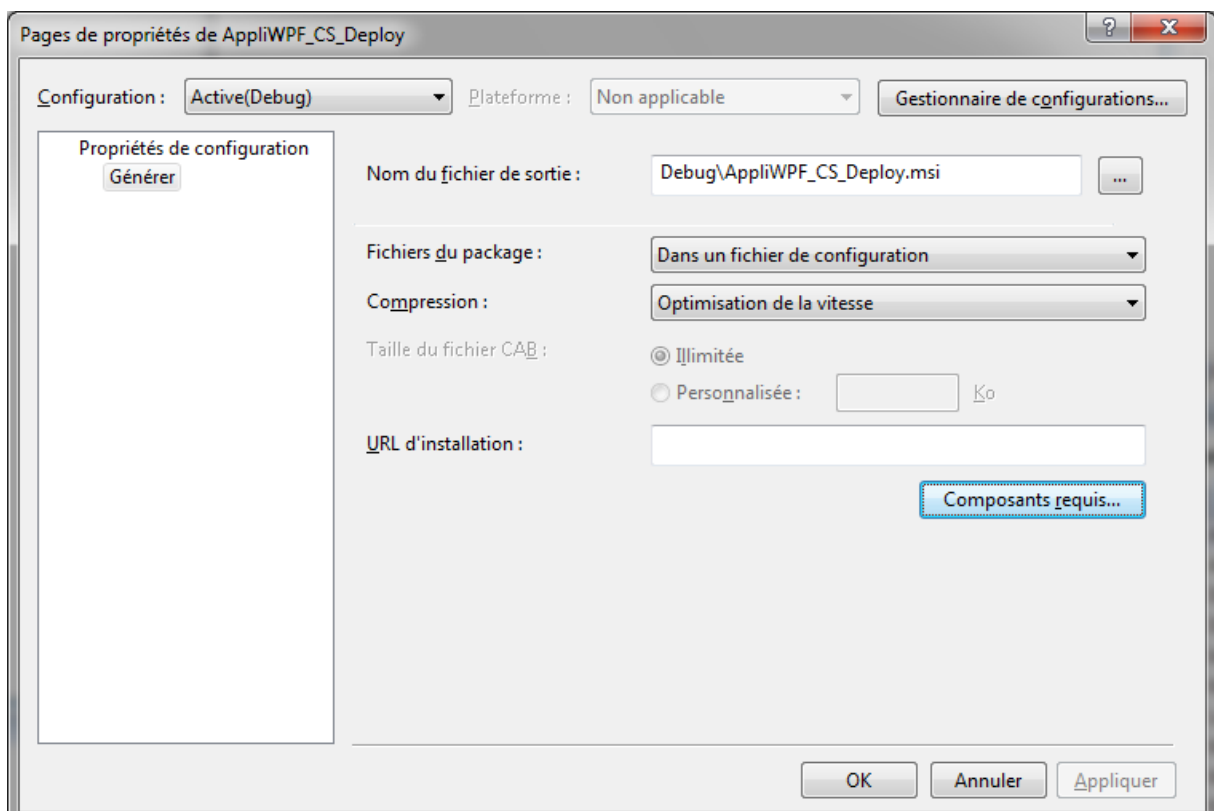
Choisissons la sortie principale du projet, de manière à déployer l'assembly correspondant à l'application, puis cliquons sur bouton « Suivant » :



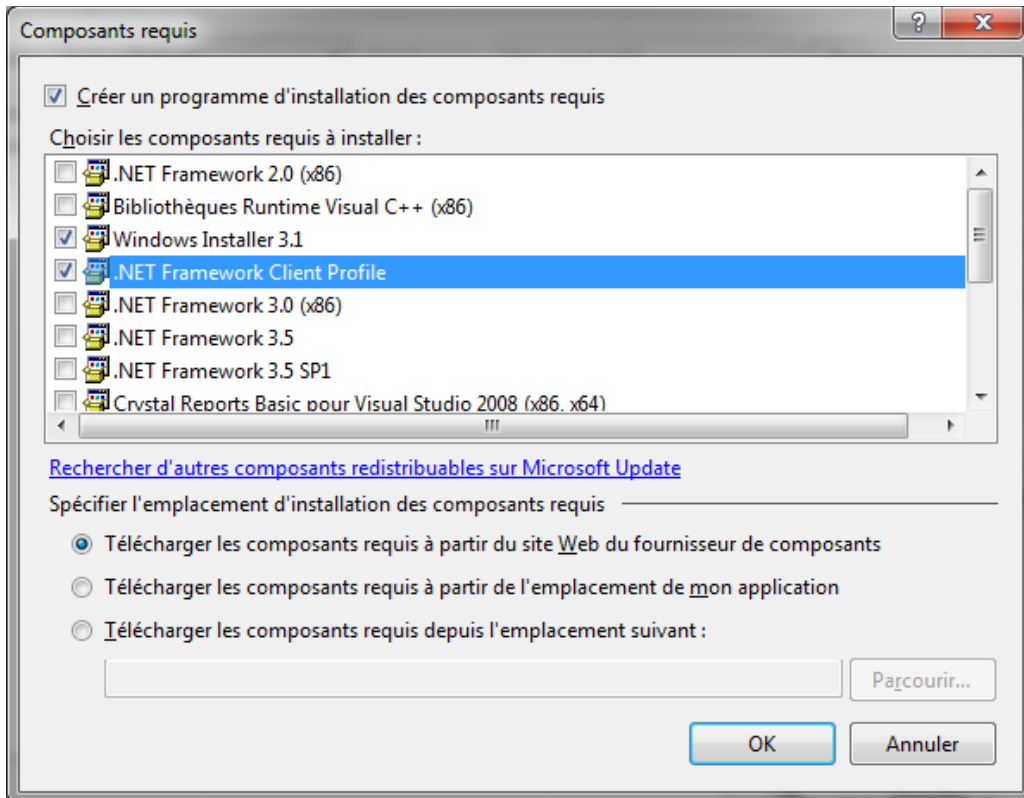
Nous ne spécifierons aucun fichier complémentaire. Cliquons sur bouton « Suivant » :



Une fois notre projet de déploiement créé, il est nécessaire de le configurer, de manière à ce qu'il n'utilise plus la version complète du Framework .NET lors du déploiement. Pour ce faire, il est nécessaire de se positionner dans les propriétés du projet, et de réaliser les étapes indiquées ci-dessous :



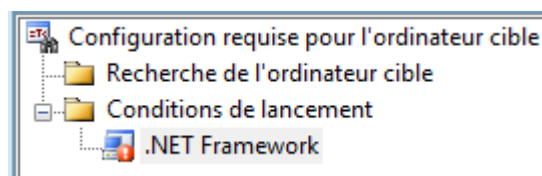
Cliquons sur le bouton « Composants Requis » :



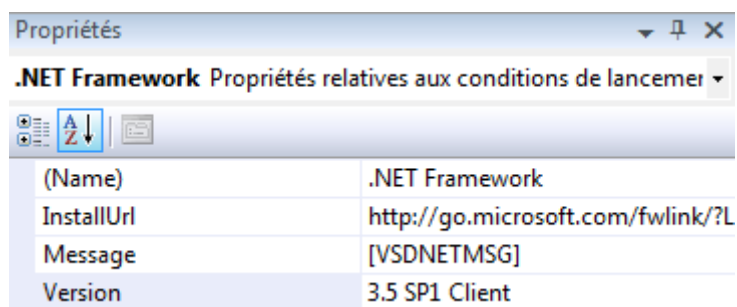
Vérifions que les composants nécessaires au bon fonctionnement de l'application soient installés sur le poste de l'utilisateur. Si un de ces composants est absent, laissons alors la possibilité de le télécharger sur le site du fournisseur (dans notre cas Microsoft). Veillons à ce que les cases à cocher :

- Correspondant à Windows Installer 3.1 et .NET Framework Client Profile sont cochées.
- Correspondant à toute autre version du Framework .NET n'est pas cochée.

Il est aussi nécessaire de modifier les conditions de lancement du projet, de manière à demander à détecter la présence du Framework .NET Client, avant d'installer l'application :



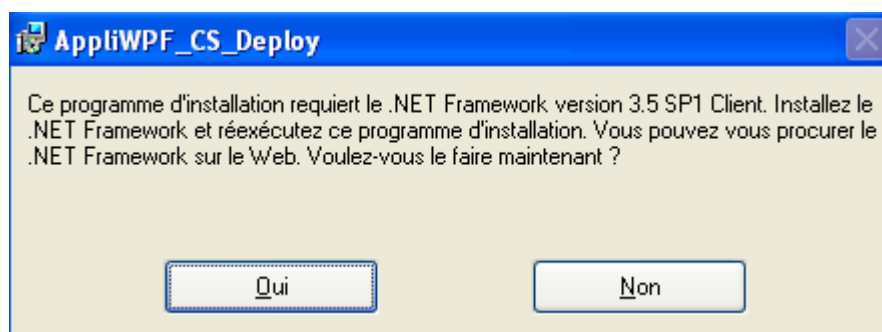
Après avoir sélectionné la condition de lancement, positionnons-nous dans la fenêtre de propriétés, sur la propriété *Version*, afin de choisir la version spécifiée ci-dessous :



.NET Framework Propriétés relatives aux conditions de lancement	
(Name)	.NET Framework
InstallUrl	http://go.microsoft.com/fwlink/?L
Message	[VSDNETMSG]
Version	3.5 SP1 Client

4.2 Installation

Il ne reste plus qu'à exécuter le programme d'installation créé. Avant le déploiement des fichiers constituant l'application, si le poste de travail de l'utilisateur ne contient pas le Framework .NET 3.5 SP1 ou le Framework .NET Client Profile, il sera proposé de télécharger et d'installer le Framework .NET Client Profile à partir du centre de téléchargement Microsoft, au travers de la fenêtre suivante :



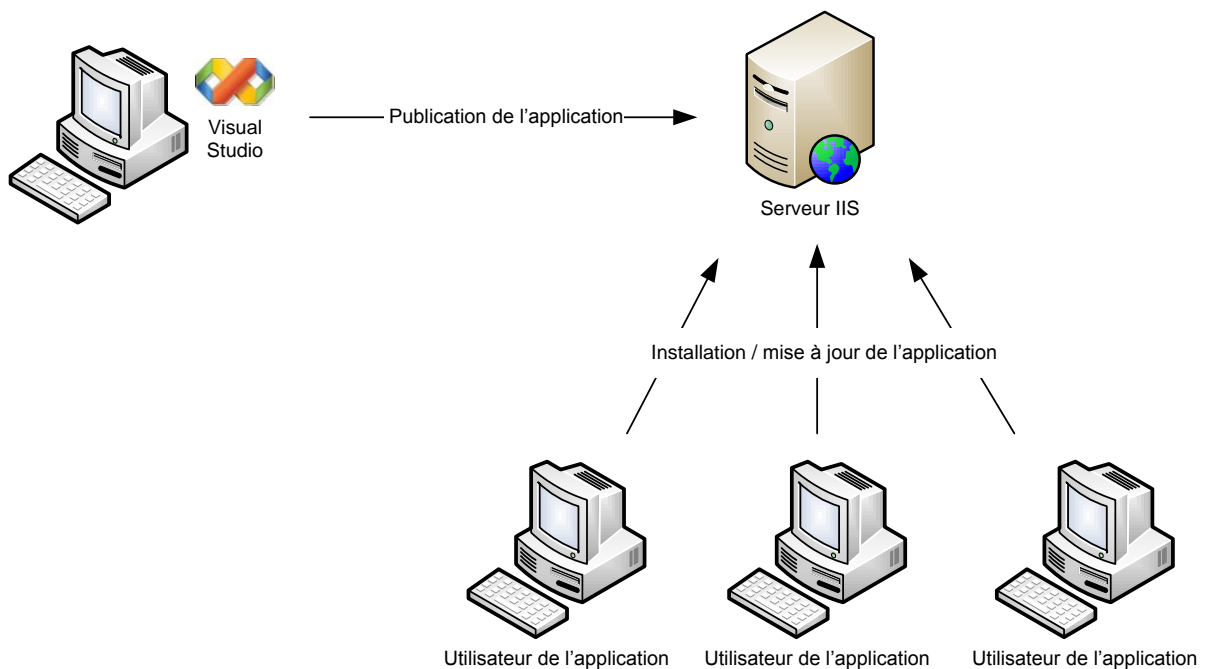


5 Déployer une application avec un sous-ensemble du Framework .NET client et un ClickOnce

Dans ce chapitre, nous ne mettrons pas l'accent sur le déploiement d'applications clientes avec ClickOnce. Ce sujet fait l'objet d'un autre cours plus général que ce cours. Nous nous consacrons sur la mise en œuvre du Framework .NET Client via ClickOnce.

5.1 Scénario de déploiement

Voici le schéma de déploiement de notre application :



Le but de notre scénario de déploiement est le suivant :

- 1 Le développeur/intégrateur de l'application :
 - a. Configure le déploiement de l'application, prépare notre package de déploiement, et signifie l'utilisation du Framework .NET Client.
 - b. Déploie l'application sur le serveur IIS.
- 2 Les utilisateurs :
 - a. Via une url, ils accèdent au programme d'installation (setup.exe) déployé sur le serveur Web, afin de l'exécuter sur leur poste de travail.
 - b. Exécutent et utilisent l'application.

5.2 Configuration du projet de déploiement

Dans le projet client (Windows ou WPF), positionnons-nous dans l'onglet *Publier* de la fenêtre de propriétés. Voici la configuration minimale que nous devons réaliser (en fonction de votre projet, vous pourriez avoir d'autres tâches à exécuter) :



- Spécifions une URI de déploiement :

The screenshot shows the 'Publier*' (Publish) tab of the Visual Studio Publish Wizard. The left sidebar contains a tree view with options: Application, Générer, Événements de génération, Déboguer, Ressources, Services, Paramètres, Chemins d'accès des références, Signature, Sécurité, and Publier* (selected). The main area is divided into sections:

- Configuration:** Configuration: N/A, Plateforme: N/A
- Emplacement de publication:** Emplacement du dossier de publication (site web, serveur ftp ou chemin d'accès du fichier): http://jrvaille-pc/AppDeploy/
- URL du dossier d'installation (si différente de ci-dessus):** (Empty field)
- Mode et paramètres d'installation:**
 - L'application est disponible en ligne uniquement
 - L'application est également disponible hors connexion (accessible depuis le menu Démarrer)
- Version de publication:**
 - Principale: 1, Secondaire: 0, Build: 0, Révision: 0
 - Incrémenter automatiquement la révision avec chaque publication

Buttons on the right: Fichiers d'application..., Composants requis..., Mises à jour..., Options...
Buttons at the bottom: Assistant Publication..., Publier maintenant

- Cliquons sur le bouton *Mises à jour...* :

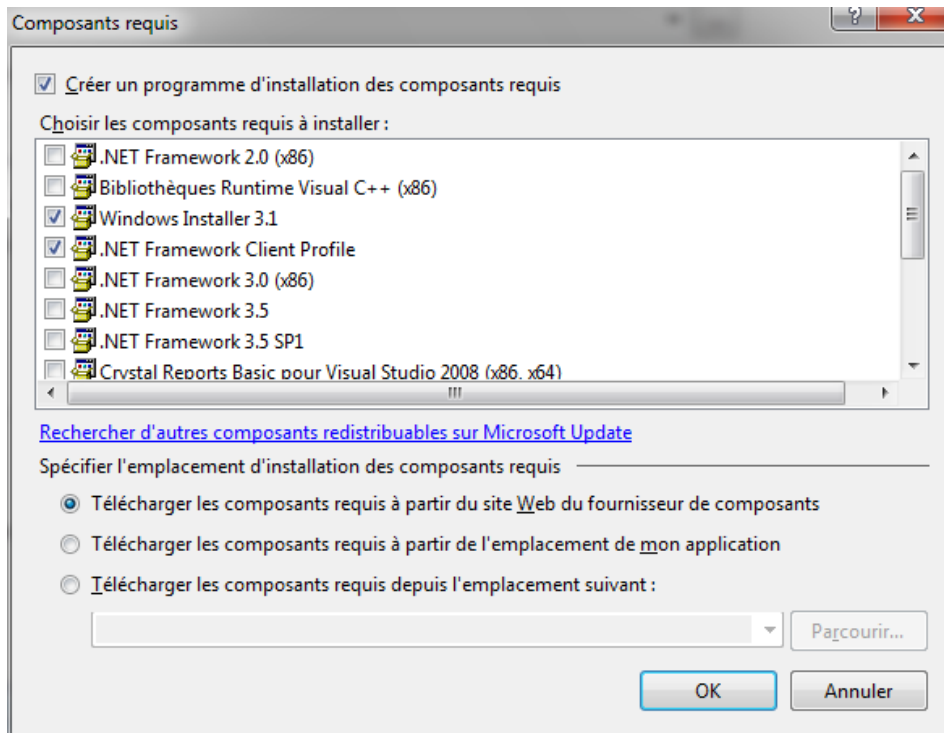
The screenshot shows the 'Mises à jour des applications' dialog box. It contains the following options:

- L'application doit vérifier la disponibilité de mises à jour
- Choisissez à quel moment l'application doit vérifier la disponibilité de mises à jour :
 - Après le démarrage de l'application
Choisissez cette option pour accélérer le démarrage de l'application. Les mises à jour ne seront pas installées avant la prochaine exécution de l'application.
 - Avant le démarrage de l'application
Choisissez cette option pour veiller à ce que les utilisateurs connectés au réseau utilisent toujours les dernières mises à jour.
- Spécifiez à quelle fréquence l'application doit vérifier la disponibilité de mises à jour :
 - Vérifier à chaque exécution de l'application
 - Vérifier toutes les : 7 jours
- Spécifier une version minimale requise pour cette application
 - Principale: , Secondaire: , Générer: , Révision:
- Emplacement de mise à jour (si différent de l'emplacement de publication) : (Empty field) Pargourir...

Buttons at the bottom: OK, Annuler

Spécifier que l'application doit être mise à jour (si nécessaire en fonction des publications de l'application sur le serveur IIS), avant le démarrage de l'application.

- Cliquons sur le bouton *Composants requis* :



Vérifions que les composants nécessaires au bon fonctionnement de l'application soient installés sur le poste de l'utilisateur. Si un de ces composants est absent, laissons alors la possibilité de le télécharger sur le site du fournisseur (dans notre cas Microsoft). Mais il est possible de spécifier une ressource réseau contenant ces composants. Veillons à ce que les cases à cocher :

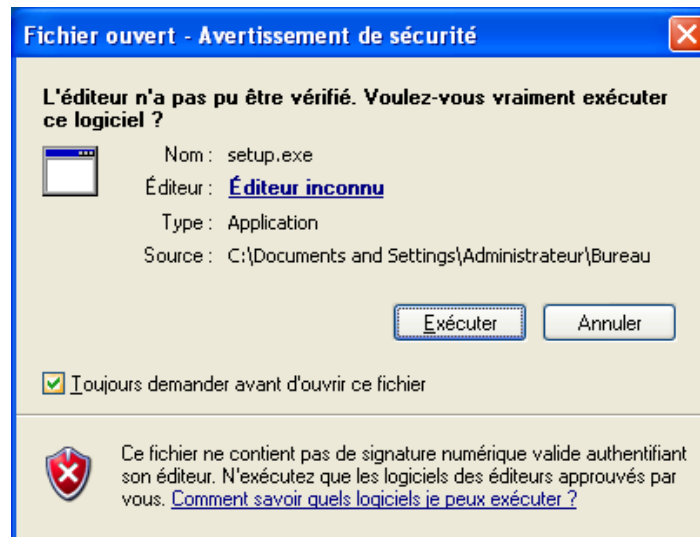
- Correspondant à Windows Installer 3.1 et .NET Framework Client Profile sont cochées.
- Correspondant à toute autre version du Framework .NET n'est pas cochée.

Enfin, après avoir fermé cette fenêtre, cliquons sur le bouton *Publier maintenant*. L'application est alors compilée, packagée dans un empaquetage de déploiement (setup), puis publiée sur le Web.

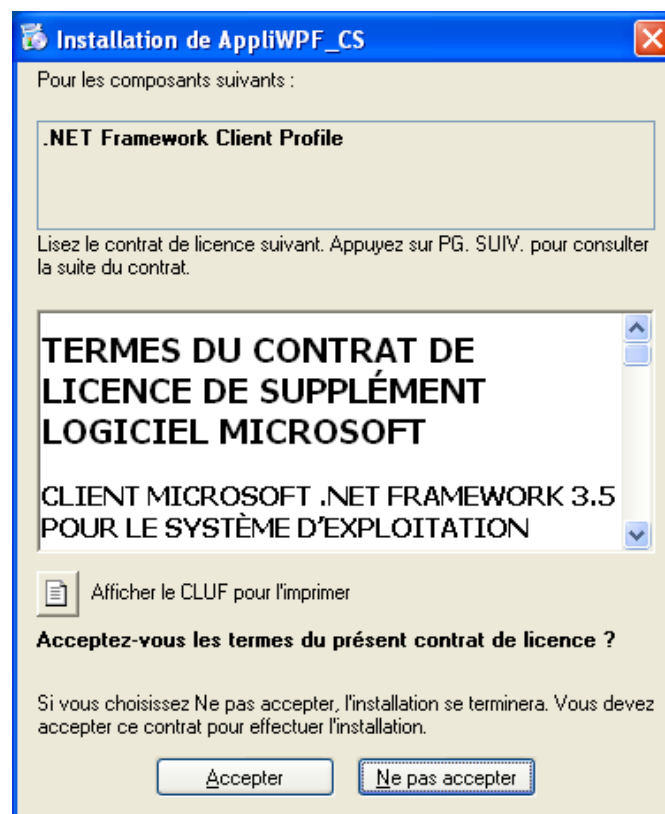
5.3 Installation

Nous allons installer l'application, sur un poste utilisateur sur lequel aucune version du Framework .NET n'a été installée.

Dans un navigateur Web, l'utilisateur accède au fichier *setup.exe*, via l'URI de déploiement spécifiée lors de la configuration du déploiement de l'application. La fenêtre apparaît alors :



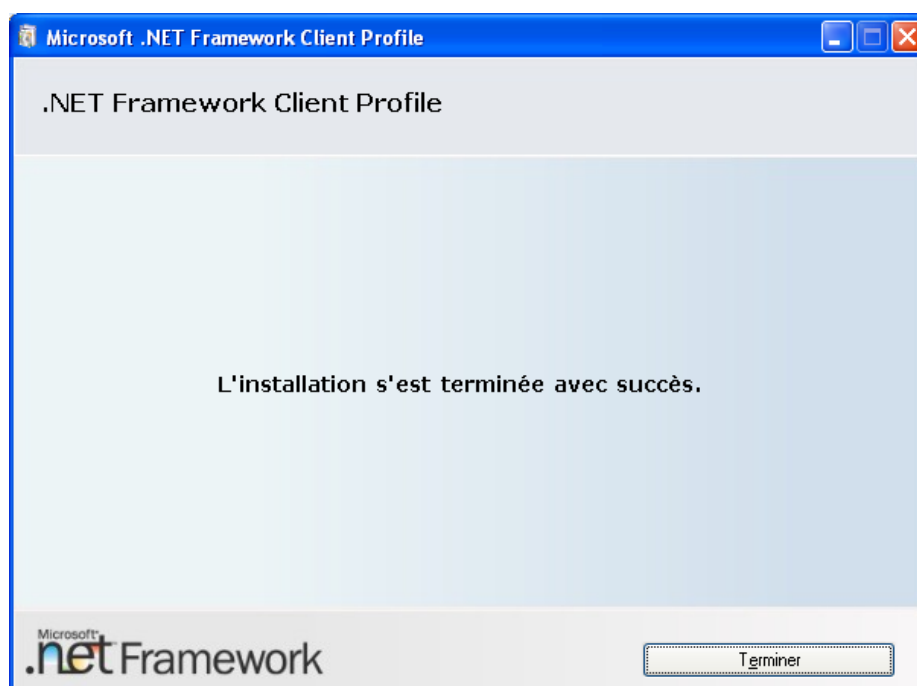
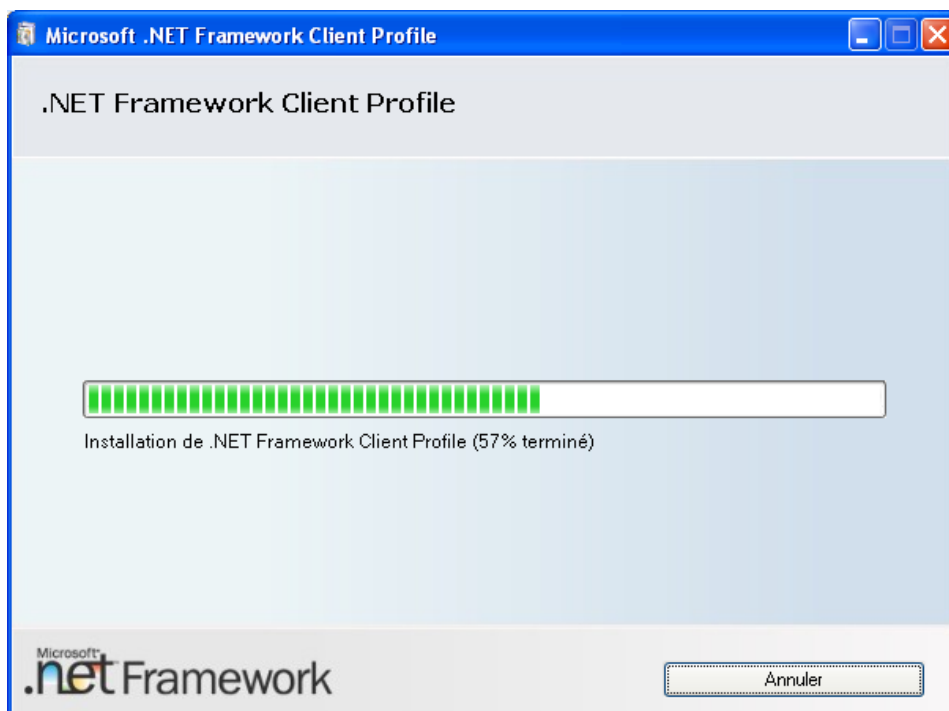
Cliquons sur le bouton *Exécuter*. Le programme d'installation détecte qu'il manque la version 3.5 (partie cliente) du Framework .NET. La fenêtre suivante apparaît alors :



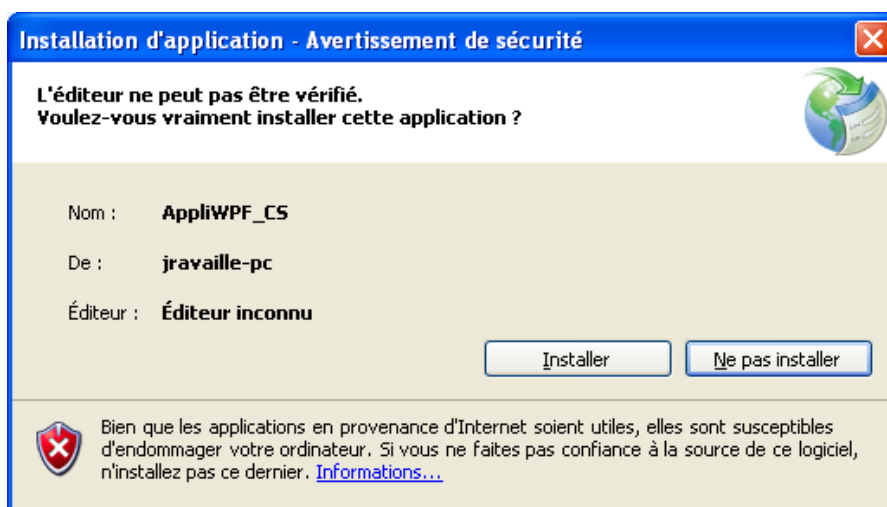
Cliquons sur le bouton *Accepter*. La fenêtre suivante apparaît alors :



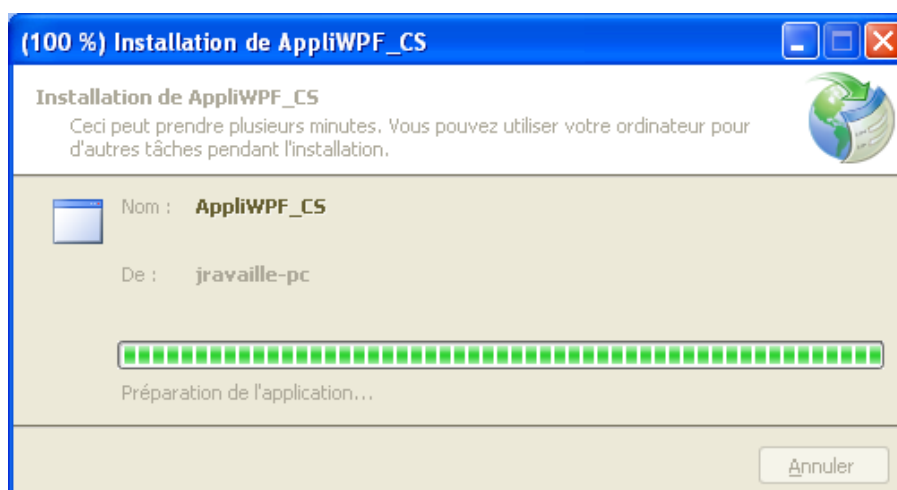
Le Framework 3.5 .NET Client Profile est en cours de téléchargement. Puis, il s'installe automatiquement :



Une fois cette tâche terminée, l'application développée peut alors être installée. Cette opération est présentée par la fenêtre suivante :



Cliquons sur le bouton Installer. La fenêtre suivante apparaît alors :



Une fois cette opération effectuée, l'application s'exécute, et le formulaire de démarrage de l'application apparaît alors :





Lors des prochaines exécutions de l'application, si des mises à jour ont été publiées, alors la dernière publication sera automatiquement proposée à l'utilisateur, en vue d'être déployée.



6 Conclusion

Ce cours vous a permis d'utiliser le Framework .NET Client pour simplifier le déploiement de vos applications .NET clientes (de type Windows Forms ou WPF), après vous l'avoir présenté. Il est sans nul doute qu'il sera pratique de l'utiliser dans le cadre d'applications clientes.