

Windows 10 – Highlights für Entwickler

Malte Lantin

Technical Evangelist
Microsoft Deutschland GmbH

@MalteLantin

malte.lantin@microsoft.com

http://blogs.msdn.com/b/malte_lantin/

Agenda

- Tools
- Convergence
- Universal Windows Apps
- Adaptive Code
- Adaptive UI
- App to App & Integration
- Bridges
- Publishing & Store

A word on „Apps“

Apps vs. Desktop Applications

Where's the difference?

Classic App	Modern App
Installed from anywhere	Installed from the store or via sideloading
Does anything during installation/update/deinstallation.	Defined installation/update/deinstallation routine.
Can access the whole system during runtime.	Limited access to limited number of APIs. Explicitly declares capabilities.
Can run as admin.	Limited permissions. No admin mode.
No-Suspend-Lifecycle	Running/Suspended/Terminated
Full communication with other processes.	Sandbox.

Visual Studio 2015 with Universal Windows App Development Tools



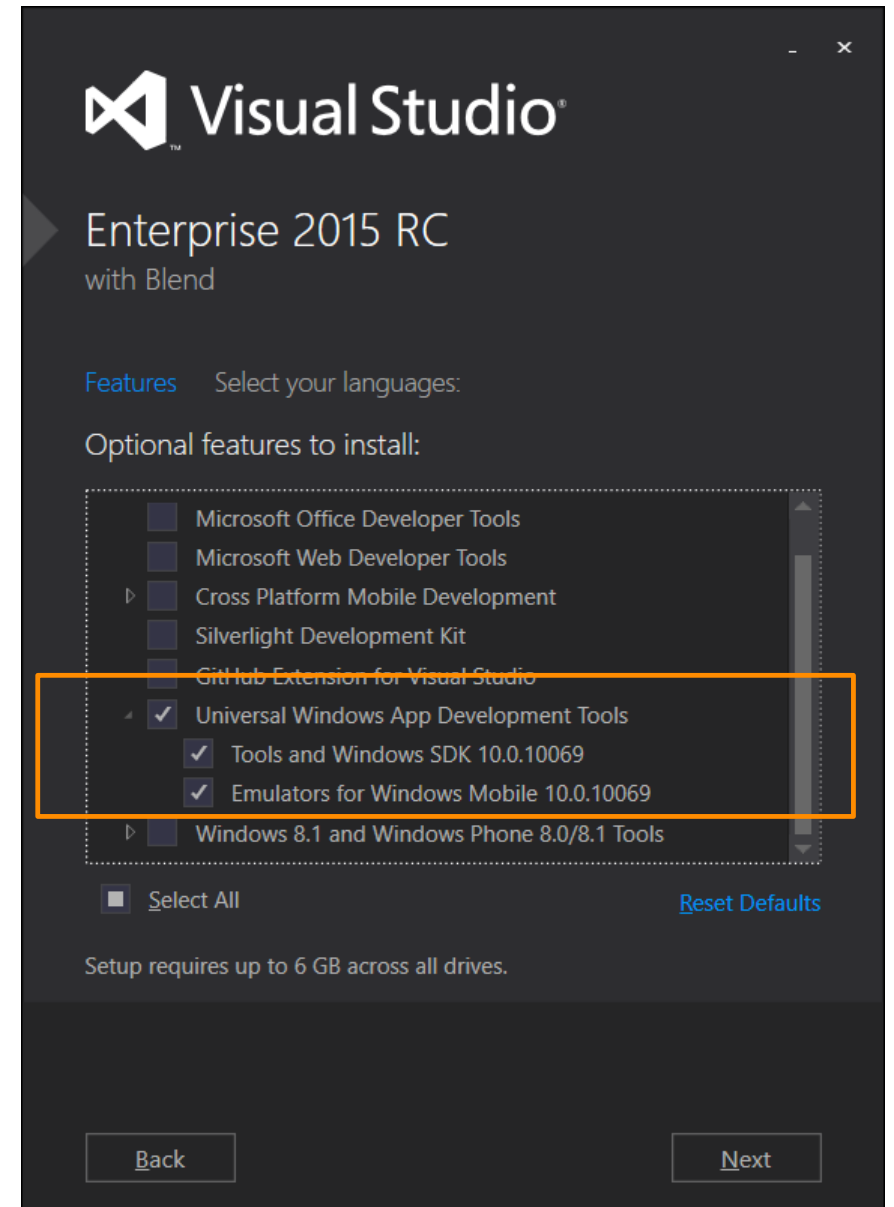
Install Windows 10

Install VS 2015 w/ Universal
Windows SDK & Tools

Build your first
Universal Windows
App

Acquiring Windows 10 Tools/SDKs/Emulators

Select "*Universal Windows App Development Tools*"
using Custom setup option



Supported OS for Universal Windows development

Windows 10

- Best developer experience for building Universal Windows apps
- Deploy/Debug/Profile
 - Simulator
 - Local Machine
 - Mobile Emulators
 - Remote Machine
 - Device
- XAML Designer/Intellisense

Windows 8.1

Windows Server 2012 R2

All other features work at parity with Windows 10, except:

- Deploy/Debug/Profile
 - Mobile Emulators
 - Remote Machine
 - Device
- XAML Intellisense
- No WACK (Server R2)

*Windows 7

All other features work at parity with Windows 10, except:

- Deploy/Debug/Profile
 - Remote Machine
 - Device
- XAML Intellisense

*will be supported at RTM

Reach customers on any device

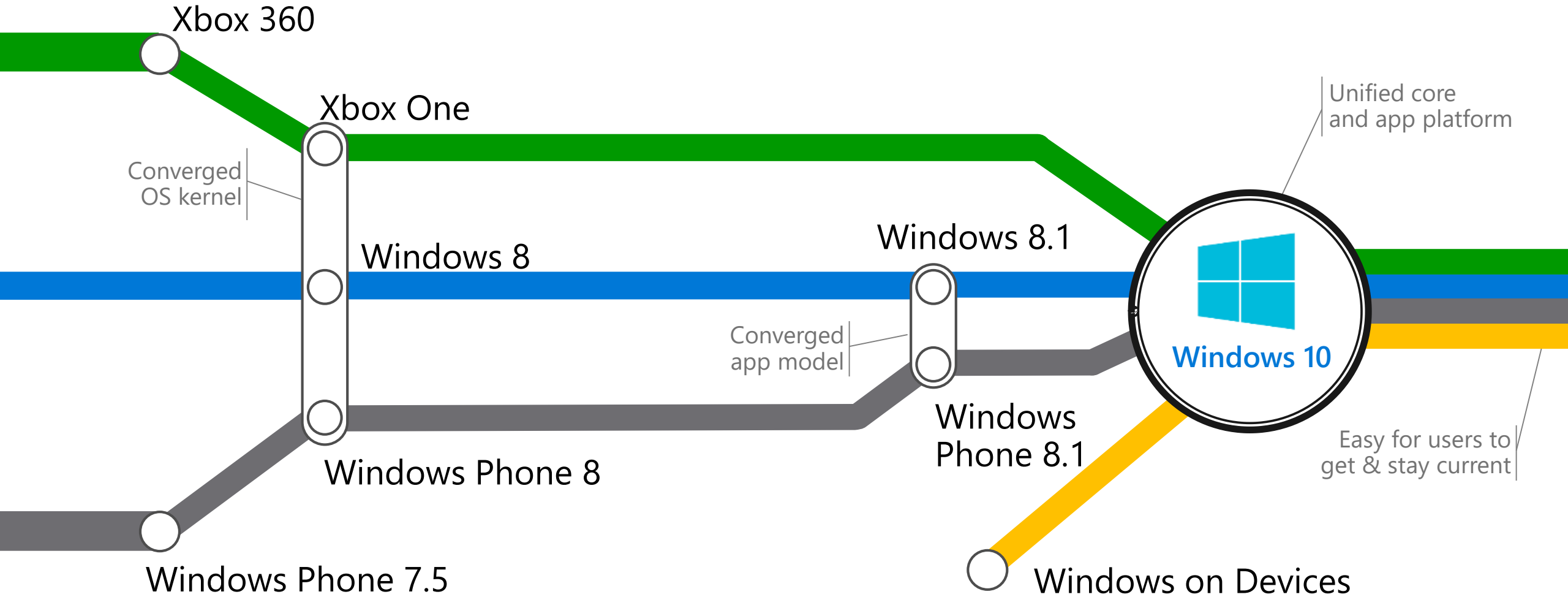


Windows 10



Free upgrade for the first year

The convergence journey



One app platform



Universal Windows platform

Adaptive User Interface

Natural User Inputs

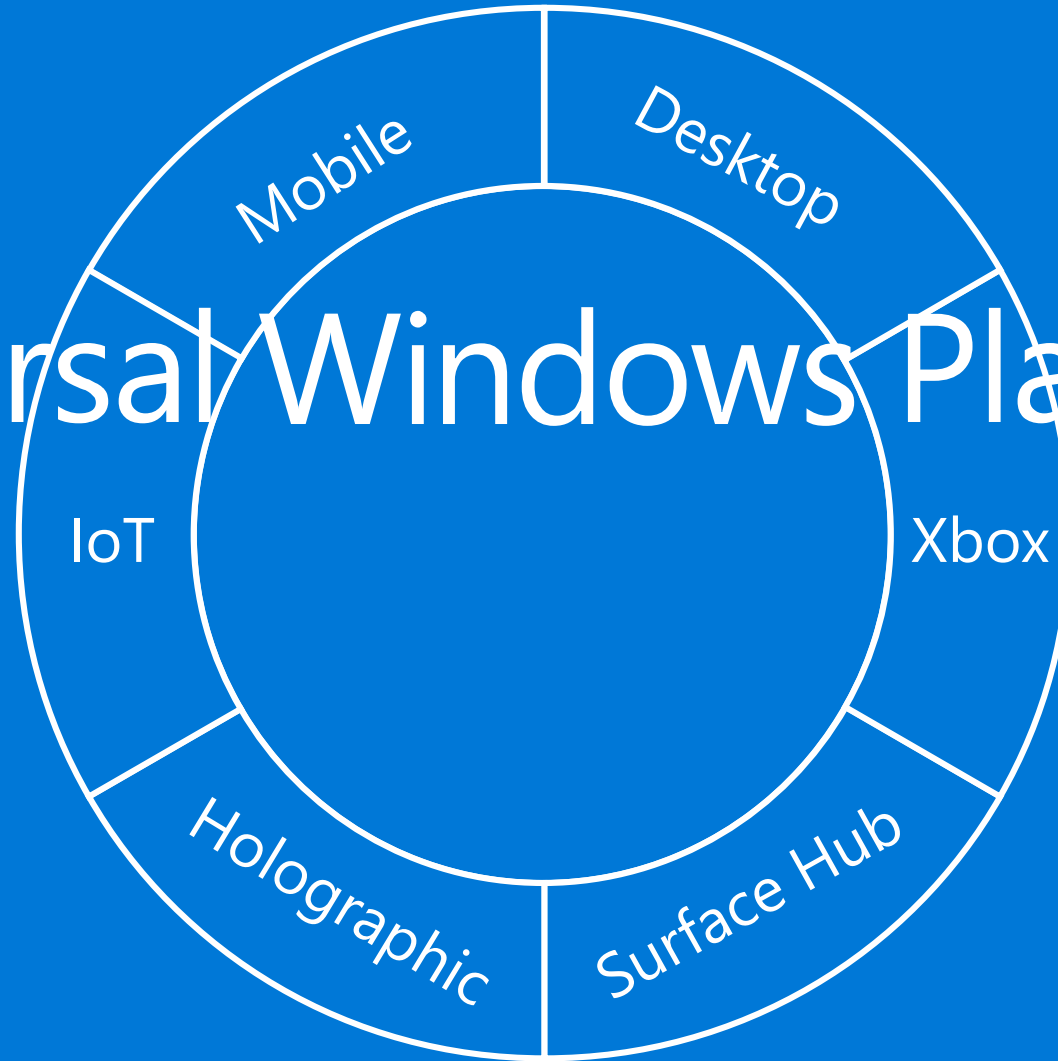
SDK + Services

One Store + One Dev Center

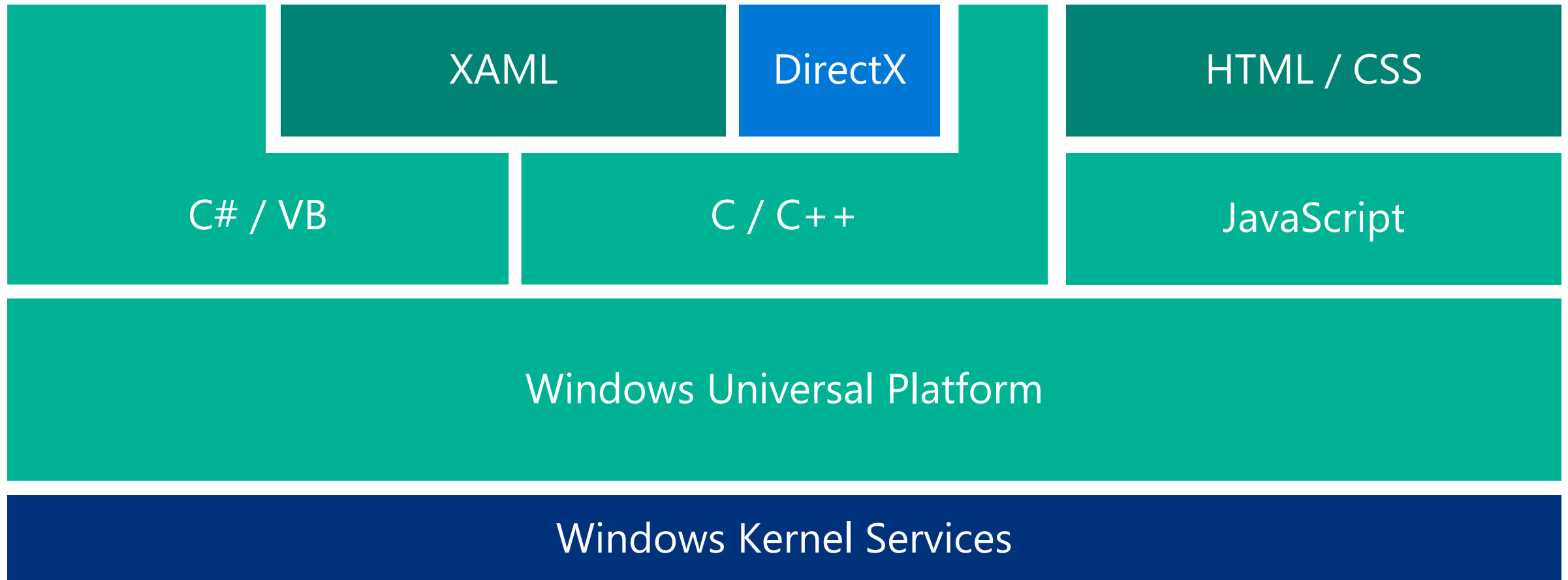
Full Support for Existing Code

One Package

Universal Windows Platform



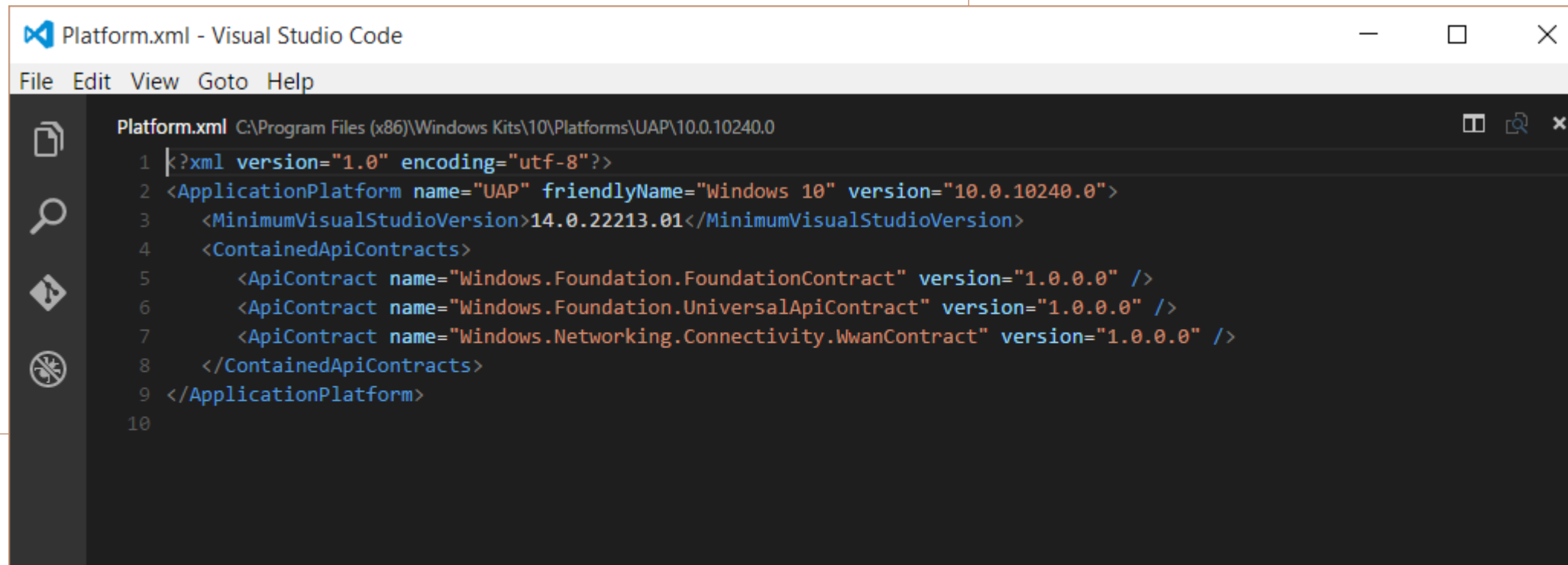
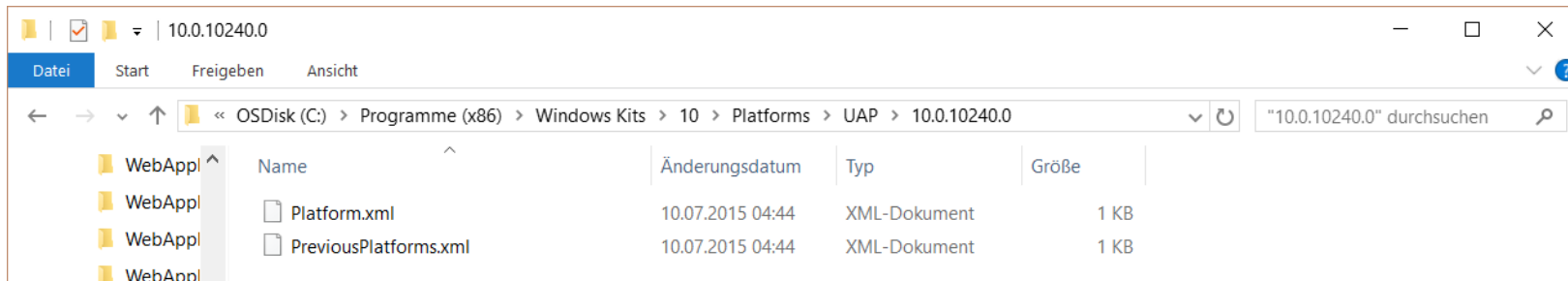
One API. One Package. Same Platform.



Target a version of UAP
not the Operating System

UWP = Universal Windows Platform

- A collection of contracts & versions



Platform Versioning

```
<Dependencies>  
  <TargetDeviceFamily  
Name="Windows.Universal"  
MinVersion="10.0.0.0"  
MaxVersionTested="10.0.0.0" />  
</Dependencies>
```


Extension SDKs

- Extends the UAP
- Targets specific platforms
- Updates at a separate cadence
- Enabled on every device

The Extension SDK/Manifest

The image shows a Windows Explorer window and a Visual Studio Code editor window. The Explorer window on the left shows the directory structure for the Extension SDK/Manifest, and the Visual Studio Code window on the right shows the contents of the SDKManifest.xml file.

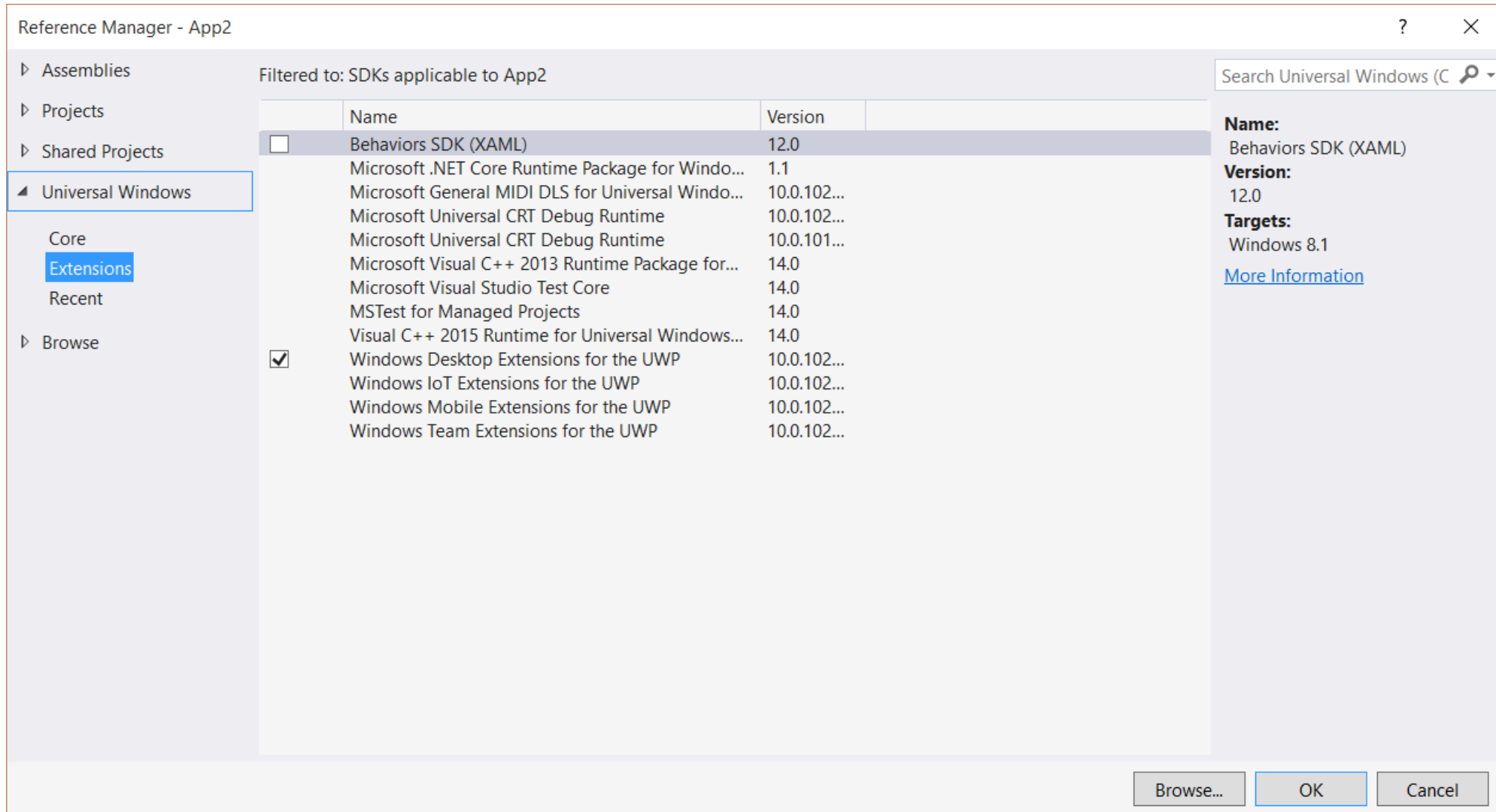
Windows Explorer (Left): Shows the directory structure for the Extension SDK/Manifest. The path is: `Programme (x86) > Windows Kits > 10 > Extension SDKs > WindowsMobile > 10.0.10240.0`. The files listed are:

Name	Änderungsdatum	Typ	Größe
bin	01.08.2015 03:54	Dateiordner	
DesignTime	01.08.2015 03:54	Dateiordner	
Include	01.08.2015 03:54	Dateiordner	
Lib	01.08.2015 03:54	Dateiordner	
SDKManifest.xml	10.07.2015 07:52	XML-Dokument	

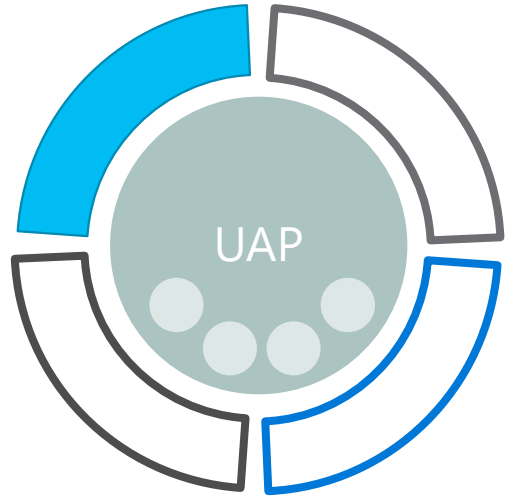
Visual Studio Code (Right): Shows the contents of the `SDKManifest.xml` file. The file is located at `C:\Program Files (x86)\Windows Kits\10\Extension SDKs\WindowsMobile\10.0.10240.0`. The XML content is:

```
<?xml version="1.0" encoding="utf-8"?>
<FileList TargetPlatform="UAP" TargetPlatformMinVersion="10.0.0.1" TargetFramework=".NETCore, version=v4.5.3;" SupportPrefer32Bit="True">
  <ContainedApiContracts>
    <ApiContract name="Windows.ApplicationModel.Activation.WebUI" version="1.0.0.0"/>
    <ApiContract name="Windows.ApplicationModel.Calls.CallsPhoneContract" version="1.0.0.0"/>
    <ApiContract name="Windows.ApplicationModel.Calls.CallsVoipContract" version="1.0.0.0"/>
    <ApiContract name="Windows.ApplicationModel.SocialInfo.SocialInfoContract" version="1.0.0.0"/>
    <ApiContract name="Windows.ApplicationModel.CommunicationBlocking.CommunicationBlockingContract" version="1.0.0.0"/>
    <ApiContract name="Windows.ApplicationModel.Wallet.WalletContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Devices.SmartCards.SmartCardBackgroundContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Devices.SmartCards.SmartCardEmulationContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Embedded.DeviceLockdown.DeviceLockdownContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Gaming.Preview.GamesEnumerationContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Networking.NetworkOperators.NetworkOperatorsFdnContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Networking.Sockets.ControlChannelTriggerContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Phone.PhoneContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Phone.PhoneContract.DualSimFileContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Services.Maps.GuidanceContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Services.Maps.LocalSearchContract" version="1.0.0.0"/>
    <ApiContract name="Windows.Security.EnterpriseData.EnterpriseDataContract" version="1.0.0.0"/>
    <ApiContract name="Windows.System.Profile.ProfileHardwareTokenContract" version="1.0.0.0"/>
  </ContainedApiContracts>
</FileList>
```

Adding extensions

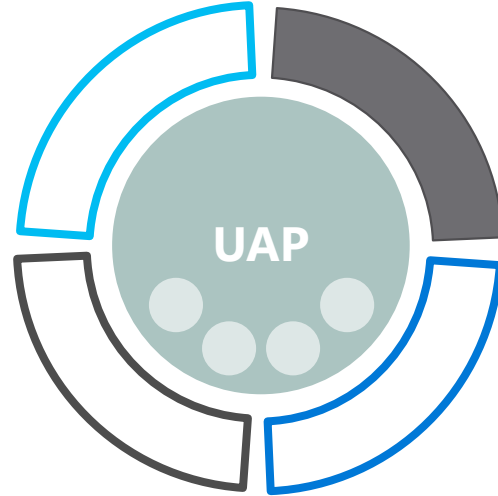


Extension SDKs



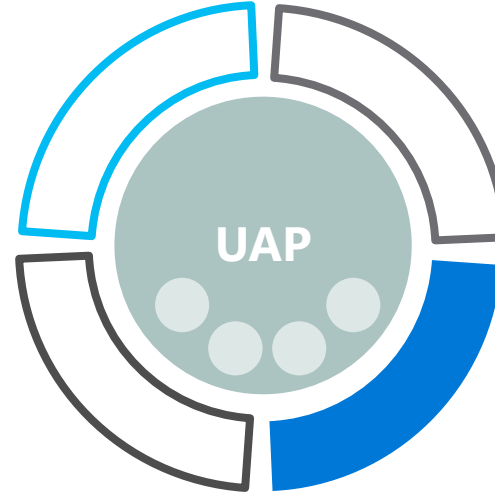
Windows Core

Desktop



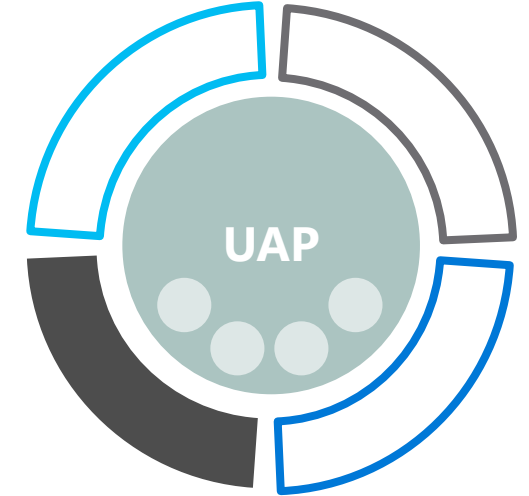
Windows Core

Mobile



Windows Core

Xbox



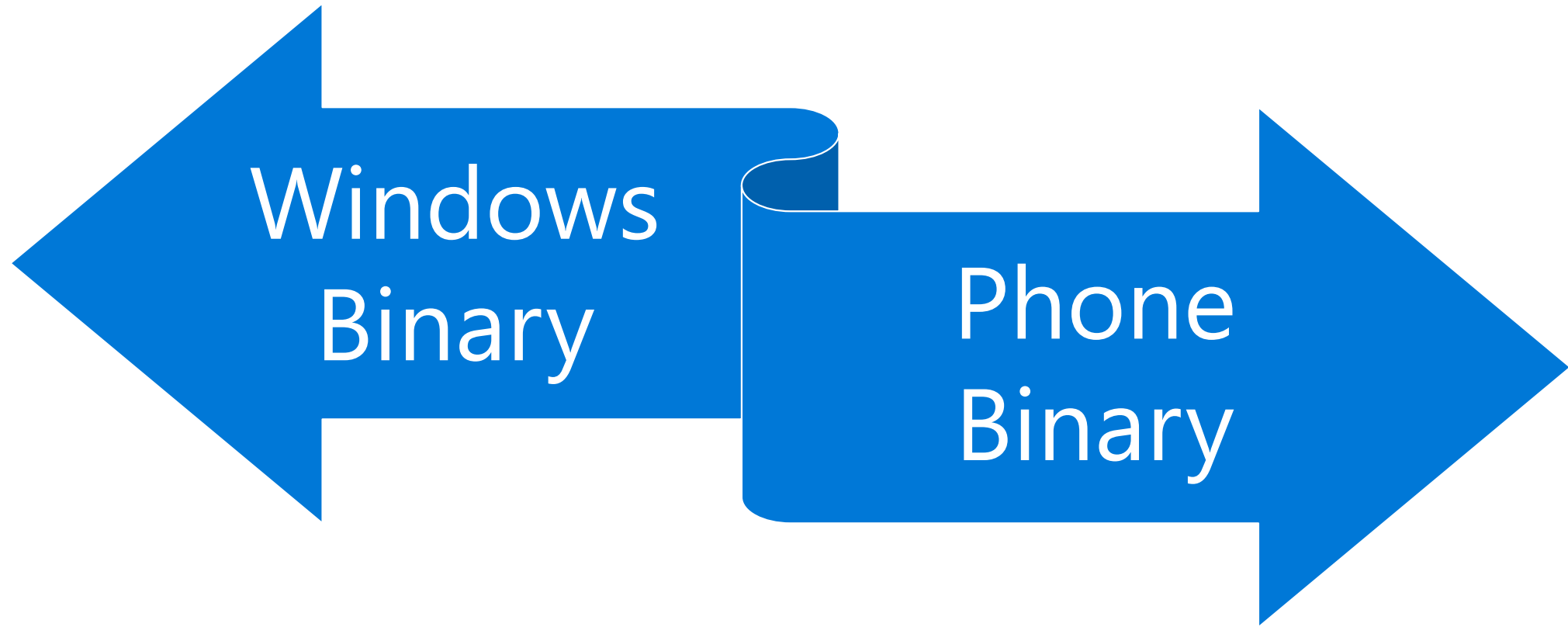
Windows Core

More...

Adaptive
Code

Run on Different Devices

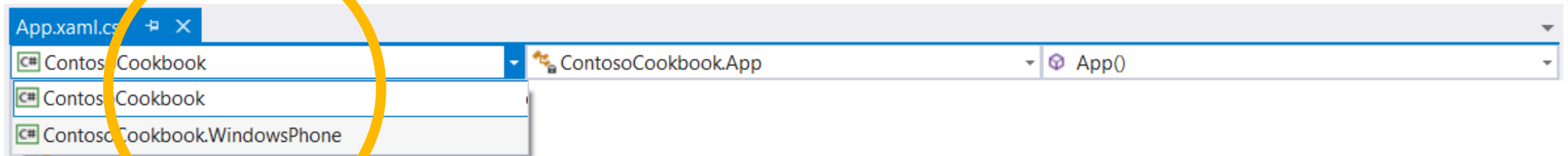
Shared Projects



Windows
Binary

Phone
Binary

Not all APIs were available everywhere



Windows.Phone.UI.Input.HardwareButtons.

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

Compilation directives

- C# Syntax

- `#if WINDOWS_PHONE_APP`
 `Windows.Phone.UI.Input.HardwareButtons`
 `.BackPressed += this.HardwareButtons_BackPressed;`
▪ `#endif`

- C++ Syntax

- `#if WINAPI_FAMILY==WINAPI_FAMILY_PHONE_APP`
 `_backPressedEventToken = HardwareButtons`
 `::BackPressed += ref new EventHandler`
 `<BackPressedEventArgs^> (this,`
 `&NavigationHelper::HardwareButton_BackPressed);`
▪ `#endif`

Feature Detection

Testing for capabilities

Windows.Foundation.Metadata.[ApiInformation](#)

IsApiContractPresent

IsEnumNamedValuePresent

IsEventPresent

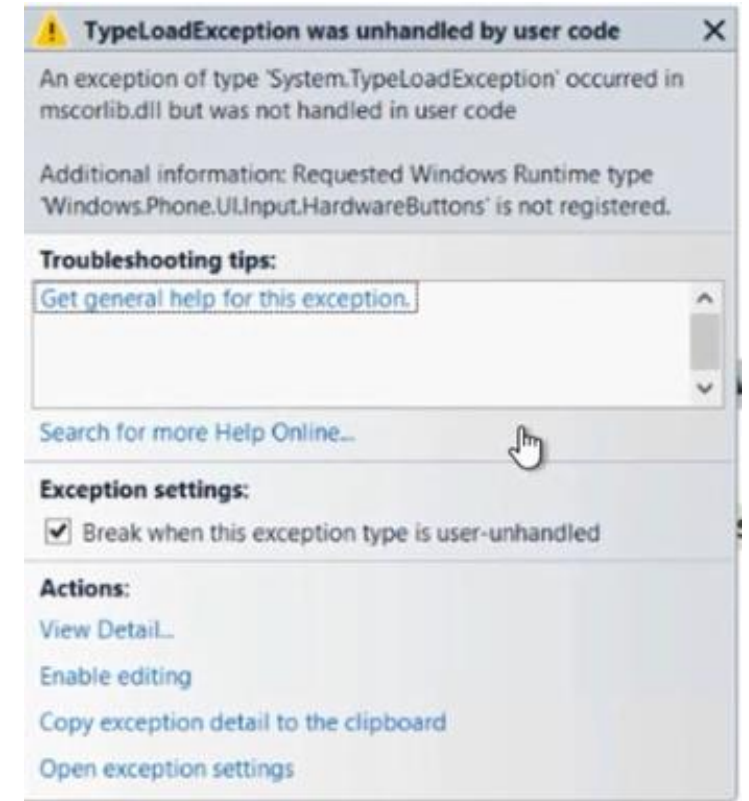
IsMethodPresent

IsPropertyPresent

IsReadOnlyPropertyPresent

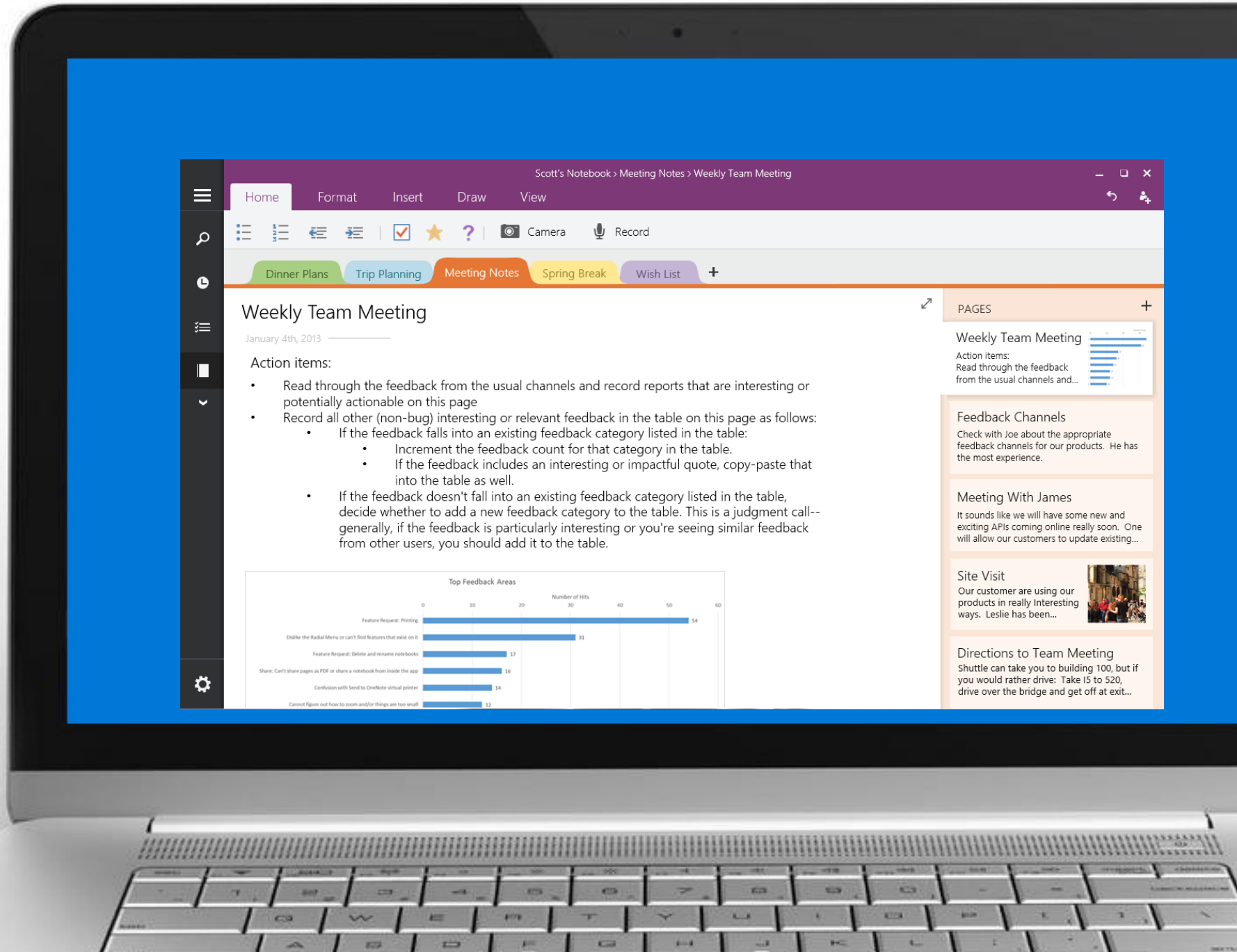
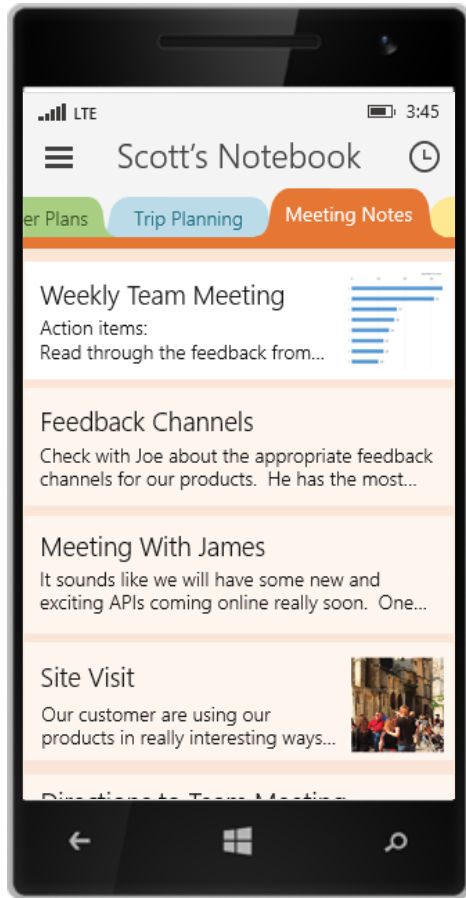
IsTypePresent

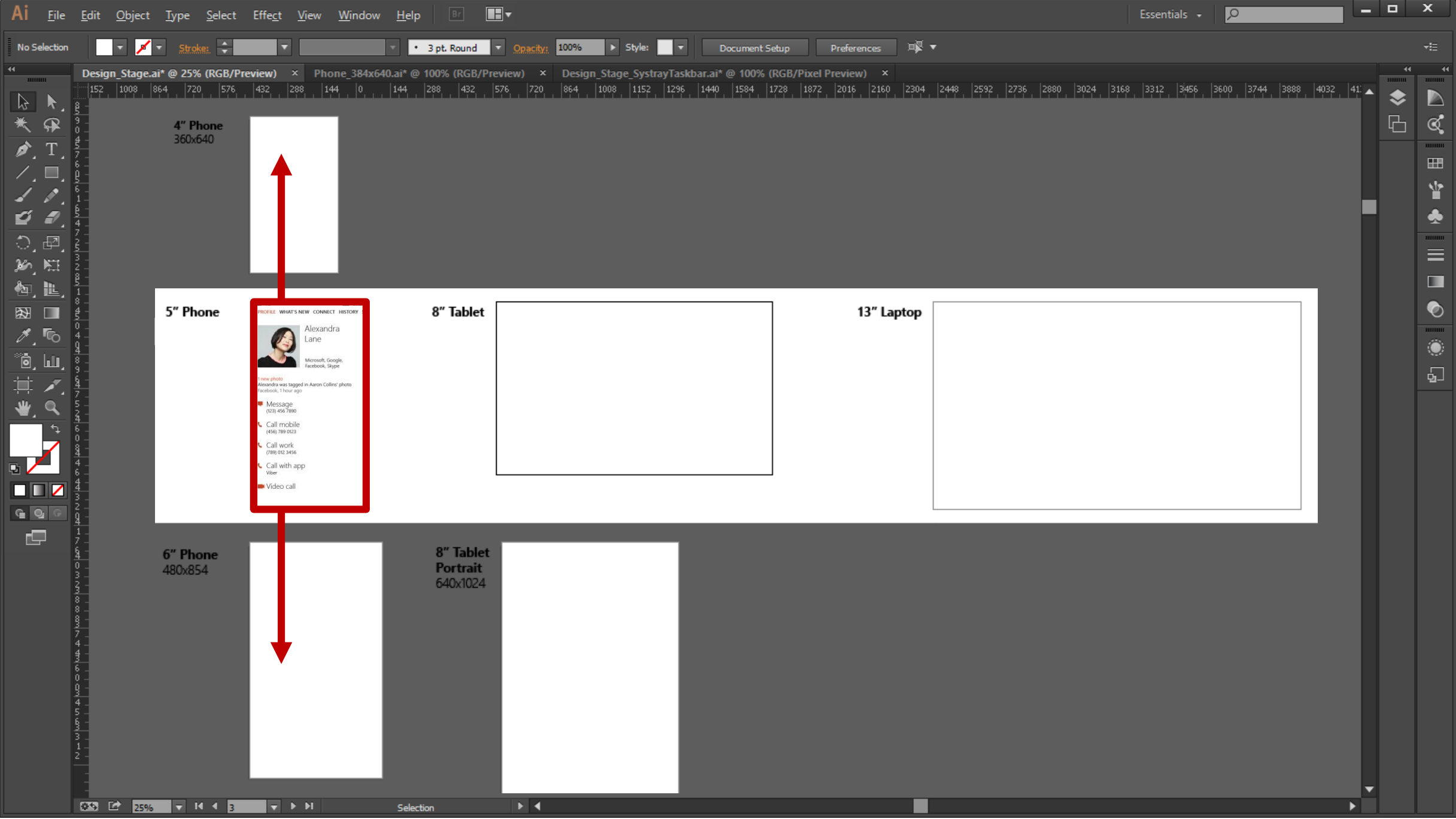
IsWriteablePropertyPresent



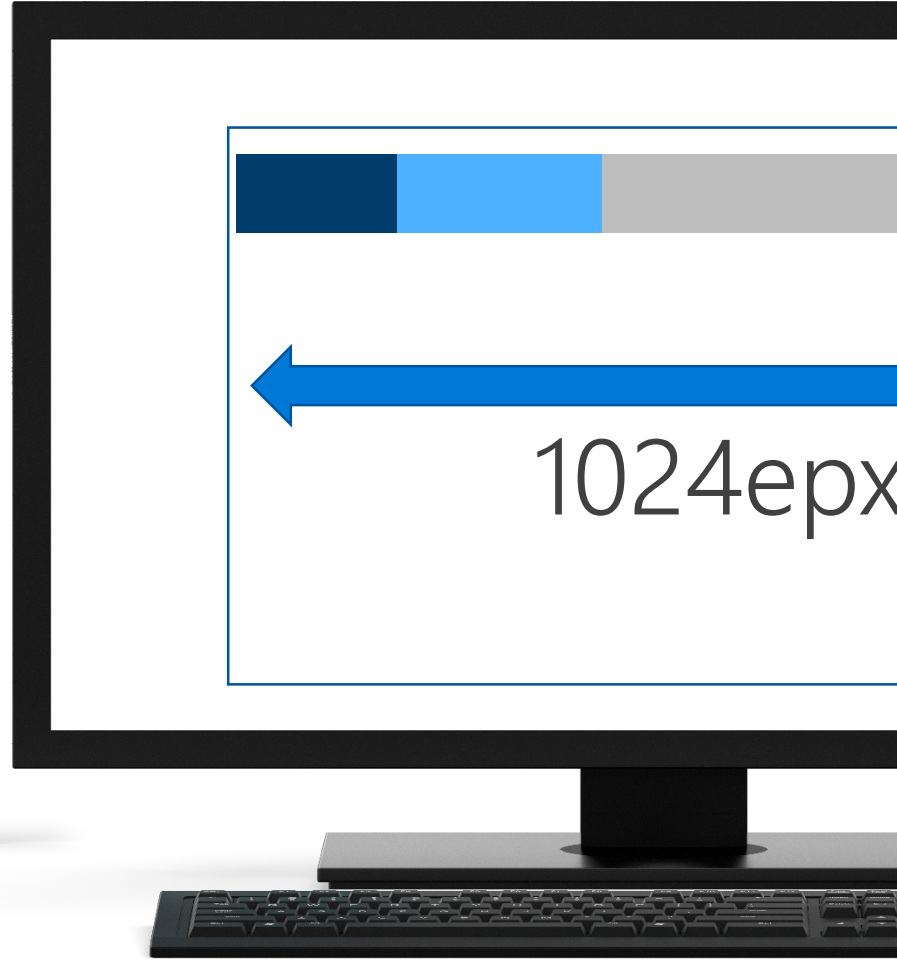
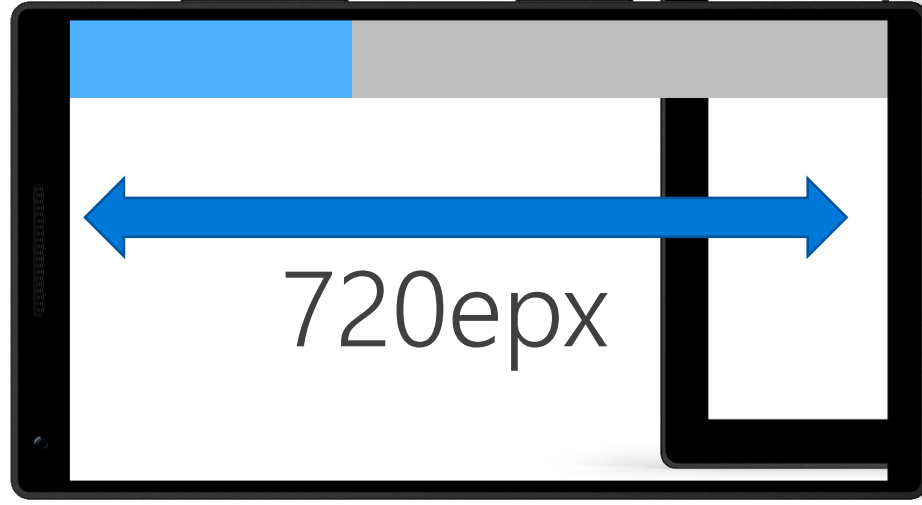
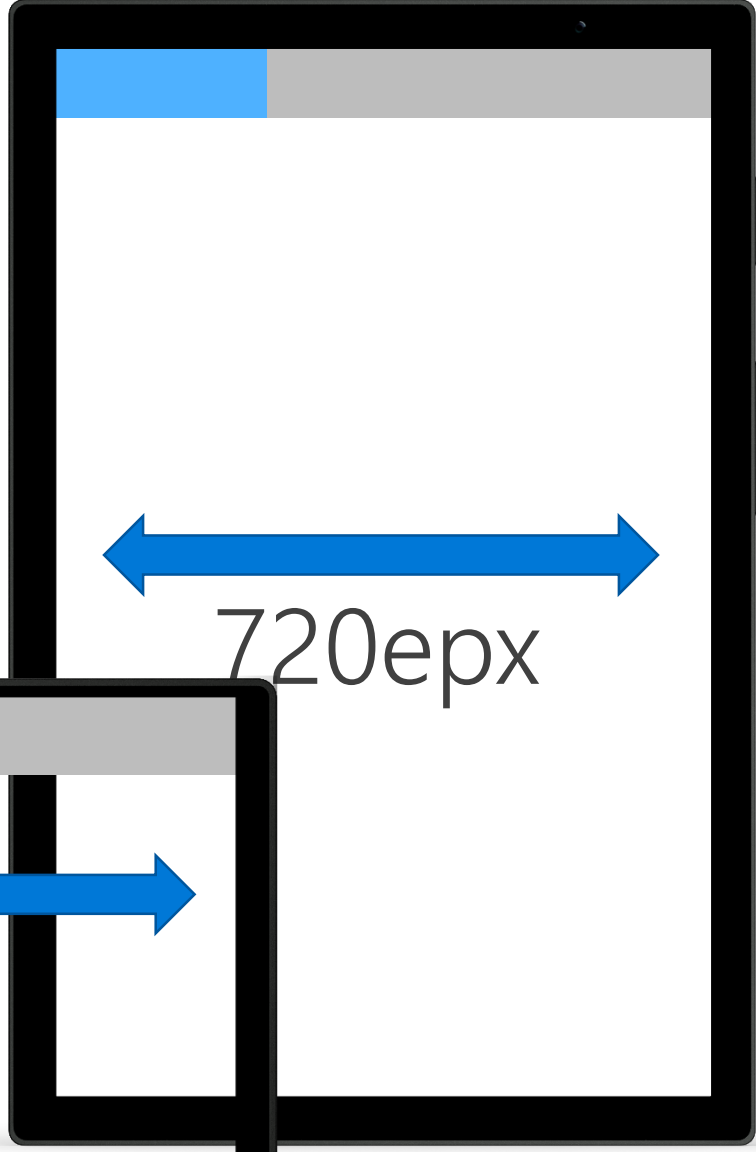
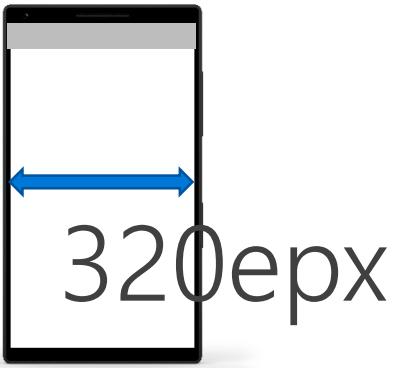
Adaptive UI

Adaptive UI

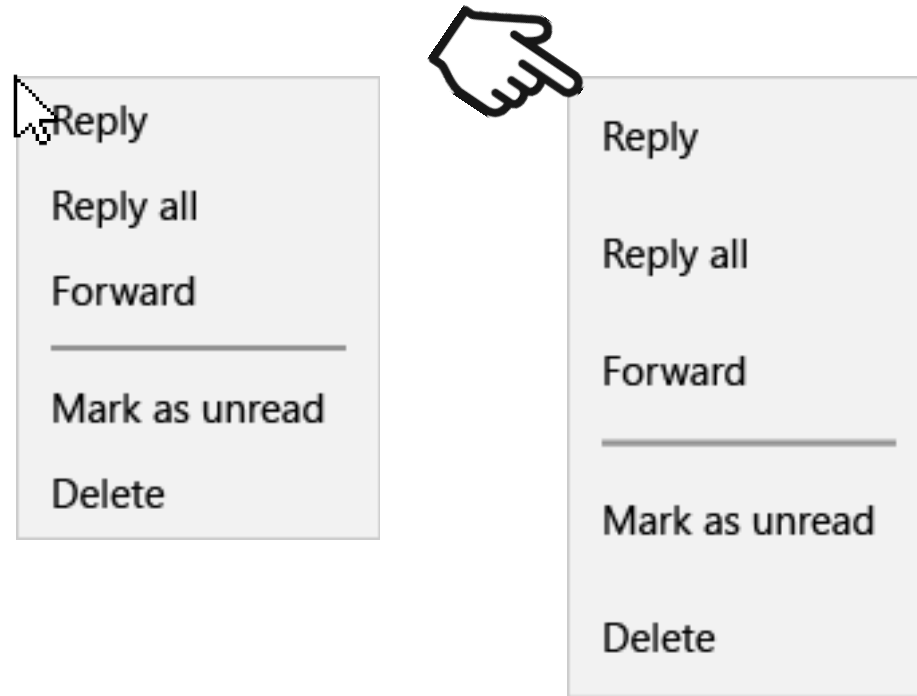




Breakpoints

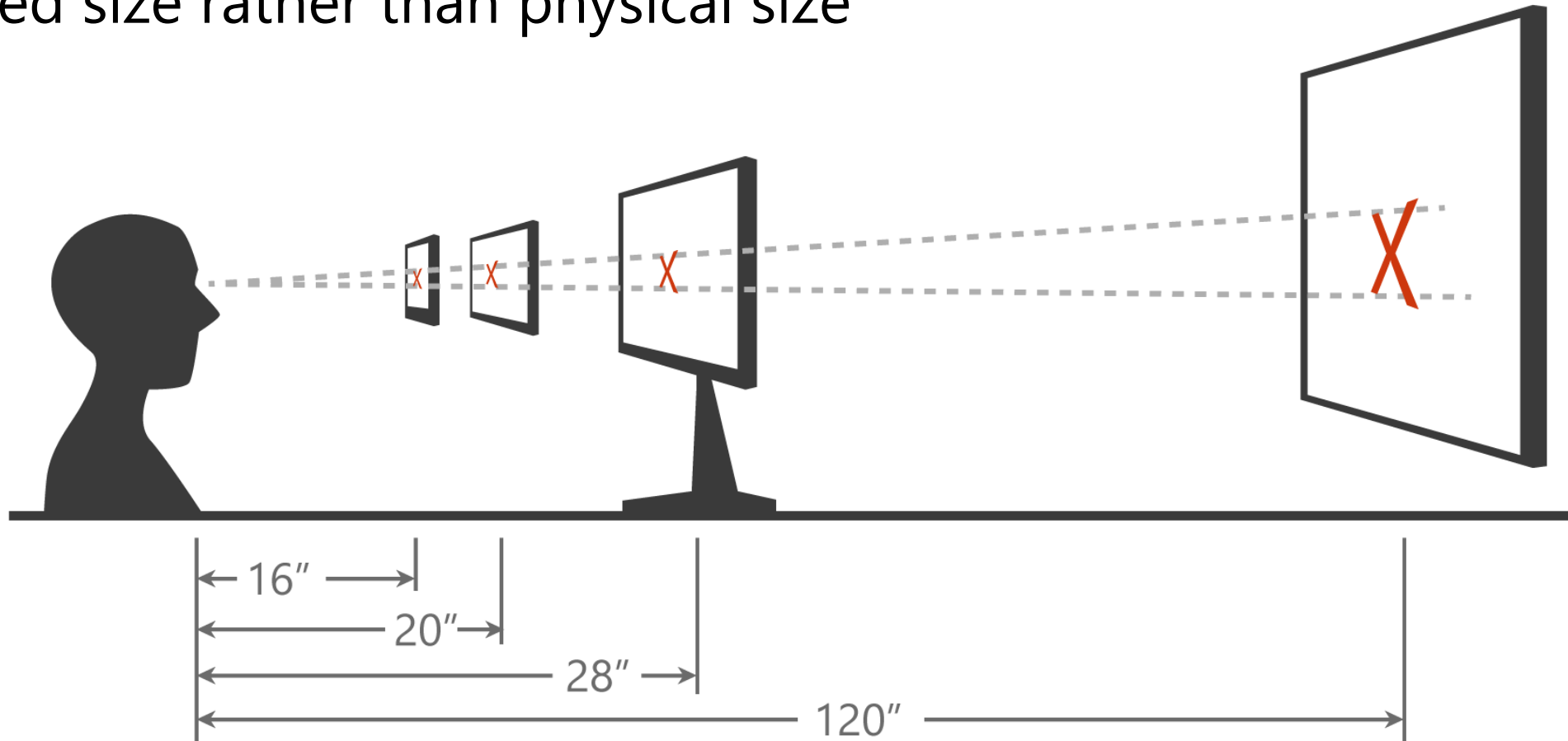


Input-Aware Context Menu



Scaling algorithm & effective pixels

Takes **viewing distance** and **pixel density** into account to optimize for perceived size rather than physical size



Introducing the Relative Panel

Windows XAML layout controls

Grid

Stack
Panel

Canvas

Scroll
Viewer

Border

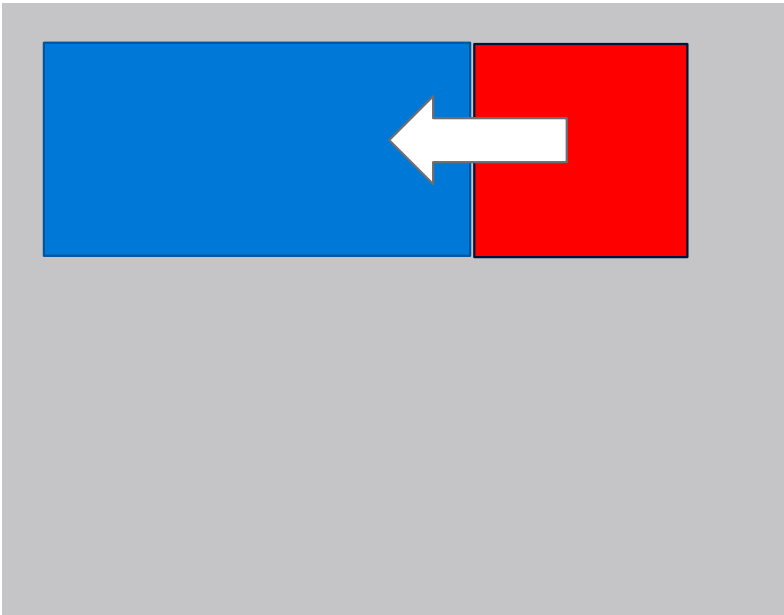
View Box

Wrap
Grid

**Relative
Panel**

Relative Panel is a XAML layout control. It arranges children by declaring relationships between them.

Relative Panel (XAML)



```
<RelativePanel>
```

```
  <Rectangle x:Name="BlueRect"  
    Height="100" Width="100" Fill="Blue" />
```

```
  <Rectangle x:Name="RedRect"  
    Height="100" Width="100" Fill="Red"  
    RelativePanel.RightOf="BlueRect"  
    RelativePanel.AlignVerticalCenterWith="BlueRect" />
```

```
</RelativePanel>
```

Simplify the visual tree

```
<Grid>  
  <StackPanel>  
    <StackPanel>  
      <TextBlock />  
      <TextBlock />  
    </StackPanel>  
    <StackPanel>  
      <TextBlock />  
      <TextBlock />  
    </StackPanel>  
  </StackPanel>  
</Grid>
```

```
<RelativePanel>  
  <TextBlock />  
  <TextBlock />  
  <TextBlock />  
  <TextBlock />  
</RelativePanel >
```

Visual State Triggers

```
<VisualStateGroup x:Name="WindowSizeStates">  
  <VisualState x:Name="WideState">  
    <VisualState.StateTriggers>  
      <AdaptiveTrigger MinWindowWidth="720" />  
    </VisualState.StateTriggers>  
    <!-- - more - - !>  
  </VisualState>  
</VisualStateGroup>
```

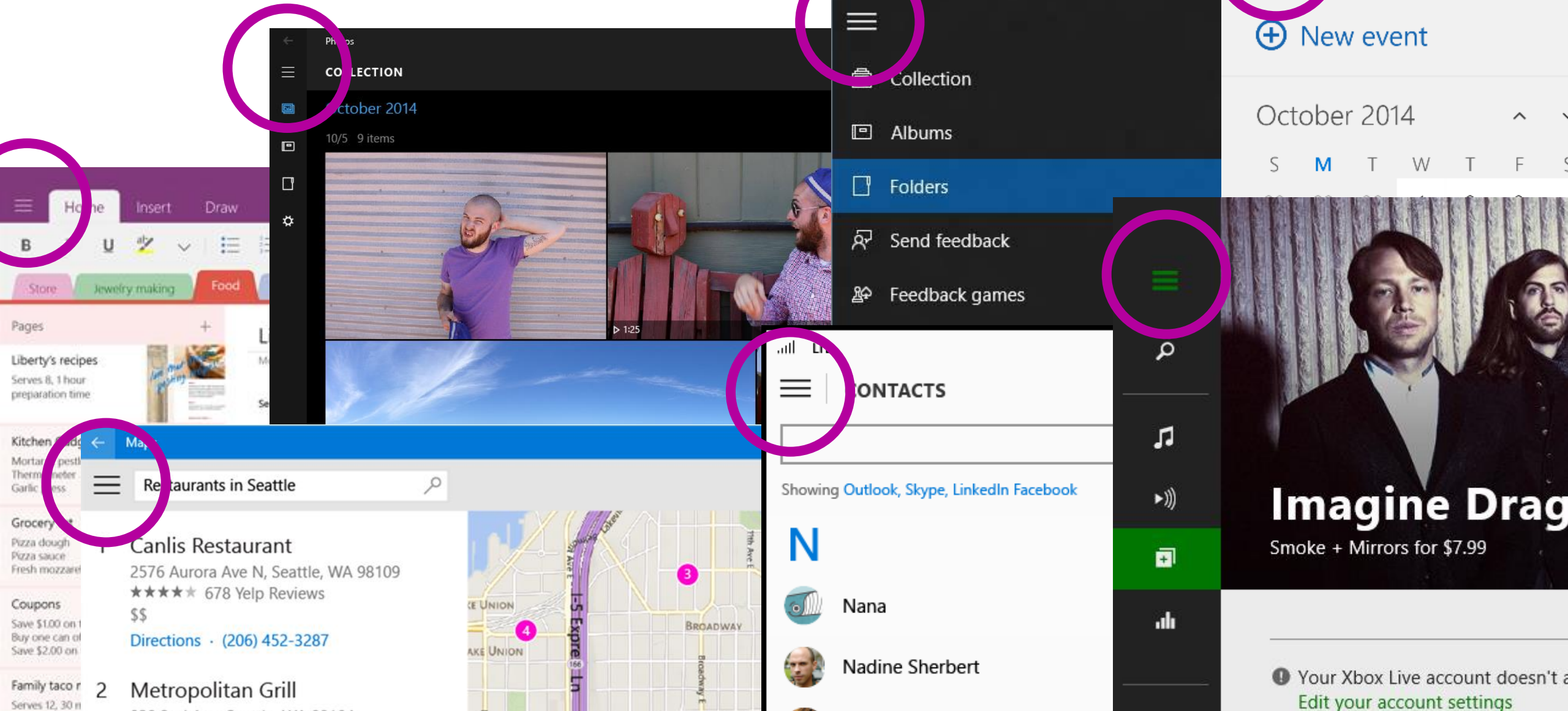
Visual State Setters (code snippet)

```
<VisualState x:Name="Pressed">
  <Storyboard>
    <ObjectAnimationUsingKeyFrames Storyboard.TargetName="RootGrid"
      Storyboard.TargetProperty="Background">
      <DiscreteObjectKeyFrame KeyTime="0" Value="{ThemeResource
        SystemControlBackgroundBaseMediumLowBrush}" />
    </ObjectAnimationUsingKeyFrames>
    <ObjectAnimationUsingKeyFrames Storyboard.TargetName="ContentPresenter"
      Storyboard.TargetProperty="BorderBrush">
      <DiscreteObjectKeyFrame KeyTime="0" Value="{ThemeResource
        SystemControlHighlightTransparentBrush}" />
    </ObjectAnimationUsingKeyFrames>
    <ObjectAnimationUsingKeyFrames Storyboard.TargetName="ContentPresenter"
      Storyboard.TargetProperty="Foreground">
      <DiscreteObjectKeyFrame KeyTime="0" Value="{ThemeResource
        SystemControlHighlightBaseHighBrush}" />
    </ObjectAnimationUsingKeyFrames>
  </Storyboard>
</VisualState>
```

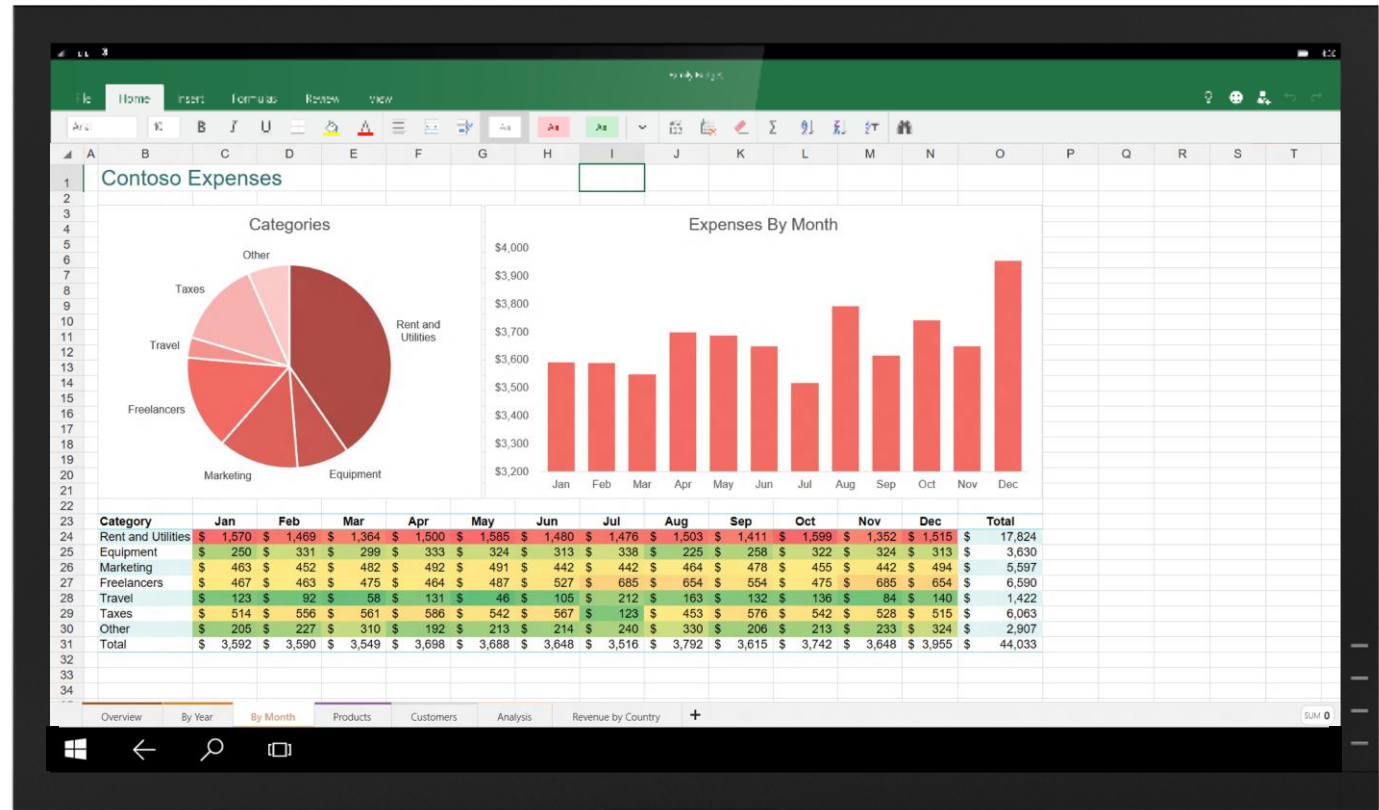
Relative Panel

DEMO

SplitView



Introducing Continuum



ScreenBeam PRO

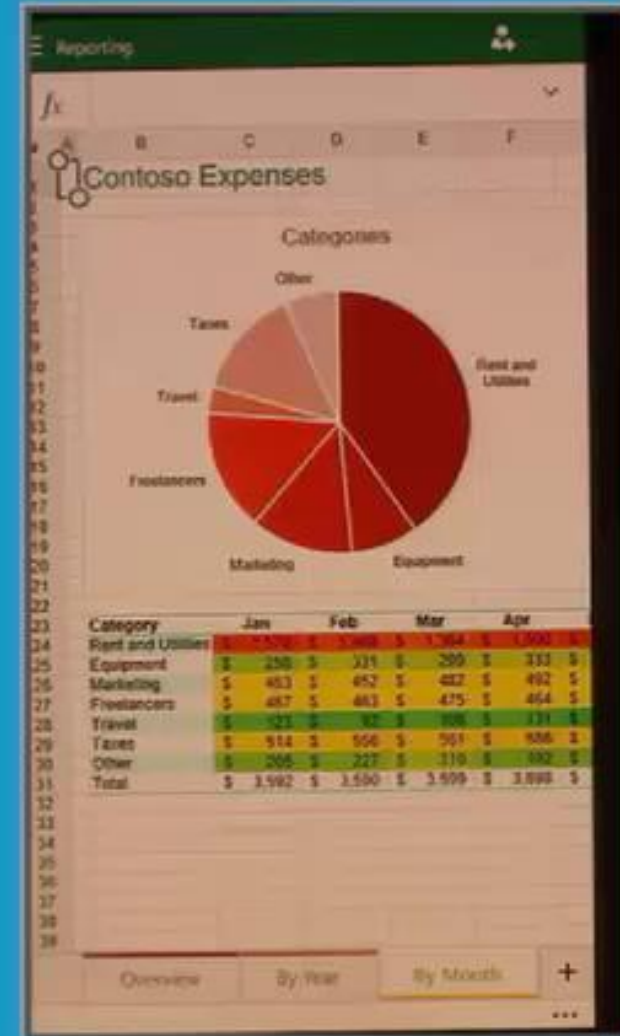
Ready To Connect

Learn how to connect your device:
actiontec.com/setupsb

Receiver name: Screenbeam

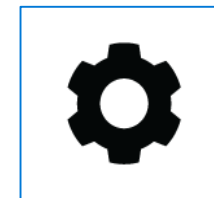
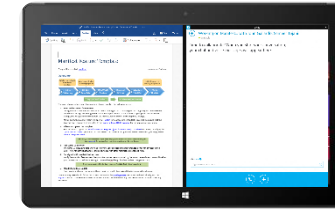
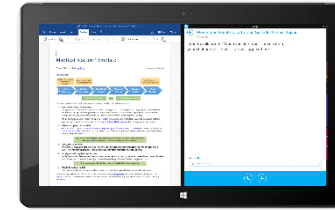
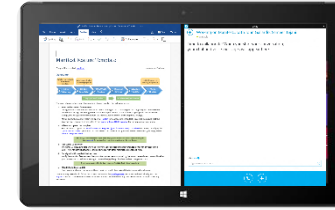
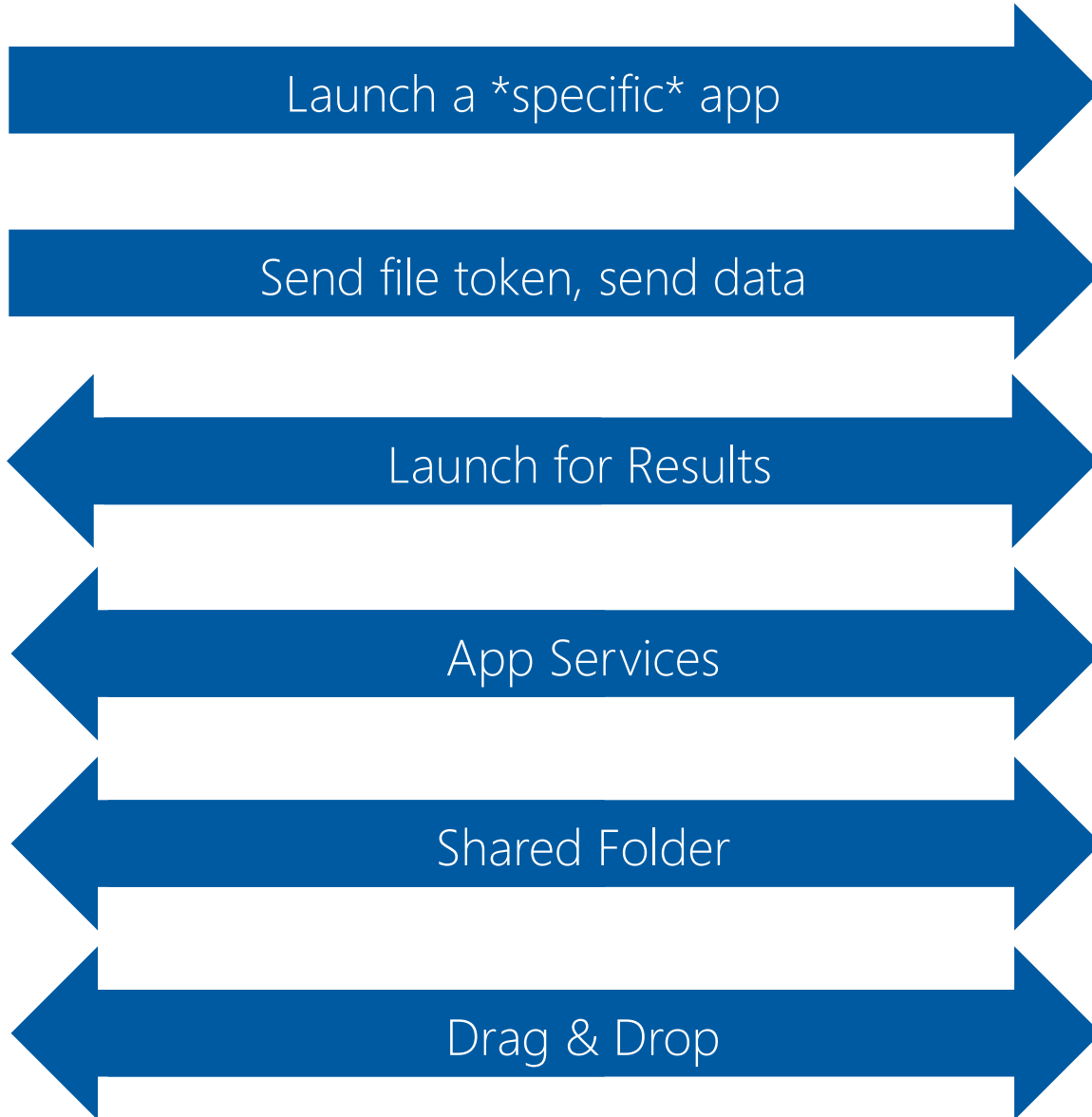
Firmware: 1.3.1.1
Model: SEWD100B

OPTIMIZED FOR



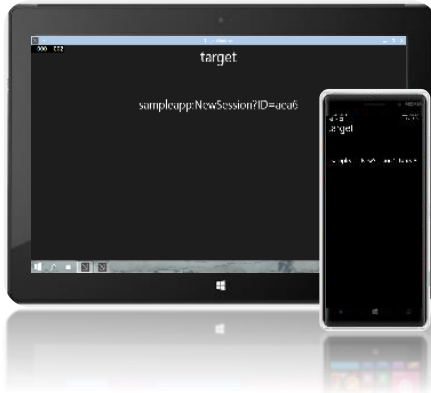
App to App in Windows 10 UWP

Enhanced App to App in Windows 10

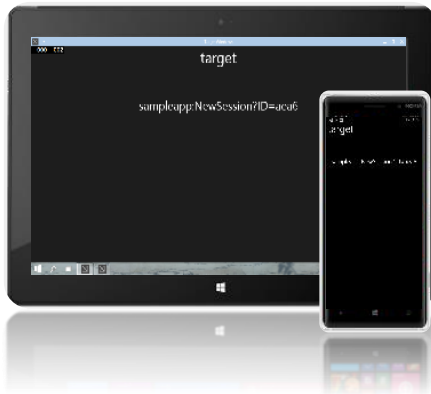


App Services

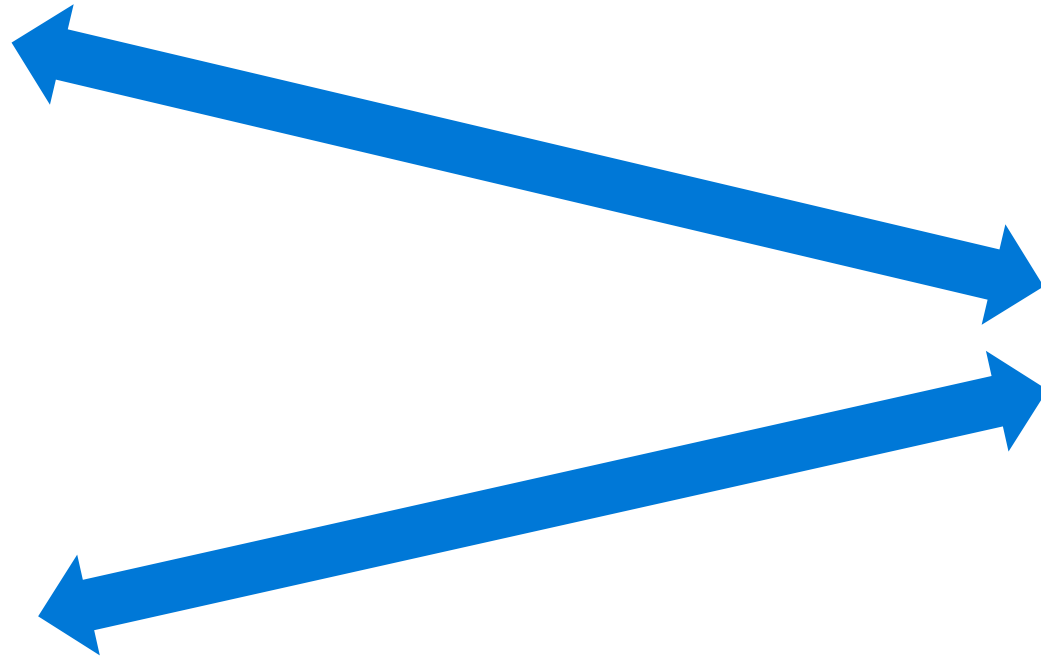
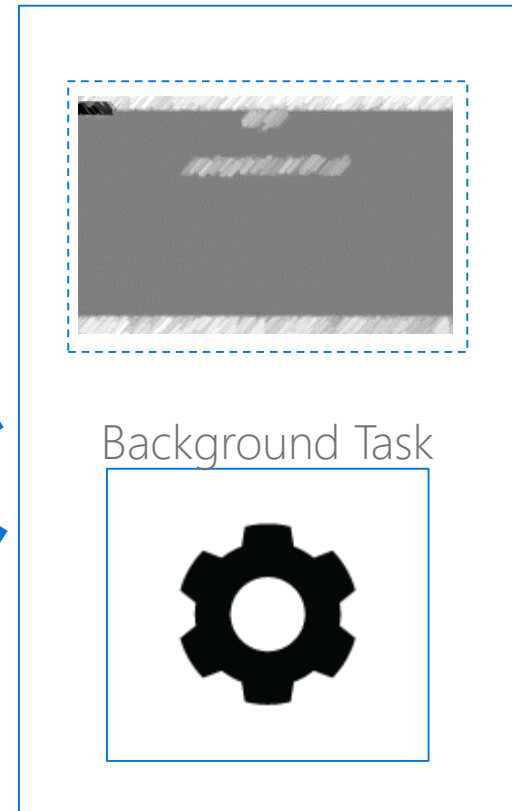
Client App A



Client App B



App with App Service



Reusing your investments & Bridges

1 Billion Windows 10 Devices



Windows Store
Universal Windows Platform

Web

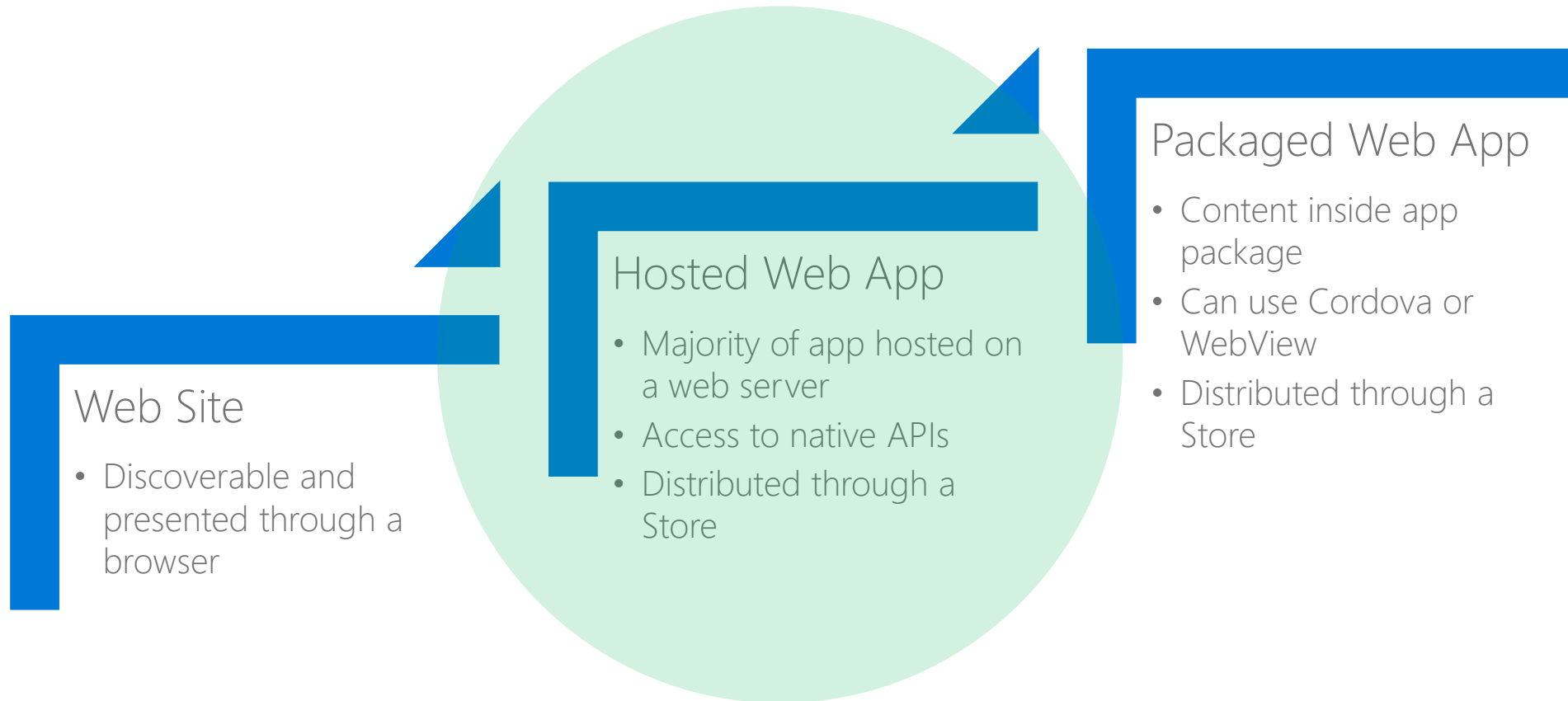
.NET & Win32

Android
Java/C++

iOS
Objective C

Project Westminster „Hosted Web App“

Bring Web Apps to the Windows Store



Hosted Web App

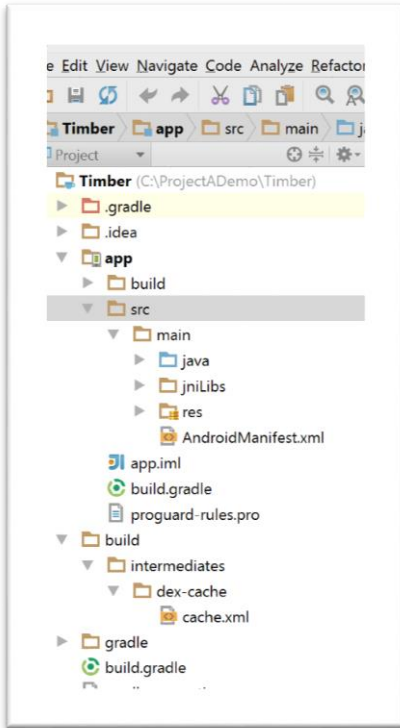
DEMO



Project Astoria

Bring Android Apps to the Windows Store for Phones

Your Android Code
(Project Astoria SDK & App analysis)



Your IDE
(Project Astoria Plugins)

IntelliJ

Android
Studio

Eclipse

Your dev
machine

Windows

Mac

Windows
Magic

Live tiles

bing ads



XBOX LIVE

...

Your app in the
Windows Store



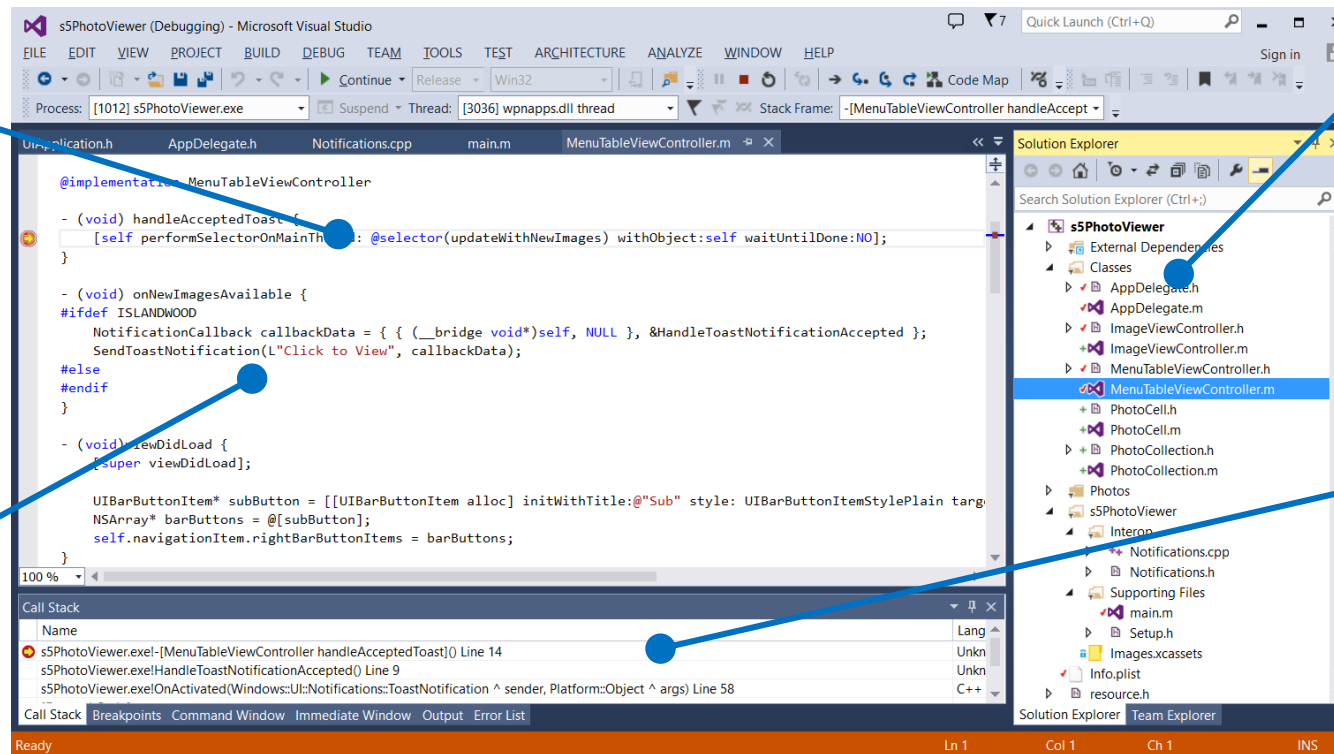
Project Islandwood

Bring iOS Apps to the Windows Store

Compile Objective-C for Windows in Visual Studio

Integrated language support:
syntax highlighting,
autocomplete

Universal API Interop

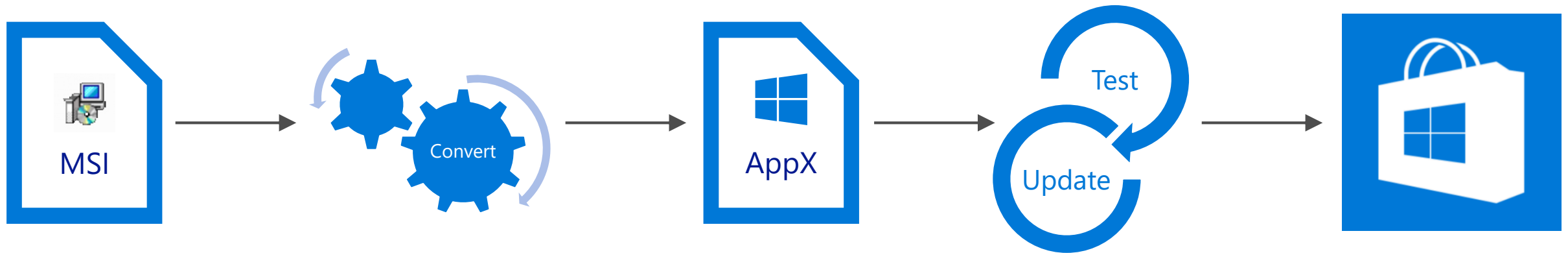


Xcode projects imported to Visual Studio

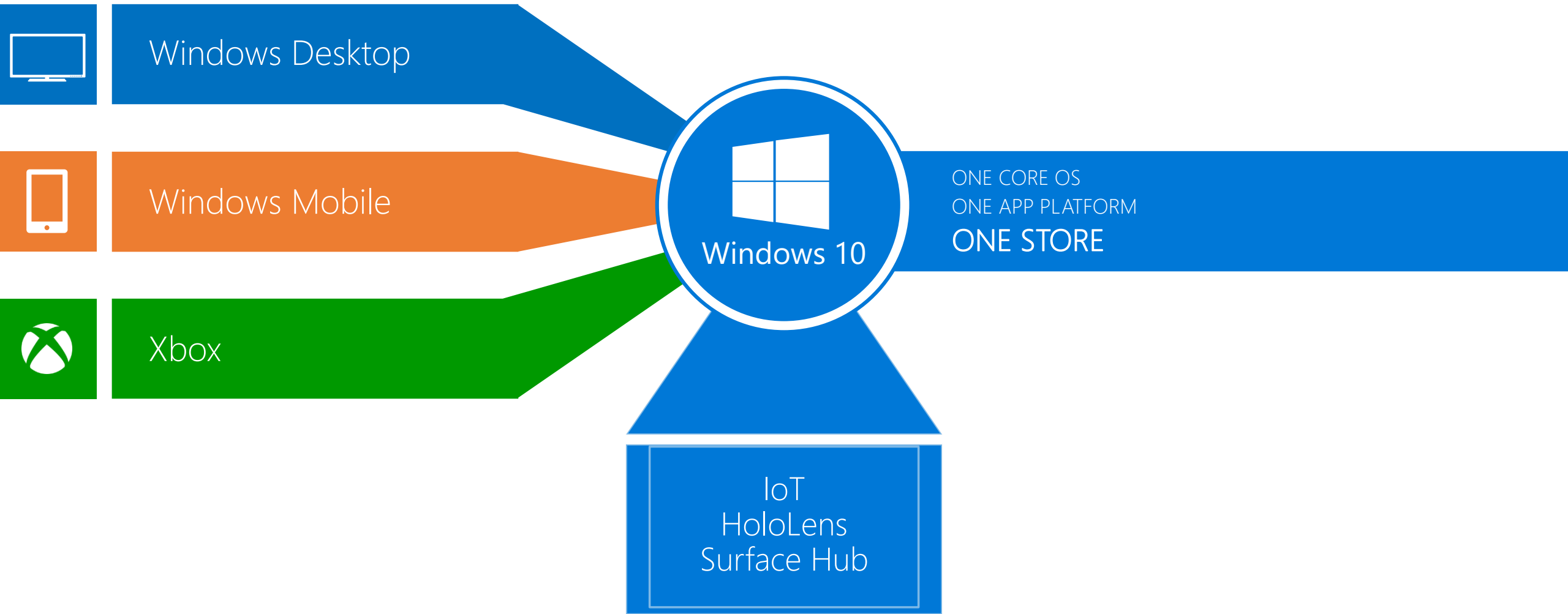
Full debugging: breakpoints, stack traces, ...

Project Centennial

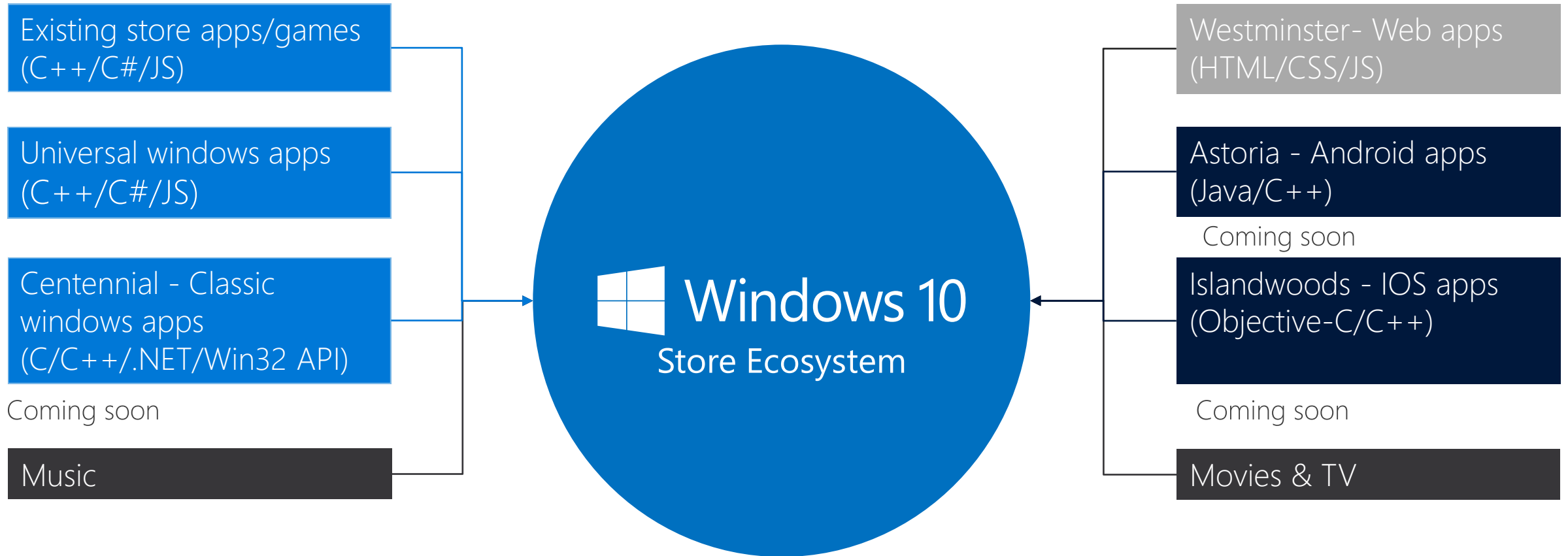
Bring Classic Windows Applications to the Windows Store



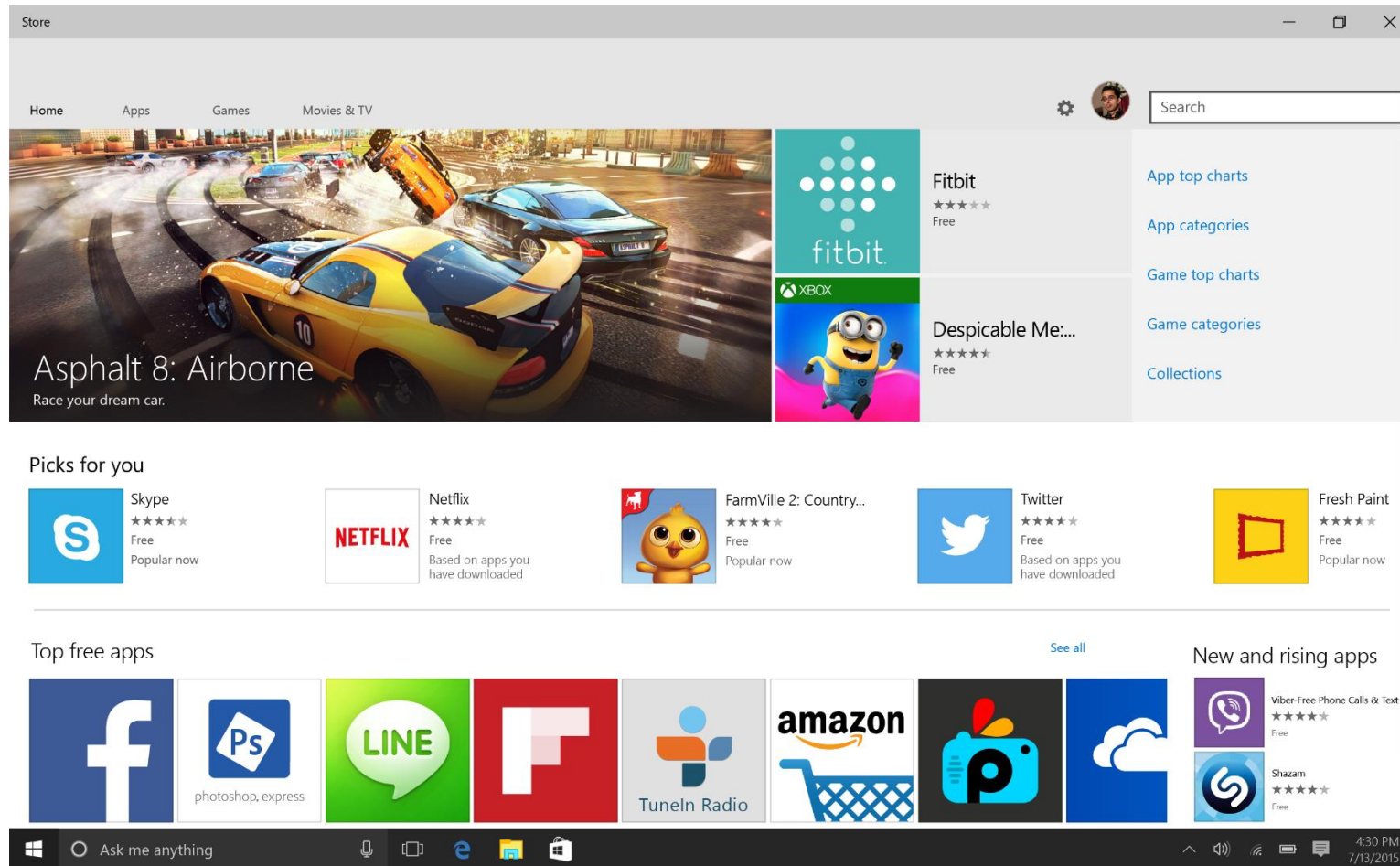
Windows 10



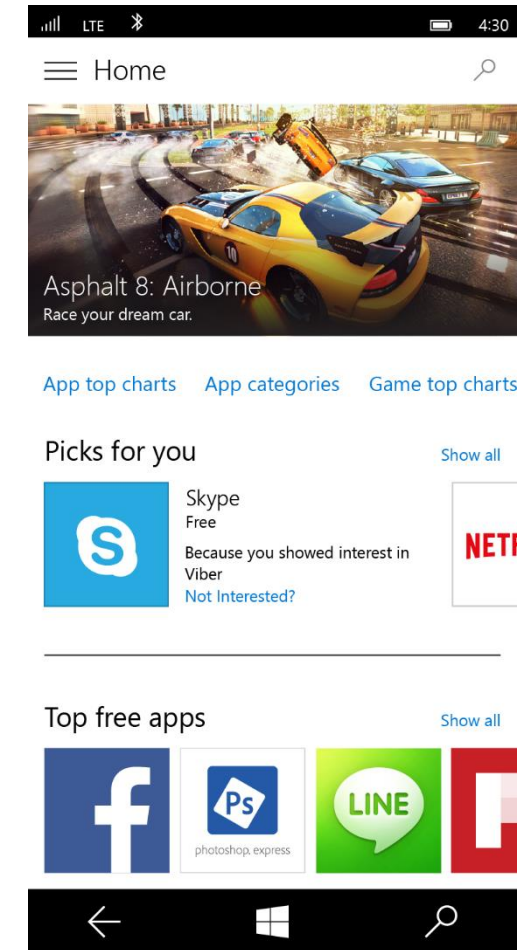
Content support in windows 10 store



One store for all windows 10 devices



Desktop



Mobile

Keeping the best features

	Windows Phone 7.x, 8.x	Windows 8.x	Windows 10
Hidden apps	✓		✓
Per market pricing	✓		✓
Independent IAP publishing	✓		✓
Betas	✓		✓
Time based trials		✓	✓
App discounts		✓	✓
Scheduled publish		✓	✓

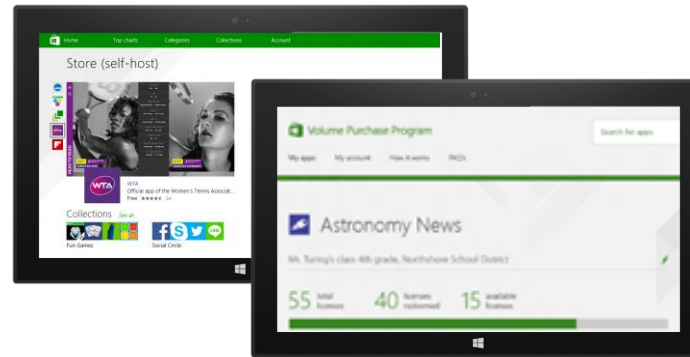
Windows 10 at a Glance

Windows Store



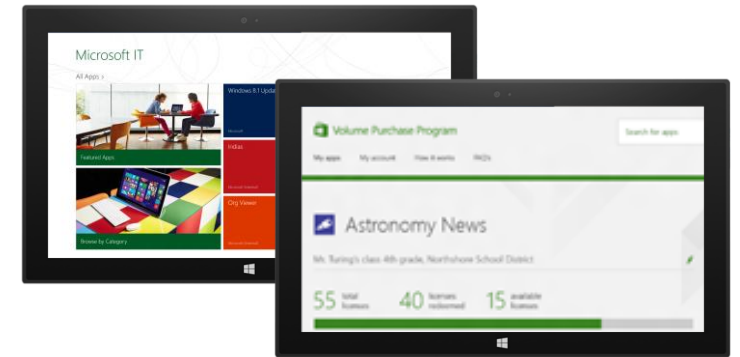
- Modern apps
- Sign in with MSA
- Pay with credit card, gift card, PayPal, Alipay, INICIS, mobile operators

Business Store



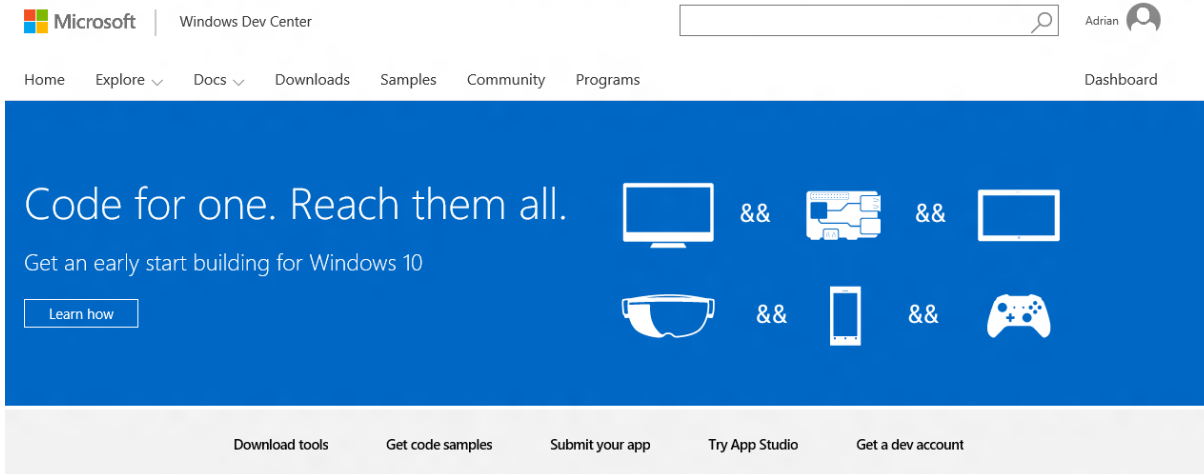
- Modern apps
- Leverages Azure Active Directory for administration, some scenarios
- Private organization store for the org's preferred or LOB apps
- Pay with credit card or PO/invoice
- Deploy modern apps offline, in images, and more
- Modern app license management

"Company Portal"



- Sideload line-of-business modern apps
- Deploy apps from the Windows Store (even when the Store UI is disabled) as well as uploaded LOB apps through Business Store integration using MDM

Introducing One Windows Dev Center



Get started

Ready to create your first Windows app? Our tutorials help you target all Windows devices from a single project, in the language you prefer.



Design and UI

User interface basics, design principles, layout guidelines, templates, and more.



Develop great apps

Guidance, reference, tutorials, and code snippets help you build your app, whether it's brand new or ported from another platform.



Publish apps

Learn how to get your app into the Store today, and see what's coming with our new dashboard preview.

One destination for supporting all Windows platforms

- ✓ One registration
- ✓ One app submission
- ✓ Customer acquisition
- ✓ Customer engagement
- ✓ Powerful monetization
- ✓ Rich analytics
- ✓ One payout

Thank you!

Download Visual Studio 2015

<https://www.visualstudio.com/en-us/downloads/visual-studio-2015-downloads-vs.aspx>

Samples

<https://github.com/Microsoft/Windows-universal-samples>

<https://github.com/DanielMeixner/w10demoking>

<https://github.com/Windows-XAML/201505-MVA>

Blog

<http://blogs.msdn.com/b/dmx>

Free Training

<http://www.microsoftvirtualacademy.com>

Ebook

<http://aka.ms/vs-tipps-ebook>



Marco Richardson

Technical Evangelist

Microsoft Deutschland GmbH

[@TileGarageMRich](https://twitter.com/TileGarageMRich)

<http://www.tilegarage.net/>

Daniel Meixner

Technical Evangelist

Microsoft Deutschland GmbH

[@DanielMeixner](https://twitter.com/DanielMeixner)

DevelopersDevelopersDevelopersDevelopers.NET