

# メトロスタイルアプリ開発 最初の一歩

青柳 臣一



## MetroStyleDeveloper #02

2012年4月19日 (土)

# 自己紹介

Twitter: @ShinichiAoyagi

ブログ: <http://shinichiaoyagi.blogspot.com/>

## 仕事

ソフトウェア開発  
WPF、Silverlight、  
C#、VB、

## 職場

淀屋橋

## 自宅

吹田

# Windows 8

- 正式名

- Windows 8

- Windows 8 Pro

- Windows RT

- 発売時期は不明

- Windows 8 Consumer Preview

- リリース中

# Windows 8 のエディション

Feature name	Windows 8	Windows 8 Pro	Windows RT
Windows 7 Starter、Home Basic、Home Premiumからのアップグレード	○	○	—
Windows 7 Professional、Ultimateからのアップグレード	—	○	—
Start画面、セマンティックズーム、ライブタイトル	○	○	○
Windows Store	○	○	○
アプリケーション (Mail、Calendar、People、Messaging、Photos、SkyDrive、Reader、Music、Video)	○	○	○
Microsoft Office (Word、Excel、PowerPoint、OneNote)	—	—	○
Internet Explorer 10	○	○	○
デバイス暗号化	—	—	○
Connectedスタンバイ	○	○	○
Microsoftアカウント	○	○	○
デスクトップ	○	○	○
x86/64環境とデスクトップ向けソフトウェアのインストール	○	○	—
拡張Windowsエクスプローラー	○	○	○
Windows Defender	○	○	○
スマートスクリーン	○	○	○
Windows Update	○	○	○
拡張タスクマネージャー	○	○	○

利用中の言語切替 (Language Pack)	○	○	○
マルチモニターサポート改善	○	○	○
Storage Spaces	○	○	—
Windows Media Player	○	○	—
Exchange ActiveSync	○	○	○
ファイル履歴	○	○	○
ISO/VHD形式ファイルのマウント	○	○	○
モバイル・ブロードバンド対応	○	○	○
ピクチャー・パスワード	○	○	○
Play To	○	○	○
リモートデスクトップ (クライアント)	○	○	○
PCの初期化	○	○	○
Snap	○	○	○
タッチキーボード	○	○	○
Trusted boot	○	○	○
VPNクライアント	○	○	○
BitLocker/BitLocker To Go	—	○	—
VHDブート	—	○	—
クライアント向けHyper-V	—	○	—
ドメイン参加	—	○	—
Encrypting File System (EFS)	—	○	—
グループポリシー	—	○	—
リモートデスクトップ (ホスト)	—	○	—

※ITPro より

# メトロスタイル Apps とデスクトップ Apps

Metro Style Apps			Desktop Apps		
XAML		HTML CSS	HTML JavaScript	C C++	C# VB
C C++	C# VB	JavaScript			
WinRT			IE	Win32	.NET Silverlight

# メトロスタイル Apps

Windows  
Store

WinRT

フル  
スクリーン

サスペンド

タイル

Share

# Windows Store

要登録

有料

審査

メトロスタイル  
Apps

デスクトップ  
Apps

??????

# WinRT

- 新しいランタイム
  - .NET Framework、Silverlight、Windows Phone のいずれとも異なる
- 制限された API (セキュア)
- 非同期を前提とした API



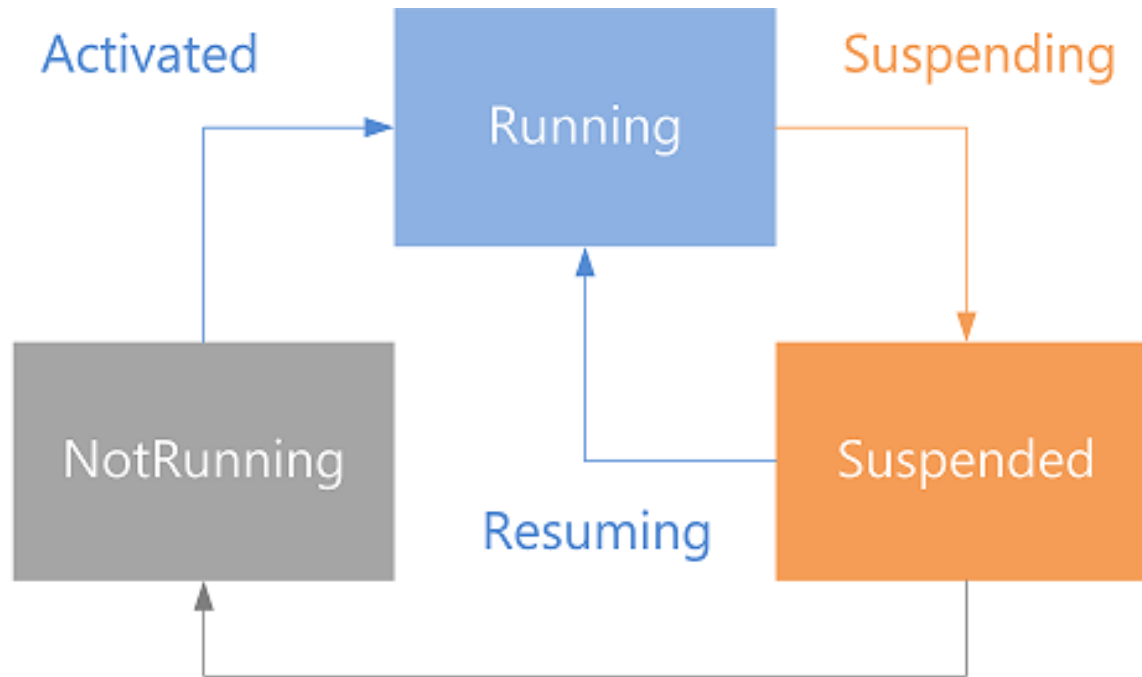
# フルスクリーン

- 最低 1024px×768px
- スナップ 320px固定



# サスペンド

- いつの間にか終了
- 起動時に復帰

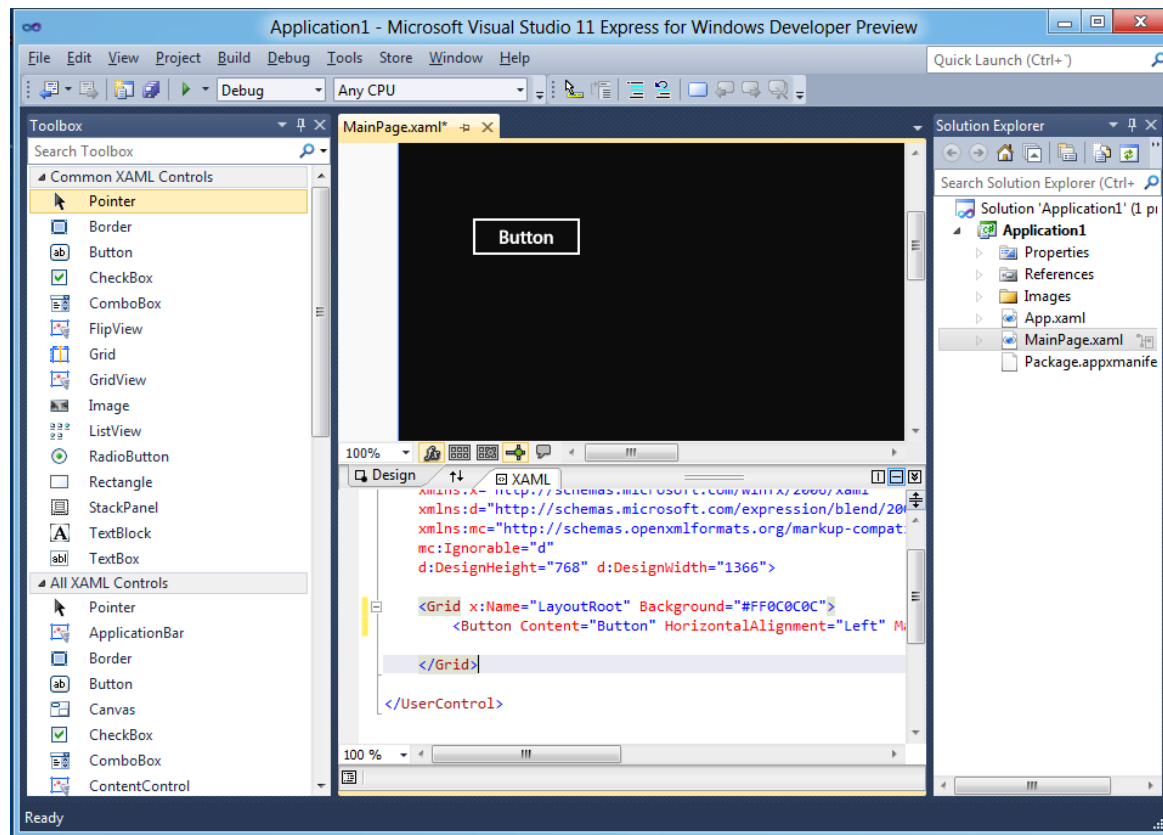


# インターネットエクスプローラー

- IE10 相当
- メトロスタイル IE とデスクトップ IE
- メトロスタイル IE から Windows Store やアプリへシームレスに連携
  - 例: <http://www.buildwindows.com/>

# デモ

- Visual Studio 11 Beta



# XAML (1)

```
<Grid Background="{StaticResource ApplicationPageBackgroundBrush}">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>
  <Grid Grid.Row="0">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto"/>
      <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>
    <Button Grid.Column="0" x:Name="OpenButton" FontSize="48"
      Content="開< " Width="300" Margin="16,16,16,16"
      Click="OpenButton_Click"/>
    <Border Grid.Column="1">
      <TextBlock x:Name="FileNameText" FontSize="48"
        Margin="16,16,16,16" VerticalAlignment="Center"/>
    </Border>
  </Grid>
  <Image x:Name="PictureImage" Grid.Row="1" Margin="16,16,16,16"/>
</Grid>
```

## XAML (2)

- だいたい WPF、Silverlight、WP7 と同じ感じ
- けど、いろいろ違うところもあり
  - そもそも属する名前空間が違ったり  
(Windows.UI.Xaml.Controls)
  - TextBlock には Background が無かったり
  - Border や TextBlock は sealed だったり

# 画像を選択 (1)

```
private void OpenButton_Click(object sender, RoutedEventArgs e)
{
    var openPicker = new FileOpenPicker();
    openPicker.ViewMode = PickerViewMode.Thumbnail;
    openPicker.SuggestedStartLocation = PickerLocationId.PicturesLibrary;
    openPicker.FileTypeFilter.Add(".jpg");
    var asyncOpe = openPicker.PickSingleFileAsync();
    asyncOpe.Completed = onCompleted;
}

private void onCompleted(IAsyncOperation<Windows.Storage.StorageFile> asyncInfo,
AsyncStatus asyncStatus)
{
    if (asyncStatus == AsyncStatus.Completed)
    {
        var file = asyncInfo.GetResults();
        var path = file.Path;
        this.Dispatcher.Invoke(Windows.UI.Core.CoreDispatcherPriority.Normal,
            (s, e) =>
            {
                this.FileNameText.Text = path;
            },
            this, null);
    }
}
```

# 非同期

- 50ms 以上かかりそうなものはみんな非同期
  - 例: ファイルを作成  
`StorageFolder.CreateFileAsync()`
  - 例: ファイルを列挙  
`CreateFileQuery()` して  
`GetFilesAsync()`
- 同期版の API はそもそも存在しない



# 画像を選択 (2)

- `async/await`

```
private async void OpenButton_Click(object sender,
                                     RoutedEventArgs e)
{
    var openPicker = new FileOpenPicker();
    openPicker.ViewMode = PickerViewMode.Thumbnail;
    openPicker.SuggestedStartLocation =
        PickerLocationId.PicturesLibrary;
    openPicker.FileTypeFilter.Add(".jpg");
    var file = await openPicker.PickSingleFileAsync();
    this.FileNameText.Text = file.Path;
}
```

ありがとうございました。