

Ma première application Windows 8

Une application de style Modern UI (anciennement nommé style Metro) est une application pouvant s'exécuter sous Windows 8. Pour vous permettre d'en découvrir la puissance et d'en apprécier sa simplicité, nous allons vous guider au travers des différentes étapes pour créer puis configurer un projet de type Modern UI sous Visual Studio 2012 et effectuer un premier déploiement sur le Windows Store.

Le développement d'une application de style Modern UI peut se faire en plusieurs langages, permettant ainsi à un grand nombre de développeurs de le faire dans celui qu'ils connaissent le mieux. Que vous maîtrisiez le développement web (HTML5/CSS3), le développement d'application .Net en WPF ou Silverlight (XAML/C#, VB, et même C++) ou encore les applications DirectX (C++), sachez que vous pouvez développer vos applications selon le style Modern UI [Fig.1]. Ces applications peuvent s'afficher sous différents modes, avec par défaut un affichage en plein écran et sans bordure. Cependant, une application de style Modern UI ne peut s'exécuter que sur Windows 8. Nous allons donc poursuivre cet article en vous présentant l'installation de Windows 8, suivie de l'installation de Visual Studio 2012. Ensuite, nous vous expliquerons comment créer votre première application en style Modern UI et terminerons par son déploiement sur le Windows Store.

> Vérification et installation des prérequis

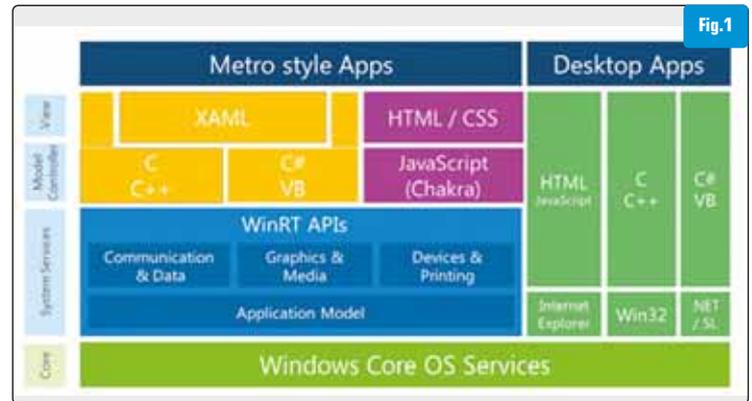
Avant de commencer, allons faire un petit tour du côté des habituels prérequis. La première étape consiste donc à vérifier et éventuellement installer et configurer votre socle applicatif.

Vous aurez besoin de :

- Windows 8 (version Professionnel, disponible sur MSDN)
- Visual Studio 2012 (version Professionnel ou Ultimate, disponible en RTW sur MSDN)

Installation de Windows 8 et de Visual Studio 2012

Démarez l'installation de Windows 8 RTM. Sélectionnez la langue



que vous désirez et entrez alors la clé produit que vous avez obtenue sur le site MSDN. Sachez que si vous n'avez pas de clé produit, vous pouvez quand même utiliser Windows 8 RTM pendant 30 jours en version d'évaluation. Une fois l'installation terminée, vous devrez donner un nom à votre ordinateur et choisir le jeu de couleur de votre interface Modern UI. Sur l'écran suivant, cliquez sur « Utiliser la configuration rapide », et pour terminer, saisissez les informations de votre compte Windows Live. Votre installation de Windows 8 est alors terminée, et vous voilà sur votre interface Modern UI [Fig.2].

Pour continuer, vous devez vous procurer la version Professionnel ou la version Ultimate de Visual Studio 2012. Elles sont également disponibles en téléchargement sur le site MSDN.

Lancez alors l'installation de Visual Studio 2012 et suivez l'assistant afin de procéder à l'installation complète de Visual Studio 2012. Une fois l'installation terminée, nous pouvons passer à la suite.

> Création du Projet sous Visual Studio 2012

Dans l'interface Modern UI de Windows 8, vous trouverez une tuile Visual Studio 2012. Lancez-le et si c'est le premier démarrage, sélectionnez les paramètres d'environnement par défaut (« Paramètres de développement Visual C# » dans notre cas).

Une fois Visual Studio lancé, sur la page d'accueil, cliquez sur « New Project » ou allez dans « File » ? « New » ? « Project ». Créez alors un nouveau projet de type « Blank App (XAML) » dans la rubrique « Windows Store » [Fig.3].

A la création de votre premier projet d'application Windows 8, Visual Studio vous demandera une licence de développeur Windows 8 [Fig.4]. Cliquez alors sur « J'accepte » et connectez-vous avec votre compte Windows Live afin d'obtenir une licence temporaire de développeur [Fig.5].

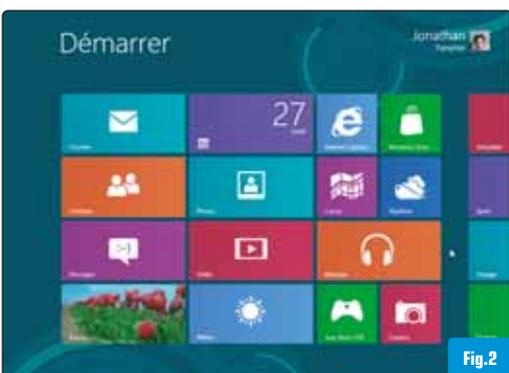


Fig.2



Fig.3



Fig.4

Une fois la procédure terminée, vous obtiendrez une confirmation avec la durée de validité de votre licence de développeur Windows 8 [Fig.6]. Vous voilà alors prêt à créer votre première application Modern UI !

> Exemple d'application : Image Viewer

Nous allons maintenant créer une application « Image Viewer ». Celle-ci affichera les images que vous possédez dans votre bibliothèque d'images (votre dossier « Mes Images »). Pour cela, ouvrez le projet que vous avez créé dans l'étape précédente.

La première chose à faire est d'autoriser votre application à accéder à votre répertoire « Mes Images ». Si vous ne le faites pas, vous obtiendrez un message d'erreur [Fig.7].

En effet, la politique de sécurité de Windows 8 vous interdit par défaut l'accès à la bibliothèque d'images. Dans l'explorateur de solution, double-cliquez sur le fichier « Package.appxmanifest ».

Allez ensuite sur l'onglet « Capacités », cochez la case correspondant à « Accéder à la bibliothèque d'images » [Fig.8] et sauvegardez le fichier. Ajoutez une nouvelle classe « ImageItem » à la solution. Pour chacune des images de la bibliothèque, on eninstanciera alors une qui contiendra le nom d'une image, son chemin et une courte description :

```
public class ImageItem
{
    public string ImageName { get; set; }
    public BitmapImage Image { get; set; }
    public string ImageDescription { get; set; }
}
```

Dans le fichier MainPage.xaml, ajoutez une GridView :

```
<Grid x:Name="LayoutRoot" Background="{StaticResource ApplicationPageBackgroundThemeBrush}">
    <GridView x:Name="GridImages" ItemsSource="{Binding}" Foreground="#FFE6D52E" Margin="40,111,56,143" >
        <GridView.Background>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop Color="Black"/>
                <GradientStop Color="#FF1A59E2" Offset="1"/>
            </LinearGradientBrush>
        </GridView.Background>
        <GridView.ItemTemplate>
            <DataTemplate>
                <StackPanel Orientation="Vertical">
                    <StackPanel.Background>
```

```
                    <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                        <GradientStop Color="Black"/>
                        <GradientStop Color="#FF443585" Offset="1"/>
                    </LinearGradientBrush>
                </StackPanel.Background>
                <TextBlock Text="{Binding ImageName}" Width="250" FontSize="25" Padding="10"
                    TextTrimming="WordEllipsis"/>
                <Image Height="100" Width="250" Source="{Binding Path=Image}" />
                <TextBlock Text="{Binding ImageDescription}" FontSize="15"/>
            </StackPanel>
        </DataTemplate>
    </GridView.ItemTemplate>
</GridView>
</Grid>
```

Allez ensuite dans le fichier MainPage.xaml.cs, et créez la méthode « InitializeGrid() » :

```
public MainPage()
{
    this.InitializeComponent();
    InitializeGrid();
}

private async void InitializeGrid()
{
    var files = await KnownFolders.PicturesLibrary.GetFilesAsync();
    if (files.Count == 0)
    {
        var dialog = new MessageDialog("Désolé, mais je n'ai trouvé aucune image à afficher.");
        await dialog.ShowAsync();
    }
    else
    {
        int index = 1;
        var lstImage = new List<ImageItem>();
        foreach (var file in files)
        {
            var item = new ImageItem();
            item.ImageName = file.Name;
```

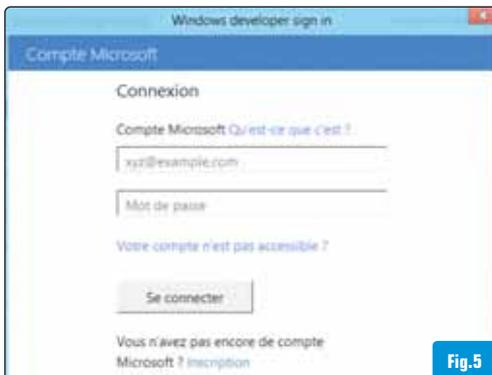


Fig.5

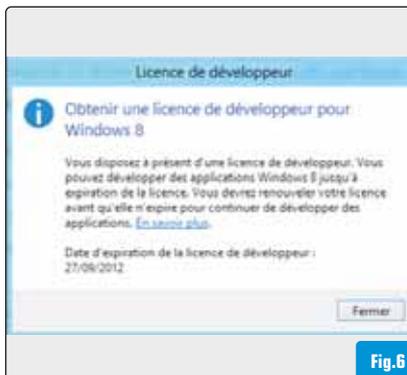


Fig.6

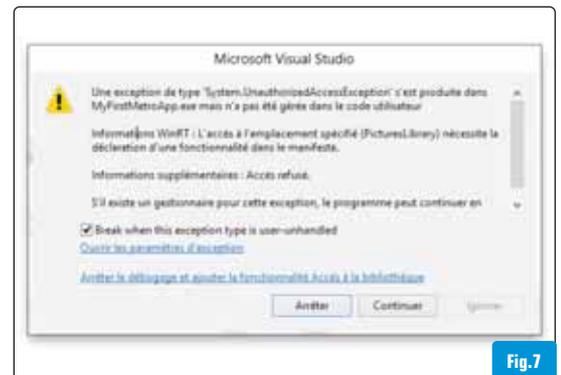


Fig.7

```

var fileStream = await file.OpenAsync(Windows.Storage.
FileAccessMode.Read);
var image = new BitmapImage();
await image.SetSourceAsync(fileStream);
item.Image = image;

item.ImageDescription = «Fichier N°» + index;
lstImage.Add(item);

index++;
}
GridImages.DataContext = lstImage;
}
}

```

Cette méthode remplira la GridView avec les images présentes dans votre dossier « Mes Images » [Fig.9] ou vous affichera un message si aucune image n'est trouvée dans ce répertoire [Fig.10].

> Déclaration des fonctionnalités d'une application Windows 8

Pour accéder aux ressources utilisateur ou aux périphériques connectés, il faut déclarer les fonctionnalités des applications Windows 8. Cela se fait comme dans l'exemple que nous avons vu précédemment, en éditant le fichier « Package.appxmanifest » (manifeste de package). Cette déclaration est requise si vous souhaitez publier votre application sur le Windows Store.

En effet, quand vous soumettez votre application, elle fait tout d'abord l'objet d'un contrôle afin de s'assurer que les fonctionnalités que vous avez déclarées correspondent à la description que vous en avez faite, et avant toute installation, ces dernières seront notifiées à l'utilisateur qui souhaitera la télécharger. Il existe deux types de fonctionnalités à déclarer dans le manifeste de package : les fonctionnalités à usage général, qui sont celles qui s'appliquent à la majorité des cas d'utilisation dans le cadre des applications de style Modern UI, et celles à usage spécial qui sont destinées à des scénarios très spécifiques. L'utilisation des fonctionnalités à usage spécial est très limitée et sujette à des contrôles complémentaires dans le cadre du déploiement sur le Windows Store.

Les fonctionnalités à usage général proposent l'accès à l'audiothèque, à la bibliothèque d'images, à la vidéothèque, aux stockages amovibles, au microphone, à la webcam, à la localisation, aux fonctionnalités de proximité, aux connexions internet et aux réseaux domestiques ou d'entreprise. Celles à usage spécial, quant à elles, donnent accès à la bibliothèque de documents, à l'authentification en entreprise et à l'utilisation de certificats utilisateurs partagés.

> Déploiement sur le Windows Store

Qu'est-ce que le Windows Store ?

Le Windows Store est la plateforme unique à laquelle tous les utilisateurs de Windows 8 ont accès pour télécharger des applications au style Modern UI. Grâce à lui, Microsoft a la possibilité de contrôler et valider chaque application avant qu'elle ne soit mise à disposition des utilisateurs de Windows 8, et ainsi en garantir la qualité.

Le contrôle des applications par Microsoft

Une fois une application soumise sur le Windows Store, elle est contrôlée avant d'être autorisée au téléchargement ou à la vente. Ce contrôle, effectué par les équipes de Microsoft, vérifie que cette dernière respecte bien les règles définies par Microsoft, qu'elle est fonctionnelle, sécurisée et fiable pour l'utilisateur. Ce test la validera alors techniquement, certifiera qu'elle ne contient pas de virus et en validera le contenu.

Création de votre compte sur le Windows Store

Pour commencer, il vous faut vous rendre sur le site « Dev Center » de Microsoft (<http://msdn.microsoft.com/en-us/windows/apps>). Vous pourrez alors créer votre compte Windows Store. Il en existe deux types, à savoir les comptes individuels et les comptes d'entreprises. A noter que les comptes d'entreprises sont les seuls à pouvoir soumettre une application de bureau ou une application utilisant les fonctionnalités à usage spécial.

Publier votre application

Pour publier votre application, après vous être connecté, cliquez sur le bouton « Submit an app ». Vous aurez alors un ensemble d'étapes à suivre, à commencer par la réservation du nom de votre application, suivie de sa description et de son envoi aux équipes de Microsoft afin qu'elles la testent et vous la valident.

CONCLUSION

Sachez que vous pouvez bien évidemment créer des applications beaucoup plus créatives, complexes et abouties. Comme vous l'avez constaté, la complexité ne réside pas dans l'implémentation, mais plutôt dans la recherche de bonnes idées et de nouvelles utilisations des interfaces utilisateurs (NUI). Voilà, vous avez en mains les principales étapes pour créer votre première application Modern UI et la proposer au grand public dans le Windows Store. A vous de jouer maintenant !



Jonathan Pamphile
Consultant chez Cellenza
Cellenza - Software Development Done Right



Jason De Oliveira
Practice Manager & Solutions Architect / MVP C#
.Net Rangers by Sogeti
Son Blog: <http://jasondeoliveira.com>



Fig.8



Fig.9



Fig.10