

# Créer des applications Metro Design avec PowerShell et WPF

Damien VAN ROBAYEYS  
@syst\_and\_deploy



# Jeu-concours Twitter

## Un pack Surface Go à gagner\* !

- 1 Prenez une photo de la session à laquelle vous participez
- 2 Ajoutez **#experienceslive** et **#experiences18** à votre tweet
- 3 Partagez !



\*Voir toutes les conditions dans le règlement lié au jeu concours

- Le tweet cumulant le plus de Retweets remporte la Surface Go (1 lot/jour)

**#experienceslive**      **#experiences18**

# Qui suis-je ?

**Damien VAN ROBAEYS**

Ingénieur systèmes chez Metsys

<http://www.systanddeploy.com/>

@syst\_and\_deploy

Damien.vanrobaeys@gmail.com

MVP Cloud & Datacenter Management



Microsoft **experiences18**



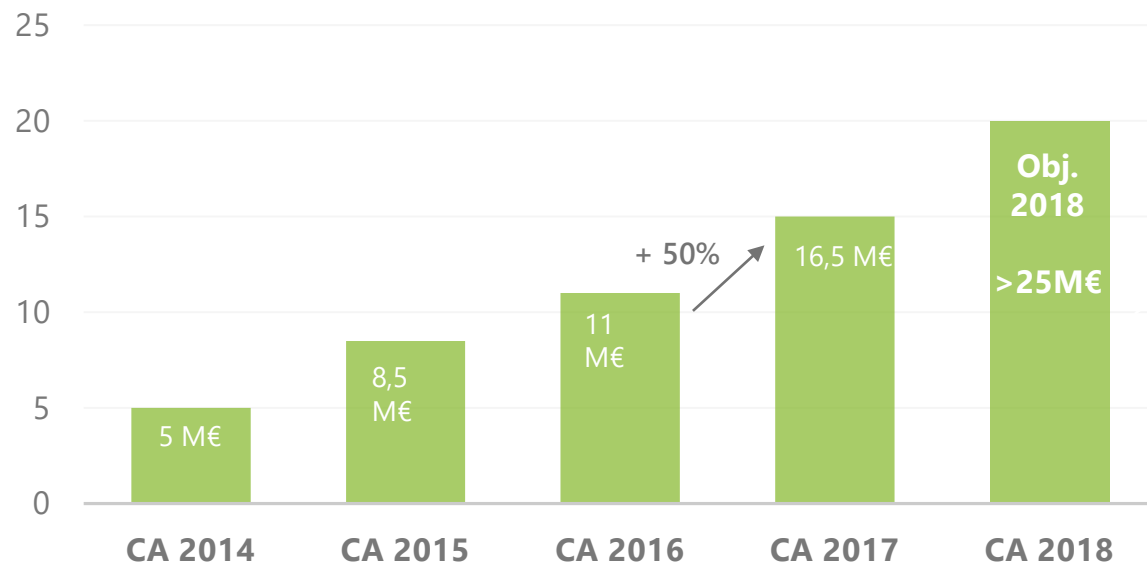
**#experiences18**

# A propos de Metsys



# Metsys

## Une société en forte croissance



Création fin 2011

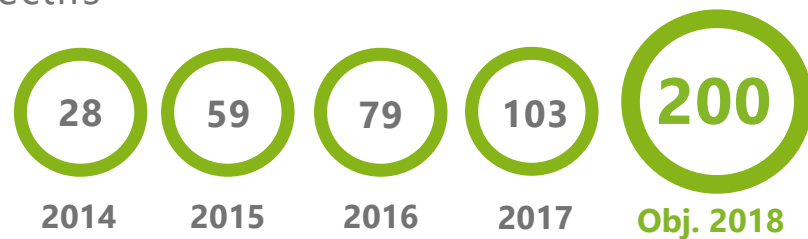
Gold  
**Microsoft  
Partner**



5 agences en France



## Effectifs



## Une relation de confiance

1. Satisfaction
2. Transparence
3. Engagement

## Socialement responsable



Microsoft **experiences18**

**#experiences18**

# Metsys – Nos offres

Cyber  
Security



Digital  
Intelligence  
& Data



Modern  
Workplace



Infrastructure  
Services



Audit

-

Intégration

-

Conseil

# WPF ? Késako ?

# Qu'est ce que le WPF ?

- ❖ WPF → Windows Presentation Foundation
- ❖ Avant XAML → Windows Forms
- ❖ Apparu en 2006 avec le **.NET 3.0**
- ❖ Permet de **créer** ses interfaces graphiques pour faire ses propres outils
- ❖ Utilise le **langage XAML**
- ❖ Peut s'utiliser dans VS, Microsoft Blend...
- ❖ Tout type d'outils: AD, Office 365, SCCM, MDT...





# Avantages de WPF par rapport à Forms ?

- ❖ Plus facile à **lire**
- ❖ Plus facile à **gérer**
- ❖ Plus facile à **maintenir**
- ❖ **Moins de lignes** de code

27 vs 11

## XAML

```
<GroupBox Header="groupbox1" HorizontalAlignment="Left" Margin="0,0,0,0" Height="118" Width="284">
  <StackPanel Orientation="Vertical">
    <StackPanel Orientation="Horizontal" Margin="0,5,0,0" >
      <Label Content="label1" Width="110"></Label>
    </StackPanel>

    <StackPanel Orientation="Horizontal" Margin="0,5,0,0" >
      <Button Content="button1" Width="110"></Button>
      <TextBox Margin="10,0,0,0" Width="200" Height="19"></TextBox>
    </StackPanel>
  </StackPanel>
</GroupBox>
```



## Forms

```
# groupbox1
#
$groupbox1.Controls.Add($textbox1)
$groupbox1.Controls.Add($button1)
$groupbox1.Controls.Add($label1)
$groupbox1.Location = '33, 22'
$groupbox1.Name = 'groupbox1'
$groupbox1.Size = '284, 118'
$groupbox1.TabIndex = 0
$groupbox1.TabStop = $False
$groupbox1.Text = 'groupbox1'
$groupbox1.add_Enter($groupbox1_Enter)
#
# textbox1
#
$textbox1.Location = '98, 71'
$textbox1.Name = 'textbox1'
$textbox1.Size = '100, 20'
$textbox1.TabIndex = 2
#
# button1
#
$button1.Location = '16, 69'
$button1.Name = 'button1'
$button1.Size = '75, 23'
$button1.TabIndex = 1
$button1.Text = 'button1'
$button1.UseVisualStyleBackColor = $True
$button1.add_Click($button1_Click)
#
# label1
#
$label1.AutoSize = $True
$label1.Location = '16, 30'
$label1.Name = 'label1'
$label1.Size = '35, 13'
$label1.TabIndex = 0
$label1.Text = 'label1'
```

# XAML ? Késako ?

# Qu'est ce que le XAML ?

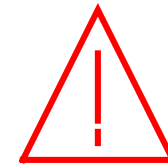
- ❖ XAML → Extensible Application **Markup Language**
- ❖ Se prononce **Zammel**
- ❖ Code permettant d'afficher votre interface
- ❖ Langage déclaratif, basé sur le **XML**, ressemblant un peu au **HTML**

## Déclaratif ?

- ❖ Comme XML, fonctionne **avec des balises**
- ❖ Une balise doit être **ouverte** et **fermée**: `<Button> </Button>` OU `<Button/>`



**XAML est case-sensitive**



# Structure d'un XAML

- ❖ Un XAML a une structure standard
- ❖ Si créé depuis VS, il faudra supprimer la ligne commençant par **x:Class**
- ❖ Définition de l'interface via balise **Window**
- ❖ Ajout du contenu de notre interface dans la balise **Grid**

```
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Height="200" Width="340">
    <Grid>
    </Grid>
</Window>
```

# La syntaxe XAML

- ❖ Chaque élément ajouté est appelé **Control**
- ❖ Un Control doit être **ouvert** et **fermé**, principe des **balises**
- ❖ Un Control peut être customisé via des **attributs**
- ❖ Name, Width, Height, Content, Text...
- ❖ Name ➔ Permet le lien entre **Interface** > **Action**
- ❖ Disposition: Orientation, HorizontalAlignment, VerticalAlignment

</>    <Button Name="MonBouton" Content="Texte" Width="80" Height="20" />

# Les Controls XAML

- ❖ **Affichage:** Label
- ❖ **Choix:** ComboBox, RadioButton, CheckBox
- ❖ **Saisie:** TextBox, PasswordBox
- ❖ **Navigation:** TabControl

Ville	Prenom	Nom
Albuquerque	Walter	White
New-Jersey	Tony	Soprano
Charming	Jax	Teller
Albuquerque	Jessy	Pinkman

Infos	Details	Other
Ville	Prenom	Nom
Albuquerque	Walter	White
New-Jersey	Tony	Soprano
Charming	Jax	Teller
Albuquerque	Jessy	Pinkman

I

I

RadioButton options

☒ Reboot ☐ Hide

CheckBox options

☒ Reboot ☐ Hide ☐ Enable

Albuquerque

Folder source

Browse

# Lier votre projet WPF à PowerShell

# Une intégration en trois étapes



## Chargement de l'Assembly

Généralement une DLL à charger



## Chargement du XAML

Affichage de la fenêtre



## Déclaration des Controls

Gestion des actions via events



## Etape 1: Chargement assembly

- ❖ Permet d'obtenir les fonctionnalités WPF: Controls, Events...
- ❖ Assembly de base → **presentationframework**
- ❖ Peut se charger en une ligne

</> `[System.Reflection.Assembly]::LoadWithPartialName('presentationframework')`

## Etape 2: Chargement XAML

- ❖ Spécifie le nom de notre fichier XAML
- ❖ Chargement via un objet XML
- ❖ Affichage via ShowDialog()

</>

```
[xml]$MonXAML = get-content -path "MonProjet.xaml"
$Reader=(New-Object System.Xml.XmlNodeReader $MonXAML)
$Interface = [Windows.Markup.XamlReader]::Load($Reader)
$Interface.ShowDialog() | Out-Null
```

## Etape 3: Déclaration des Controls

- ❖ Déclarer les Controls, boutons, zone de saisie... pour les utiliser
- ❖ On **réutilise** la valeur configurée dans l'attribut **Name** du Control dans le XAML
- ❖ Exemple: <Button Name=«MonBouton»/>

```
</>    $MonBouton = $Form.findname("MonBouton")  
        $MonTextBox = $Form.findname("MonTextBox")  
        $MonBouton = $Form.findname("MonDataGrid")
```

# Des Events pour gérer nos actions

- ❖ Events permettent **d'ajouter une action** sur un Control
- ❖ Différents events en fonction de ce que l'on souhaite faire, **à quel moment** ?
- ❖ Clic gauche, clic droit, double clic, passage du curseur de la souris...
- ❖ Utilisation ➔ `$MonBouton.Add_MonEvent({ ACTION })`

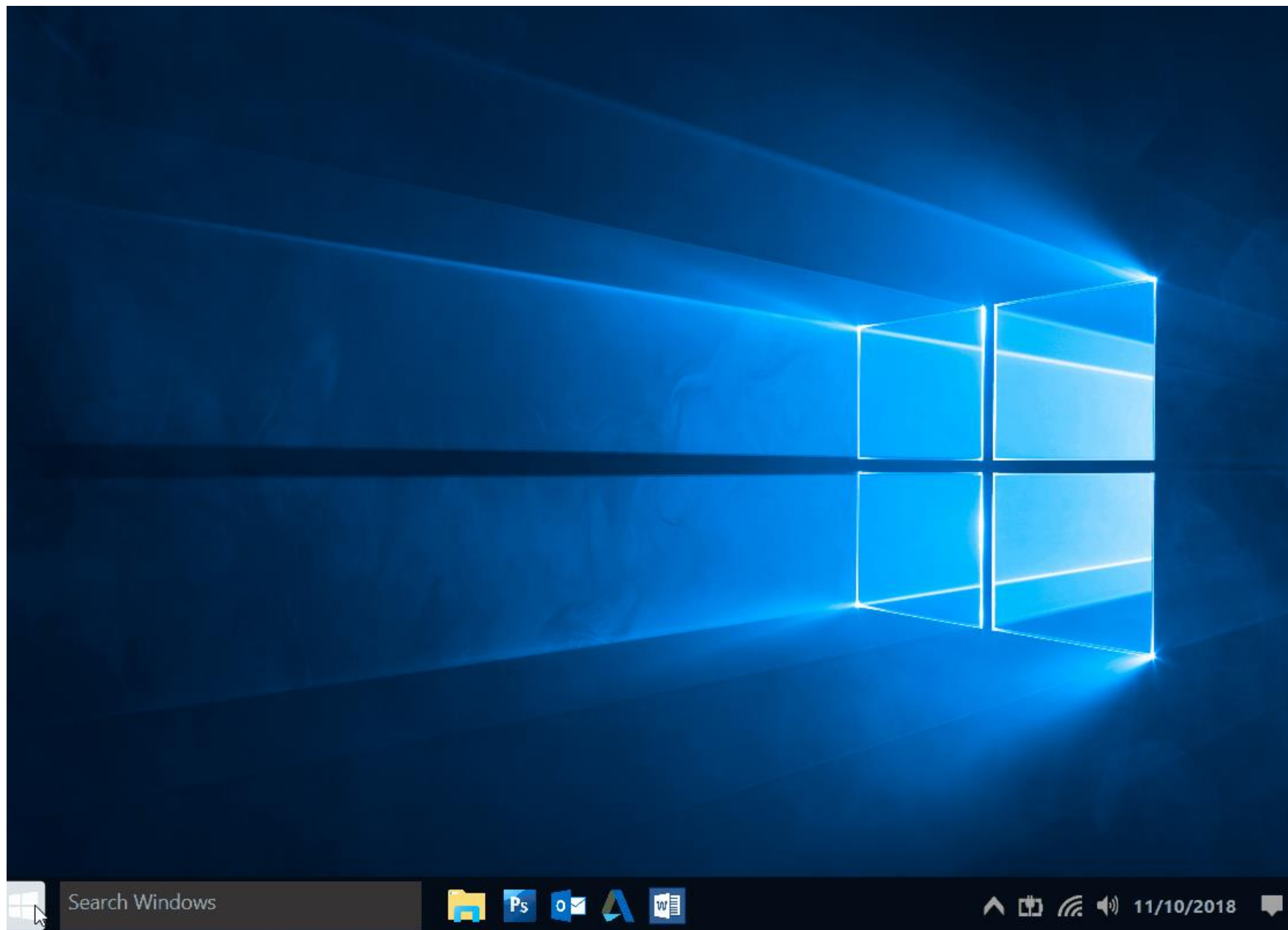
</>

```
$MonBouton.Add_Click({ #ACTION })  
$MonBouton.Add_MouseRightButtonDown({ #ACTION })  
$MonBouton.Add_MouseDoubleClick({ #ACTION })  
$MonBouton.Add_MouseEnter({ #ACTION })  
$MonInterface.Add_Closing({ #ACTION })
```

## Où les trouver ?

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls?view=netframework-4.7.2>

# Des outils Metro Design en quelques clics...



Microsoft **experiences18**

**#experiences18**

# Thème Mahapps

# Mahapps, c'est quoi ?

- ❖ Créer par Paul Jenkins en 2011
- ❖ But ? Apporter du Metro Design dans les interfaces utilisateurs
- ❖ Maintenu par Jan Karger → <https://github.com/punker76>
- ❖ Site web → <https://mahapps.com/>
- ❖ GitHub Repo → <https://github.com/MahApps/MahApps.Metro>



A quoi ca ressemble ?





# Interface de base, avec Mahapps

- ❖ Interface basique en WPF
- ❖ Avec différents Controls, les plus utilisés
- ❖ Nous allons l'updater avec le thème Mahapps

The screenshot shows a standard WPF application window titled "Mon interface". It contains several controls: a "Server selection" dropdown menu with "Albuquerque" selected; a "Folder sources" section with a "Browse" button and an empty text box; a "CheckBox options" section with three checkboxes labeled "Reboot", "Hide", and "Enable", all of which are unchecked; and a "RadioButton options" section with two radio buttons labeled "Reboot" and "Hide", both of which are unselected. At the bottom, there is a tabbed interface with three tabs: "Infos", "Details", and "Other". The "Infos" tab is active, displaying a table with four columns: "Ville", "Prenom", "Nom", and an empty column. The table contains four rows of data.

Ville	Prenom	Nom	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

The screenshot shows the same WPF application window, but with the Mahapps theme applied. The window title is "MON INTERFACE AVEC MAHAPPS". The controls are styled with a blue and white color scheme. The "Server selection" dropdown, "Folder sources" section, "CheckBox options", and "RadioButton options" sections are all highlighted with blue headers. The tabbed interface at the bottom has tabs labeled "Infos", "Details", and "Other". The "Infos" tab is active, displaying a table with columns "VILLE", "PRENOM", "NOM", and an empty column. The table contains the same four rows of data as the previous screenshot.

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

# Ajout dans notre projet – Partie XAML ?

- ❖ Ajout d'une ligne dans balise Window
- ❖ Ajout d'un bloc **resources**, permettant customisation globale de l'interface

```
<Controls:MetroWindow
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Controls="clr-namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
    Height="300" Width="340">
    <Window.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source=".\\resources\\Icons.xaml" />
                <ResourceDictionary Source=".\\resources\\custom.xaml" />
                <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />
                <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />
                <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Colors.xaml" />
                <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Cobalt.xaml"/>
                <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Window.Resources>

    <Grid>
    </Grid>
</Controls:MetroWindow>
```

## Ajout dans notre projet – Partie PS1 ?

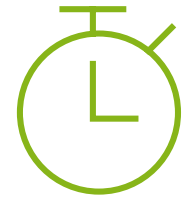
- ❖ Assembly à ajouter comme pour presentationframework, MahApps.Metro.dll
- ❖ DLL à récupérer sur Github ou dans son projet VS
- ❖ En utilisant la ligne suivante :

```
</> [System.Reflection.Assembly]::LoadFrom('assembly\MahApps.Metro.dll') | out-null
```



# Nouvelle couleur en une minute

- ❖ Possibilité de changer rapidement la couleur des Controls **en un clic**
- ❖ Modifier la valeur de la ligne ci-dessous ➔ *Cobalt* > *Red* > *Pink*



</>

```
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Cobalt.xaml"/>  
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Red.xaml"/>  
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Pink.xaml"/>
```

```
<Window.Resources>  
  <ResourceDictionary>  
    <ResourceDictionary.MergedDictionaries>  
      <!-- Create Metro Style -->  
      <ResourceDictionary Source="..\resources\Icons.xaml" />  
      <ResourceDictionary Source="..\resources\custom.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Colors.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Cobalt.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml" />  
    </ResourceDictionary.MergedDictionaries>  
  </ResourceDictionary>  
</Window.Resources>
```

# Des nouvelles couleurs rapidement

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

# Nouveau thème en une minute

- ❖ Possibilité de changer rapidement le thème global de l'interface
- ❖ Modifier la valeur de la ligne ci-dessous ➔ **BaseLight** OU **Basedark**



</>

```
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml"/>  
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseDark.xaml"/>
```

```
<Window.Resources>  
  <ResourceDictionary>  
    <ResourceDictionary.MergedDictionaries>  
      <!-- Create Metro Style -->  
      <ResourceDictionary Source=".\\resources\\Icons.xaml" />  
      <ResourceDictionary Source=".\\resources\\custom.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Colors.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Cobalt.xaml" />  
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml" />  
    </ResourceDictionary.MergedDictionaries>  
  </ResourceDictionary>  
</Window.Resources>
```

# Un nouveau thème rapidement

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

MON INTERFACE AVEC MAHAPPS

Mon interface

SERVER SELECTION

Albuquerque

FOLDER SOURCES

BROWSE

CHECKBOX OPTIONS

☐ Reboot ☐ Hide ☐ Enable

RADIOBUTTON OPTIONS

☐ Reboot ☐ Hide

Infos Details Other

VILLE	PRENOM	NOM	
Albuquerque	Walter	White	
New-Jersey	Tony	Soprano	
Charming	Jax	Teller	
Albuquerque	Jessy	Pinkman	

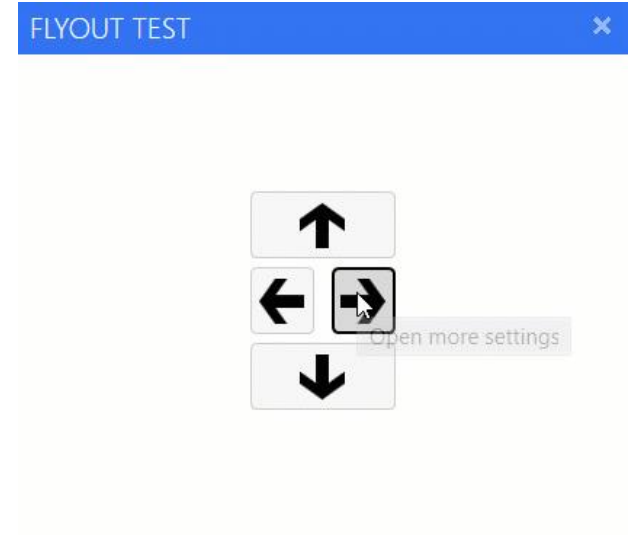
# Des effets en mode Windows 10



# Le Flyout

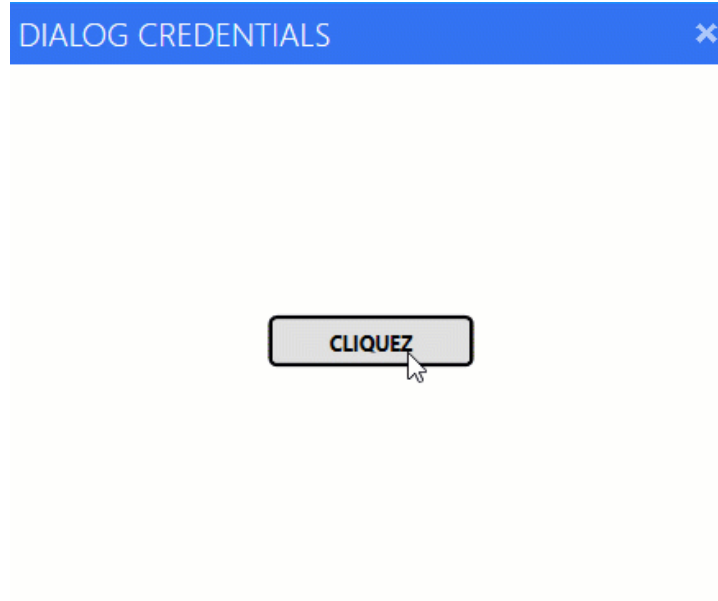
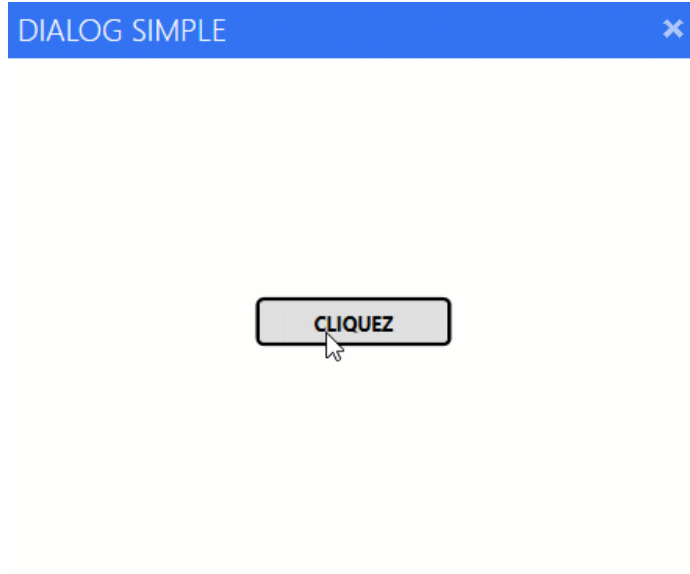
- ❖ Un Design Windows 10, look du clic sur Menu démarrer
- ❖ S'active via un bouton par exemple ➔ `$Flyout.IsOpen = $True / $False`
- ❖ Permet par exemple d'afficher des propriétés

```
<Controls:MetroWindow.Flyouts>
  <Controls:FlyoutsControl>
    <Controls:Flyout x:Name="FlyOutContent" Header="Options" Position="Right" Width="200">
      <!-- Vos controls ici -->
    </Controls:Flyout>
  </Controls:FlyoutsControl>
</Controls:MetroWindow.Flyouts>
```



# Les Dialogs

- ❖ Remplace les vieux message box
- ❖ Via [MahApps.Metro.Controls.Dialogs.DialogManager]
- ❖ Plusieurs types : Dialog simple, authentication, validation



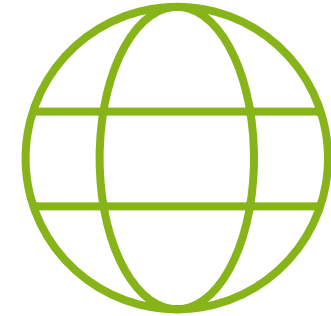
# Quelques démos rapides

# Les démos

- ❖ Les events en démos
- ❖ Update d'une interface simple en Metro Design
- ❖ Barre de progression au look Windows 10
- ❖ Fake Windows 10
- ❖ Quick AD Support: analyse utilisateur et ordinateur Active Directory

# Liens intéressants

- ❖ <http://www.powershell-wpf.com/>
- ❖ <https://www.facebook.com/groups/PowershellGUI/>
- ❖ <https://www.dev4sys.com/>
- ❖ <https://jm2k69.github.io/>
- ❖ <https://smsagent.wordpress.com/>
- ❖ <http://www.systanddeploy.com/>
- ❖ <https://youtu.be/mU8M3955reg>
- ❖ <https://www.youtube.com/watch?v=uJcvwgbTeks&list=PLsg-xXEEemCJozYQCiBxO5ydYI1KIV2pI4>
- ❖ <https://www.wpftutorial.net/>
- ❖ <http://www.powershell-wpf.com/p/starts-wpf-from-scratch.html>
- ❖ <https://foxdeploy.com/series/learning-gui-toolmaking-series/>



# Où trouver les slides et démos ?

Le diaporama et les démos seront disponibles sur mon Github et celui de Metsys

- ❖ [https://github.com/damienvanrobaeys/MS\\_Experiences\\_2018](https://github.com/damienvanrobaeys/MS_Experiences_2018)
- ❖ <https://github.com/metsys-fr/Conference>
- ❖ <http://www.systanddeploy.com>



Notez maintenant cette session via le QR code

Ou en scannant ce QR code



Cliquez sur le bouton « Evaluer »

# Merci !

Microsoft  
experiences18



# Microsoft experiences18