

# Sviluppare Universal App per Windows e Windows Phone

Daniele Bochicchio

*Network Manager @ ASPItalia.com | Chief Digital Officer @icubedsrl*

*Microsoft Regional Director | Microsoft MVP per ASP.NET*

[daniele@aspitalia.com](mailto:daniele@aspitalia.com) - Twitter @dbochicchio

<http://blogs.aspitalia.com/daniele/>



//22 Ottobre 2014

Universal App & Web Live

#aspilive

# Agenda

- Windows + Windows Phone = Universal App
- Shared Project
- XAML vs WinJS
- Novità di Visual Studio
- Migrazione di app esistenti

# Universal app = tablet/desktop + smartphone

- Alla base di tutto c'è WinRT
    - Convergenza di app model, tool e store
  - Si pubblica una sola app sullo store
    - Condividono il canale delle push, gli acquisti, etc
  - Si sviluppano usando WinRT
    - Un unico framework per semplificarci la vita
- 
- WinRT = Windows RunTime

# Cos'è un'Universal App?

- Un'app WinRT è potenzialmente già un'Universal App
- Un'Universal App è un'app WinRT
- In futuro ci sarà il supporto anche per Xbox One
- Windows 10 consentirà l'uso in finestra
  - Sono esattamente come app desktop, ma girano in un ambiente sandboxed

# Supporto per le Universal App

- Tablet + desktop:
  - Windows 8.1 (Update 1), Windows 10 Preview
- Smartphone
  - Windows Phone 8.1
- Xbox One
  - Ogni Xbox One è un dev kit, ma non ci sono date di rilascio
- Per sviluppare ci serve Visual Studio 2013 Update  $\geq 2$ 
  - <http://aspit.co/auk>

# Universal App = magia

- Un'Universal App è *magia* in fase di compilazione
- Il template aggiunge uno Shared Project in cui inserire il codice condiviso
- I file in questo progetto vengono automaticamente copiati nei progetti da cui è referenziato
- Consente di tenere condivisi qualsiasi tipo di risorsa
  - XAML/WinJS
  - Codice
- Supporta C#, C++ e JavaScript

# One app, multiple screens

- Bisogna iniziare a pensare ai layout in maniera fluida
- Windows ha infinite risoluzioni, Windows Phone ha 3 form factor (16:9, 16:10, 15:9)
- XAML: layout con dimensioni in percentuale
  - In alcuni casi può aiutarci il controllo ViewBox
- Ricordate che da Windows 10 le finestre sono ridimensionabili come quelle desktop
- WinJS: i CSS e le media queries consentono di adattare il layout alle differenti risoluzioni

# Alcuni trucchi per le view

- Usate il più possibile gli stili
  - Consentono di diversificare la resa grafica in base al tipo di piattaforma
- Se parte del layout cambia, incapsulatelo in uno User Control
  - Potrete così variare il risultato e riutilizzarlo in parti della vostra app
- Tenete semplice lo XAML/markup
- Capite i diversi use case dell'UX di tablet e telefono
  - E non abbiate paura di diversificare il comportamento

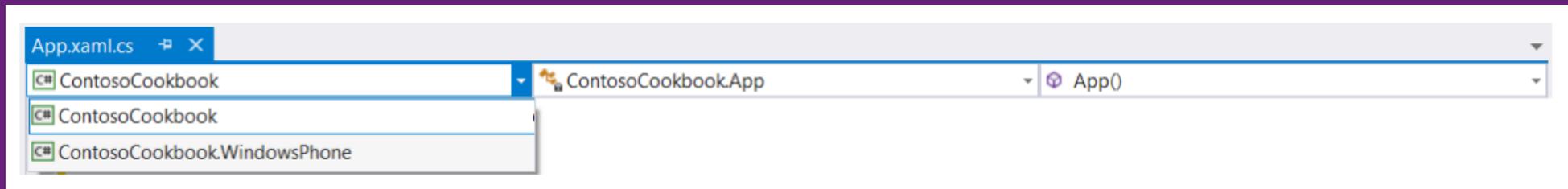
# Demo

Universal App per tutti

#aspilive

# Visual Studio: navbar

- Contesto dalla navbar



- Consente di cambiare al volo il contesto del codice e del designer

# Visual Studio: Intellisense

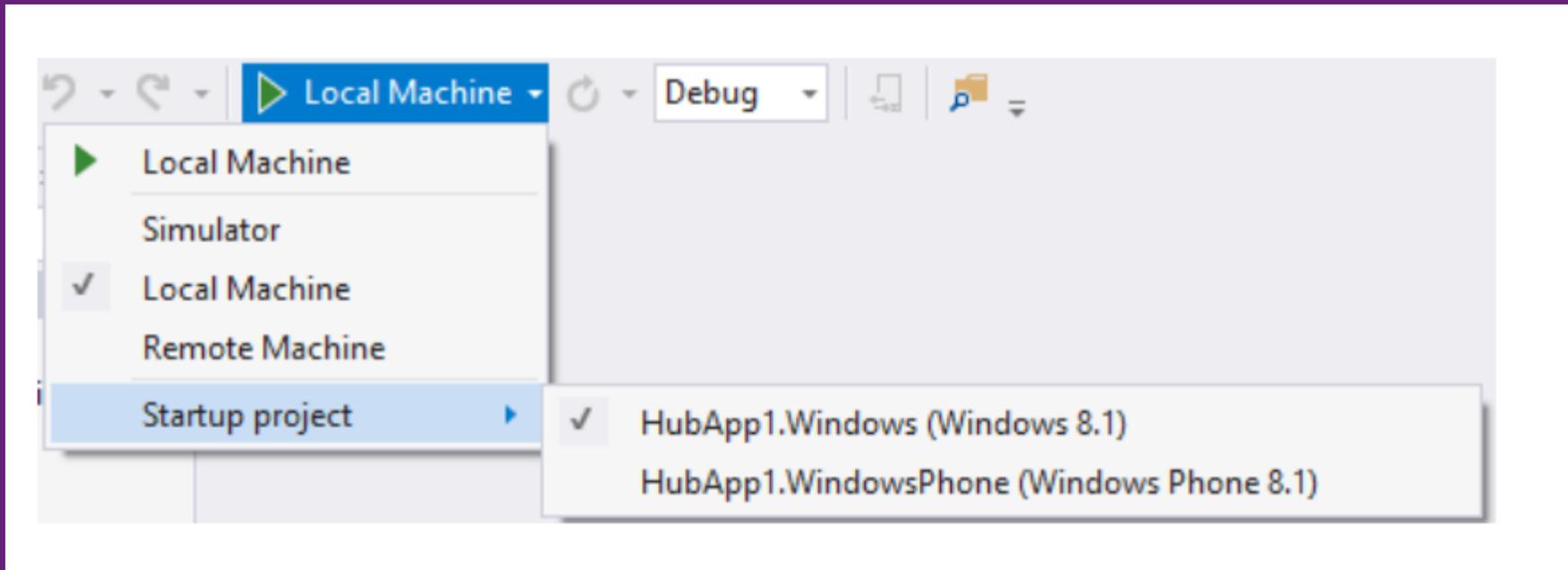
- L'Intellisense avvisa di mancanze specifiche e/o differenze tra le piattaforme

```
Window.Current.CoreWindow.Dispatcher.AcceleratorKeyActivated +=
    CoreDispatcher_AcceleratorKeyActivated;
Window.Current.CoreWindow.PointerPressed +=
    this.CoreWindow_PointerPressed;
}
#if WINDOWS_PHONE_APP
    Windows.Phone.UI.Input.HardwareButtons.
#endif
};
```

⚡ BackPressed	⚠	EventHandler<Windows.Phone.UI.Input.BackPressedEventArgs> HardwareButtons.BackPressed
⚡ CameraHalfPressed	⚠	Occurs when the user presses the hardware Back button.
⚡ CameraPressed	⚠	
⚡ CameraReleased	⚠	Windows Phone 8.1 - Available Windows 8.1 - Not available
⊕ Equals		
⊕ ReferenceEquals		You can use the navigation bar to switch context

# Visual Studio: startup project

- Facilmente modificabile dalla barra Run, per semplificare il debug



- Visualizzato in caso di presenza di più di un progetto

# Quanto posso migrare?

- Se avete un'app per Windows Store, siete sulla buona strada
  - Tasto destro dentro il progetto in Visual Studio, poi *Add Windows Phone 8.1 app*
- Ma probabilmente partire da un'app Silverlight...
- E quindi dovete rifare l'app
  - Avrete a *costo-quasi-zero* anche quella per tablet
- Lo XAML si migra facilmente
  - Al netto di alcuni differenti controlli
    - Ad esempio, Panorama = Hub, Pivot = FlipView

# Quanto posso migrare?

- Per il codice: dipende da cosa c'è all'interno
  - Se è codice che accede all'hardware o è specifico, occorrerà probabilmente rifarlo
- Se avete usato M-V-VM per la vostra app siete a buon punto
  - Avete già pensato la vostra app in maniera agnostica rispetto alla View
- Ci sono delle zone grigie
  - Niente WCF, CameraCaptureTask, LockScreen, RingTone, etc

# Demo

Un po' di esempi

#aspilive

# Live Tile

- Le Live Tile sono il sistema più semplice per convogliare
- Le Universal App consentono di avere un solo formato per le Tile
- Windows Phone non supporta le large (310x310)
- Stesso sistema di push lato server
  - Meno investimenti in codice
- Questo concetto si estende anche alle tile associate ad un sito in IE per Windows Phone 8.1

# Bonus Demo

Live Tile

#aspilive

# Recap

- Universal App è per eliminare l'effort di avere 2 app differenti per tablet e smartphone
  - E con Windows 10, si avrà anche il desktop
- Le Universal App offrono una serie di API per l'accesso alle funzionalità del sistema operativo attraverso WinRT
- Serve Visual Studio 2013 Update  $\geq 2$  per sviluppare
- Ci sono tante opportunità per condividere codice: non perdetevi le prossime sessioni!
- Il materiale è disponibile subito su <http://aspit.co/web14-live>

# PROMOTION



- In uscita da Hoepli ad ottobre
- Prezzo di copertina: 39,90 EUR
- In prenotazione a 33,92 EUR
- <http://aspit.co/axk>