



eBook Gratuit

APPRENEZ

windows-10-universal

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#windows-

10-universal

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec Windows-10-universal.....	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Créer un nouveau projet (C # / XAML).....	5
Contenu du projet par défaut.....	6
Modifier la vue.....	7
Lancer l'application.....	8
Crédits.....	9

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [windows-10-universal](#)

It is an unofficial and free windows-10-universal ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official windows-10-universal.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec Windows-10-universal

Remarques

Cette section fournit une vue d'ensemble de Windows-10-universal et explique pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tout sujet important dans windows-10-universal et établir un lien avec les sujets connexes. La documentation de windows-10-universal étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

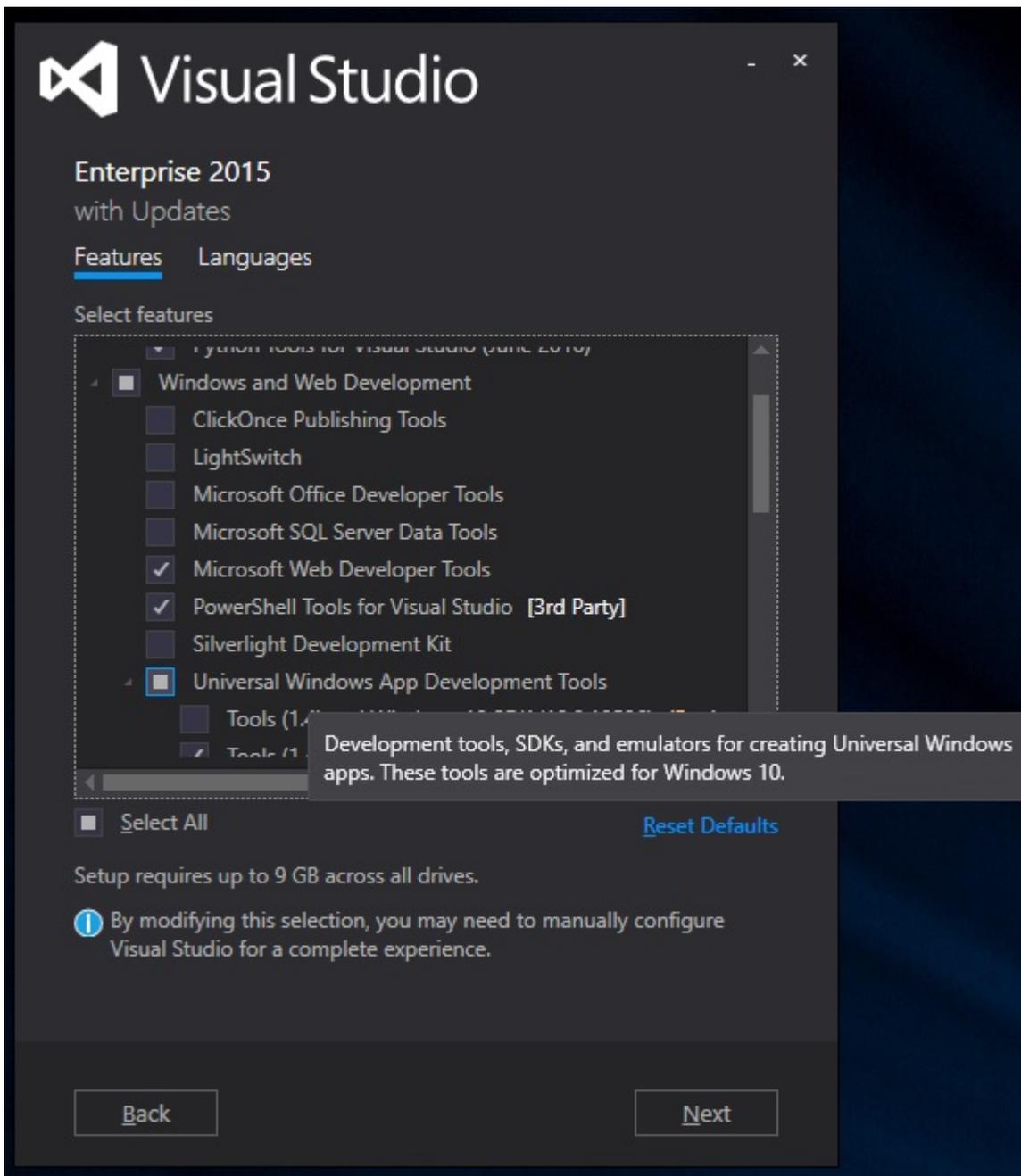
Exemples

Installation ou configuration

Si vous ne disposez pas de Visual Studio 2015, accédez à <https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx> et sélectionnez Visual Studio Community. C'est une version gratuite qui possède toutes les capacités requises pour développer une application Windows 10. Vous pouvez installer Enterprise Edition si vous disposez de la licence requise. Puis lancez le programme d'installation.

Si Visual Studio 2015 est déjà installé, dans le `Windows Control Panel`, accédez à *Programmes et fonctionnalités* et dans la liste cliquez avec le bouton droit sur *Microsoft Visual Studio 2015* (ce nom peut changer en fonction de la version installée de Visual Studio 2015) et sélectionnez *Modifier*.

Dans la boîte de dialogue Installation de Visual Studio, cliquez sur `Modify` et sous *Windows et Développement Web*, installez tous les *outils de développement Universal Windows App*. Il installera le SDK Windows 10, un émulateur mobile Windows 10 (optimisé par Hyper-V) et Visual Studio requis pour créer des applications Windows 10.



Une fois l'installation terminée, vous pouvez créer un nouveau projet Windows 10.

Configuration requise: Windows 10

Visual Studio 2015 (fonctionnalité Outils de développement d'applications Windows universelles)

Instructions pour activer les outils de développement Universal Windows App lors de l'installation pendant Visual Studio 2015.

i. Durant l'installation, il est nécessaire de sélectionner Personnaliser et de vérifier les outils de développement Windows Universal.

ii. Si vous disposez déjà de Visual Studio 2015, vous pouvez désormais ajouter ces outils à votre installation Visual Studio existante. Vous pouvez exécuter le programme d'installation ou ouvrir Programmes et fonctionnalités à partir du Panneau de configuration, sélectionner Visual Studio et cliquer sur Modifier. Ensuite, dans la configuration, cliquez sur Modifier et sélectionnez les outils

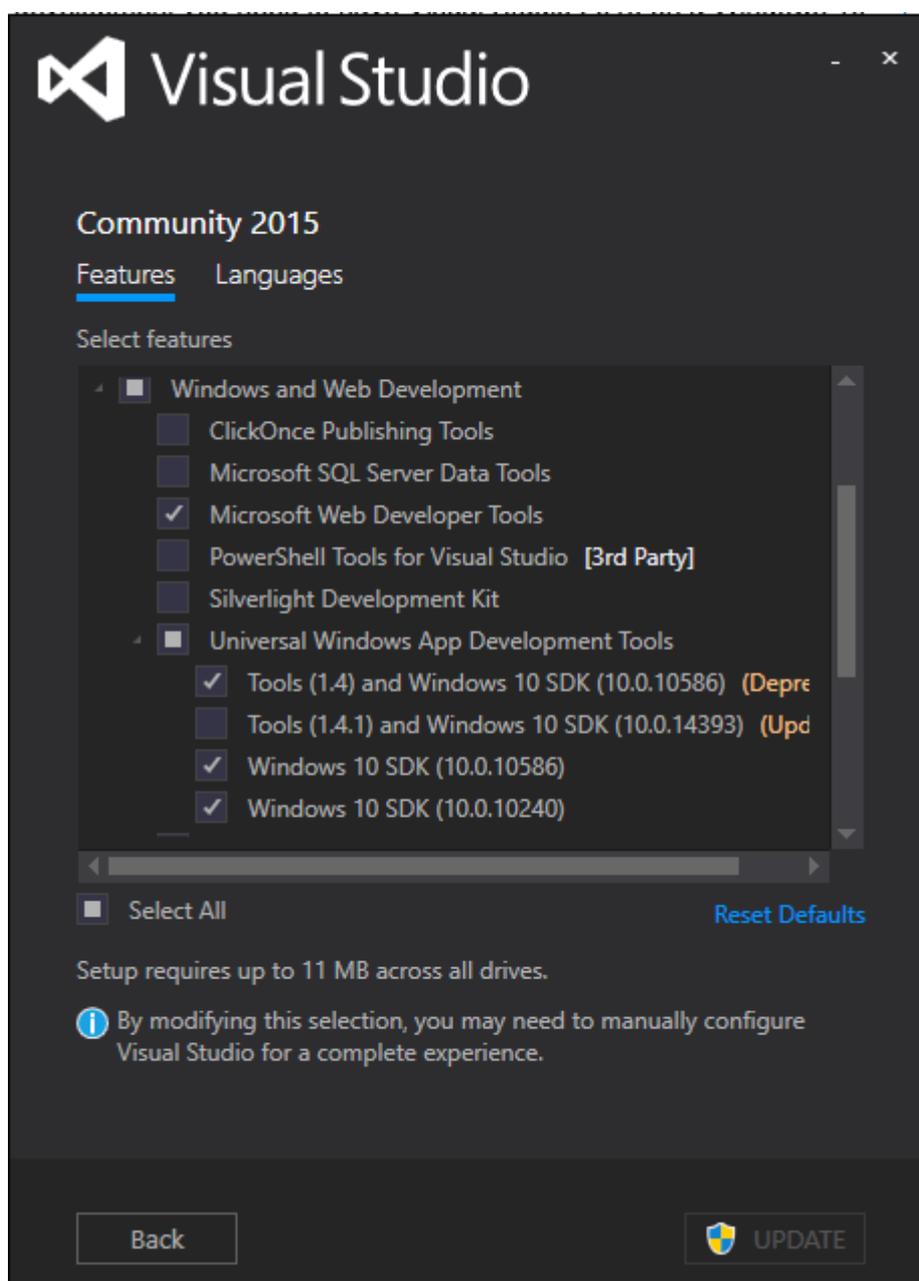
pour les applications Windows universelles.

iii. Après avoir installé les outils, vous pouvez créer un projet à l'aide de nouveaux modèles de projet (Windows-> Universal) dans Visual Studio 2015 dans la langue de votre choix - C #, VB, C ++ ou JavaScript.

Pour démarrer avec le développement d'applications Windows 10 Universal, vous devez disposer de Visual Studio 2015 sur un ordinateur Windows 10. Vous pouvez acheter Windows 10 à partir d'[ici](#) . Vous pouvez télécharger Visual Studio 2015 à partir d' [ici](#) . Vous pouvez utiliser les versions Community, Professional ou Enterprise de Visual Studio. Certaines fonctionnalités seront limitées dans la version Express.

Vous pouvez trouver un guide détaillé sur l'installation de Windows 10 [ici](#) .

Lors de l'installation de Visual Studio, choisissez Personnalisé et sous «Développement Windows et Web», sélectionnez «Outils de développement d'applications Windows universels» dans la



sélection des fonctionnalités.

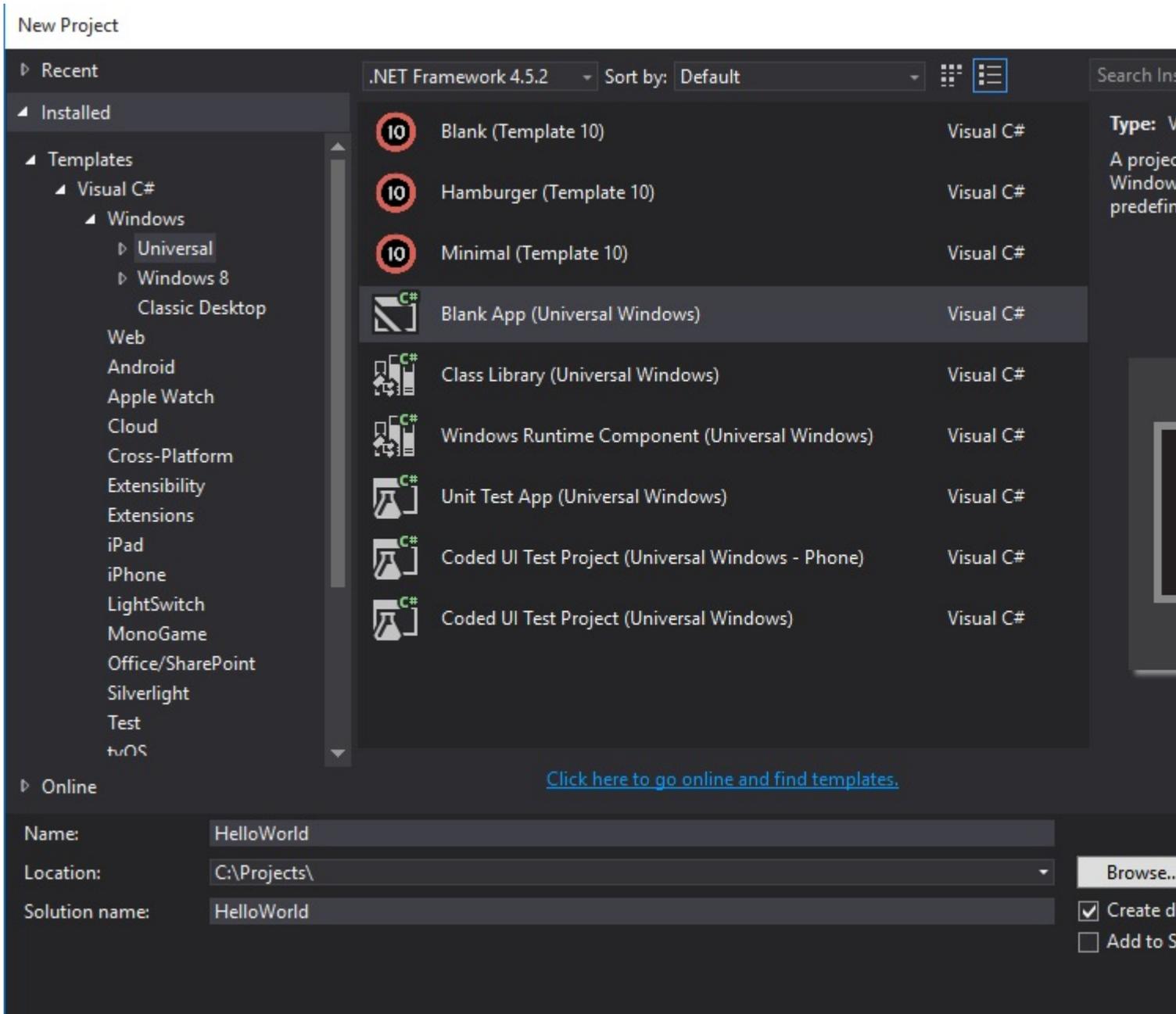
Lorsque vous avez terminé, vous êtes prêt à développer des applications Windows universelles!

Créer un nouveau projet (C # / XAML)

Après avoir lancé Visual Studio 2015, accédez à `File → New → Project`. Dans la boîte de dialogue *Nouveau projet*, parcourez l'arborescence des modèles pour `Visual C# → Windows → Universal` et sélectionnez `Blank App (Universal Windows)`.

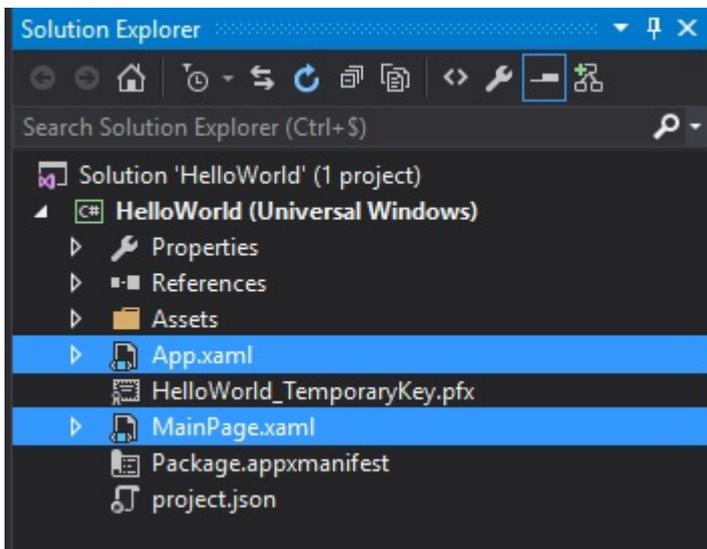
Ensuite, nous devons remplir le formulaire pour décrire l'application:

1. **Nom** : c'est le nom de l'application qui sera affichée à l'utilisateur. Définissez-le sur `HelloWorld` ou utilisez un titre personnalisé.
2. **Lieu** : indique où le projet sera stocké
3. **Nom de la solution** : il s'agit d'une sorte de conteneur de projets regroupant plusieurs projets liés à la même application (par exemple, une solution peut être composée d'un projet d'interface utilisateur et d'un projet de modèle). Vous pouvez mettre le même `Name` que votre projet initial.



Contenu du projet par défaut

Vous obtiendrez un projet avec les fichiers suivants:



1. **Package.appxmanifest** : décrit les propriétés de votre application. Il contient des paramètres d'interface utilisateur tels que son nom display, son logo, les rotations prises en charge. Et il contient également des paramètres techniques tels que le point d'entrée de l'application (qui est la classe `App` par défaut). Enfin, il répertorie également les autorisations requises par votre application dans l'onglet *Capabilities* ; Par exemple, si vous souhaitez utiliser la webcam dans votre application, vous devrez vérifier les capacités correspondantes.
2. **App.xaml / App.xaml.cs** : la classe `App` est le point d'entrée par défaut de votre application. Les fichiers xaml peuvent contenir des ressources partagées dans l'ensemble de l'application, telles que les paramètres de styles ou l'instance d'une classe que vous souhaitez partager, comme un localisateur ViewModel. Les fichiers code-behind contiennent tout le code de démarrage de l'application. Par défaut, il implémente la méthode `OnLaunched` qui est appelée par l'utilisateur final. Il initialise la fenêtre et navigue jusqu'à la première page de l'application (par défaut la classe `MainPage`).
3. **MainPage.xaml / MainPage.xaml.cs** : c'est la page initiale de notre application. Il ne contient qu'une grille vide qui est un contrôle de mise en page.

Modifier la vue

Ouvrez le `MainPage.xaml` et remplacez le contrôle Grid par

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="auto" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <Button Click="Button_Click">Say Hello !</Button>
  <TextBlock Grid.Column="1"
    VerticalAlignment="Center"
    x:Name="myText"
    Text="Click the button." />
</Grid>
```

Cela va créer une grille avec deux colonnes. La première colonne en tant que largeur définie sur

`auto` ce qui signifie qu'elle sera automatiquement définie en fonction de la taille de ses enfants. La deuxième colonne s'étirera pour remplir l'espace restant dans la fenêtre.

Cette grille contient deux éléments:

- un `Button` situé dans la première colonne. L'événement `click` est `Button_Click` à la méthode `Button_Click` sur le code-behind et sa légende `Text` est *"Say Hello!"* .
- un `TextBlock` qui se trouve dans la deuxième colonne. Son texte est défini sur *"Cliquez sur le bouton"* . Et nous avons donné un nom à ce contrôle à l'aide de l'attribut `x:Name` . Ceci est nécessaire pour pouvoir utiliser le contrôle dans le code-behind.

Dans le `MainPage.xaml.cs` , ajoutez le code suivant:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    this.myText.Text = "Hello World!";
}
```

C'est la méthode qui sera appelée lorsque l'utilisateur clique (ou touche) le bouton. Et il mettra à jour le `TextBlock` et définira son texte sur *"Hello World!"* .

Lancer l'application

Pour exécuter l'application, vous pouvez utiliser le menu `Debug` → `Start Debugging` ou le raccourci `F5` . Par défaut, l'application exécutera l'application sur votre ordinateur local.

Lire [Démarrer avec Windows-10-universal en ligne](https://riptutorial.com/fr/windows-10-universal/topic/5304/demarrer-avec-windows-10-universal): <https://riptutorial.com/fr/windows-10-universal/topic/5304/demarrer-avec-windows-10-universal>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec Windows-10-universal	Arnaud Develay , arun , Community , Nuwan Karunarathna