Windows 10
Universal Windows Platform

Michael Samarin
Futurice Oy
@MichaelSamarin

Phone

Phablet

Small Tablet

Large Tablet

2-in-1s
(Tablet or Laptop)

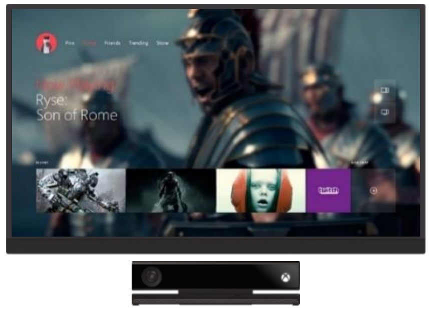
Classic
Laptop

Desktops
& All-in-Ones

Windows 10

Surface Hub

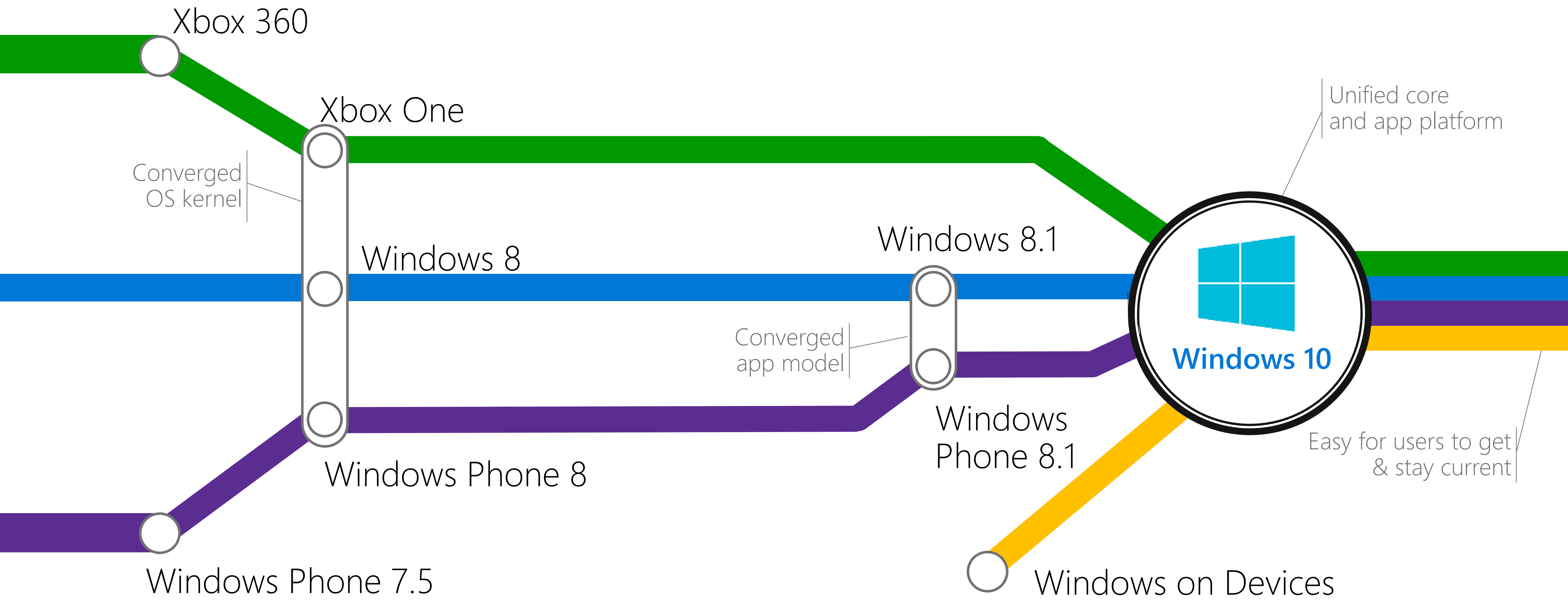
Xbox



Holographic

IoT

The convergence journey



Universal Windows Platform

One Operating System

One Windows core for all devices

One App Platform

Apps run across every family

One Dev Center

Single submission flow and dashboard

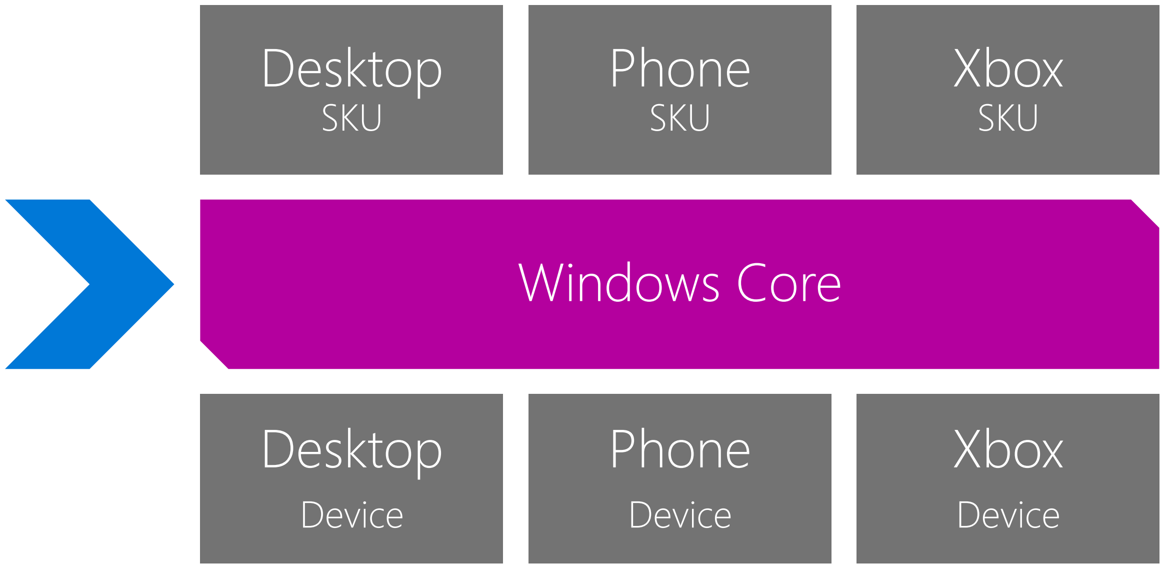
One Store

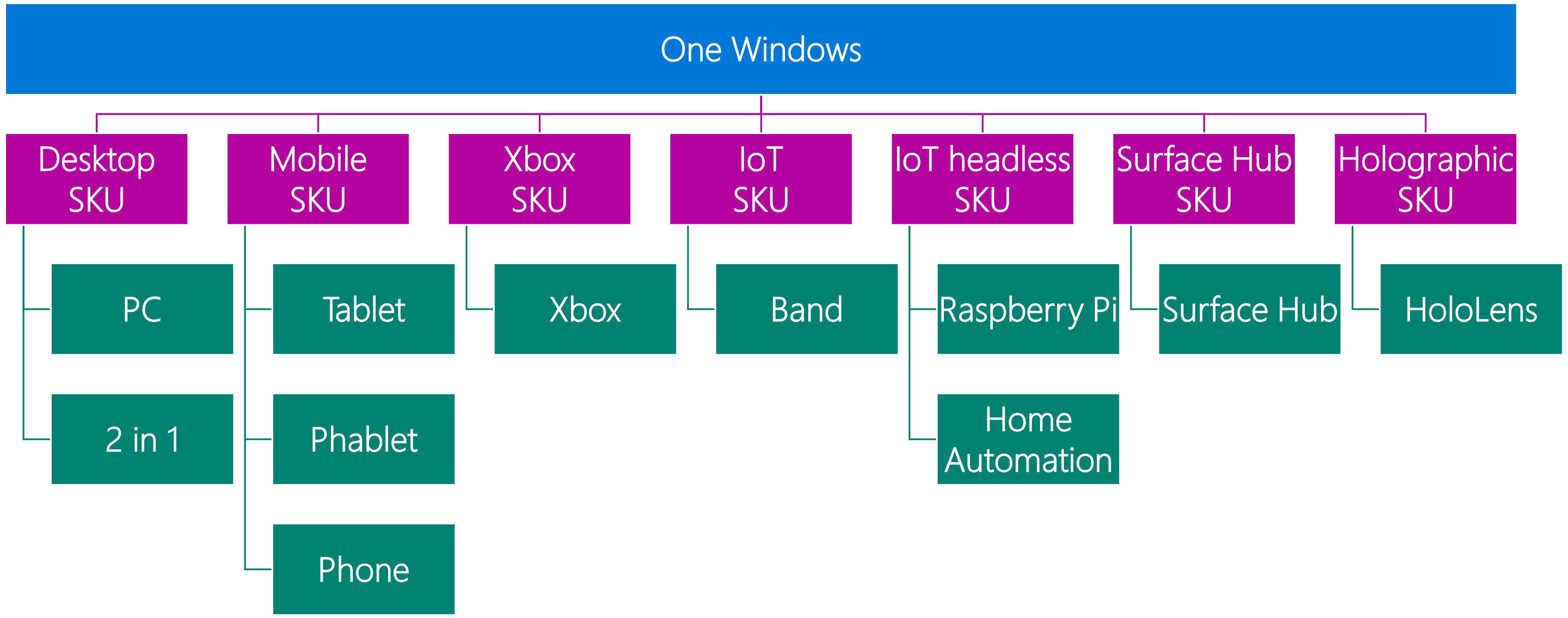
Global reach, local monetization
Consumers, Business & Education

Windows Core

The refactored common core

One hardware platform
Universal hardware driver
Standard network and I/O





http://windows.Microsoft.com

Each family adds features to the one it inherits

Windows

One simple, unified, integrated development environment

Windows

Visual Studio 15 IDE

Every project type

Desktop, Windows, Phone, Service, Web, Game, More...

Every developer task

Code edit, Architecture design, UX design, Debug, Profile, Review, Test, More...

Every development language

C++/CX, C#, Visual Basic, JavaScript, XAML, HTML, More...

Visual Studio Online

Source repository, project management, bug tracking, More...

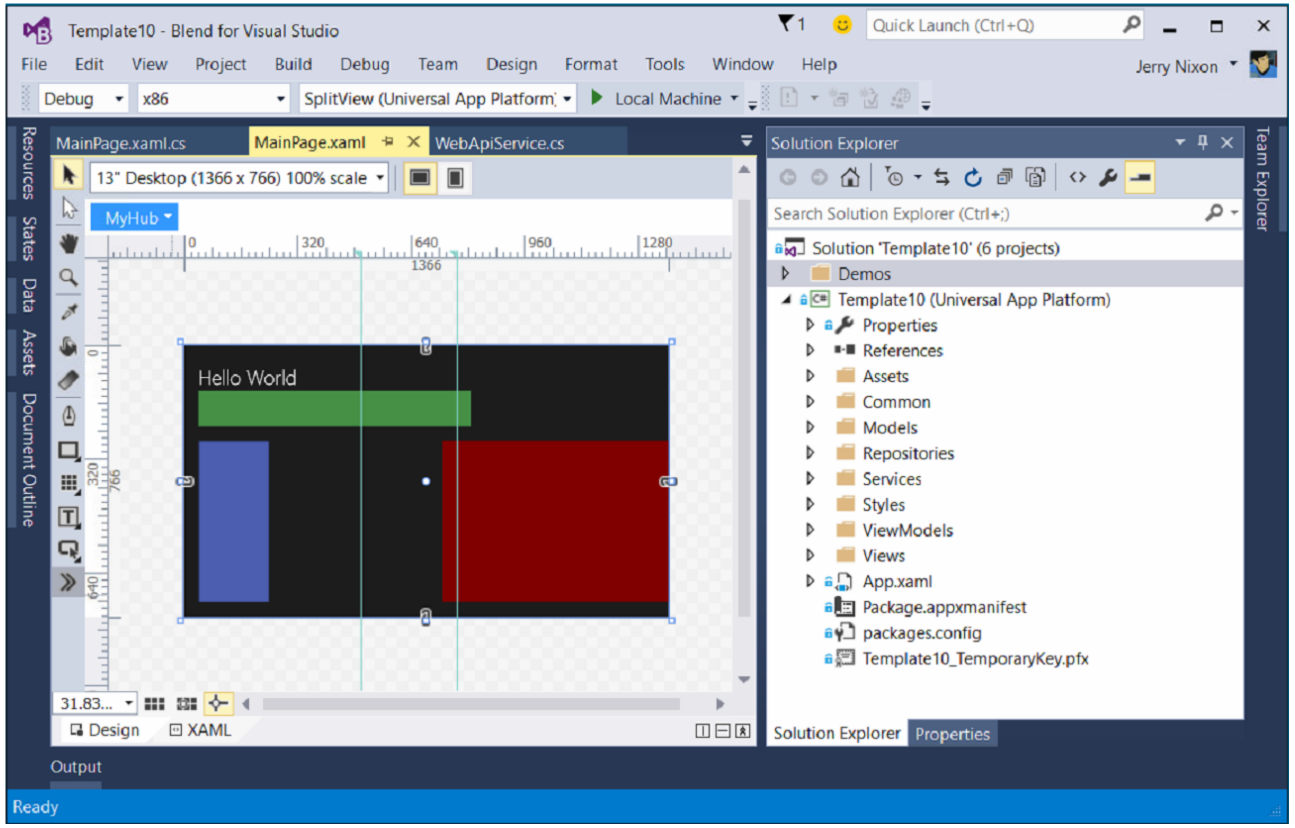
Blend for Visual Studio 15

The XAML Developer's IDE

Always part of Visual Studio
Uses the Visual Studio shell
Full auto-complete & intellisense

- Validation
- Snippets
- Peek

File & solution management
Resource management
Data management
Animation
States



Template10 - Blend for Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Design Format Tools Window Help
Jerry Nixon
Debug
x86
SplitView (Universal App Platform)
Local Machine
MainPage.xaml.cs
MainPage.xaml
WebApiService.cs
13" Desktop (1366 x 766) 100% scale
MyHub
Hello World
Resources States Data Assets Document Outline
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Template10' (6 projects)
Demos
Template10 (Universal App Platform)
Properties
References
Assets
Common
Models
Repositories
Services
Styles
ViewModels
Views
App.xaml
Package.appxmanifest
packages.config
Template10_TemporaryKey.pfx
Design
XAML
Solution Explorer
Properties
Team Explorer
Output
Ready

http://windows.Microsoft.com

Visual Studio 2015 Editions

Enterprise

Architecture Modeling, Diagnostics, VSO/ALM & Release Management

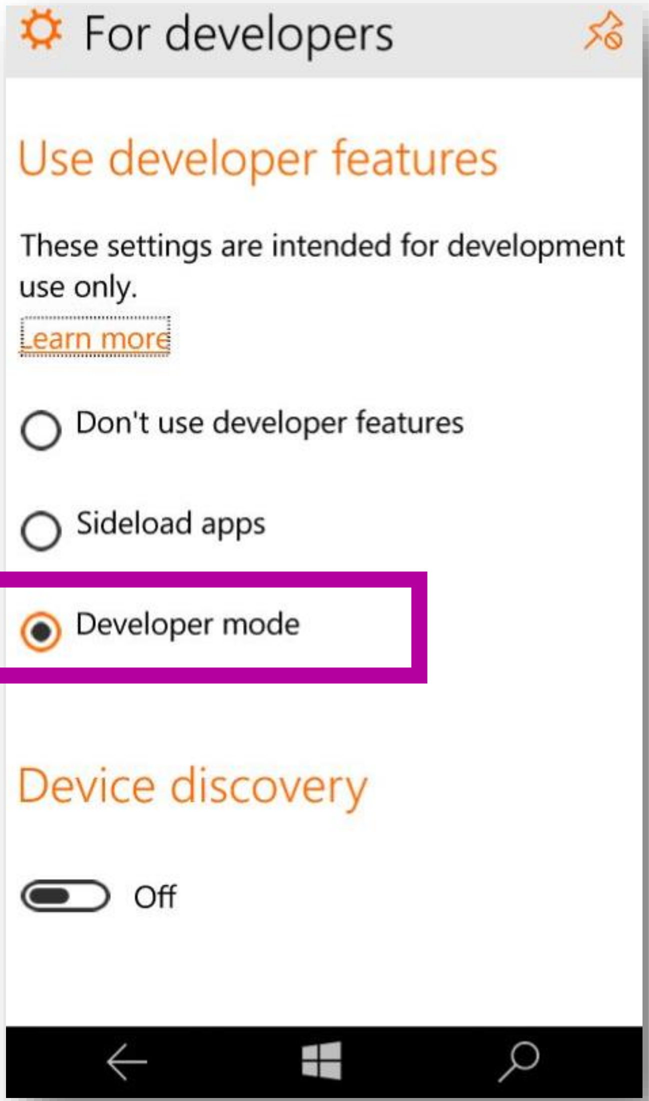
Professional

Architecture Validation, VSO/ALM & Feedback Management

Community Editions

Visual Studio Professional Edition

Developer unlock


For developers

Use developer features

These settings are intended for development use only.
Learn more

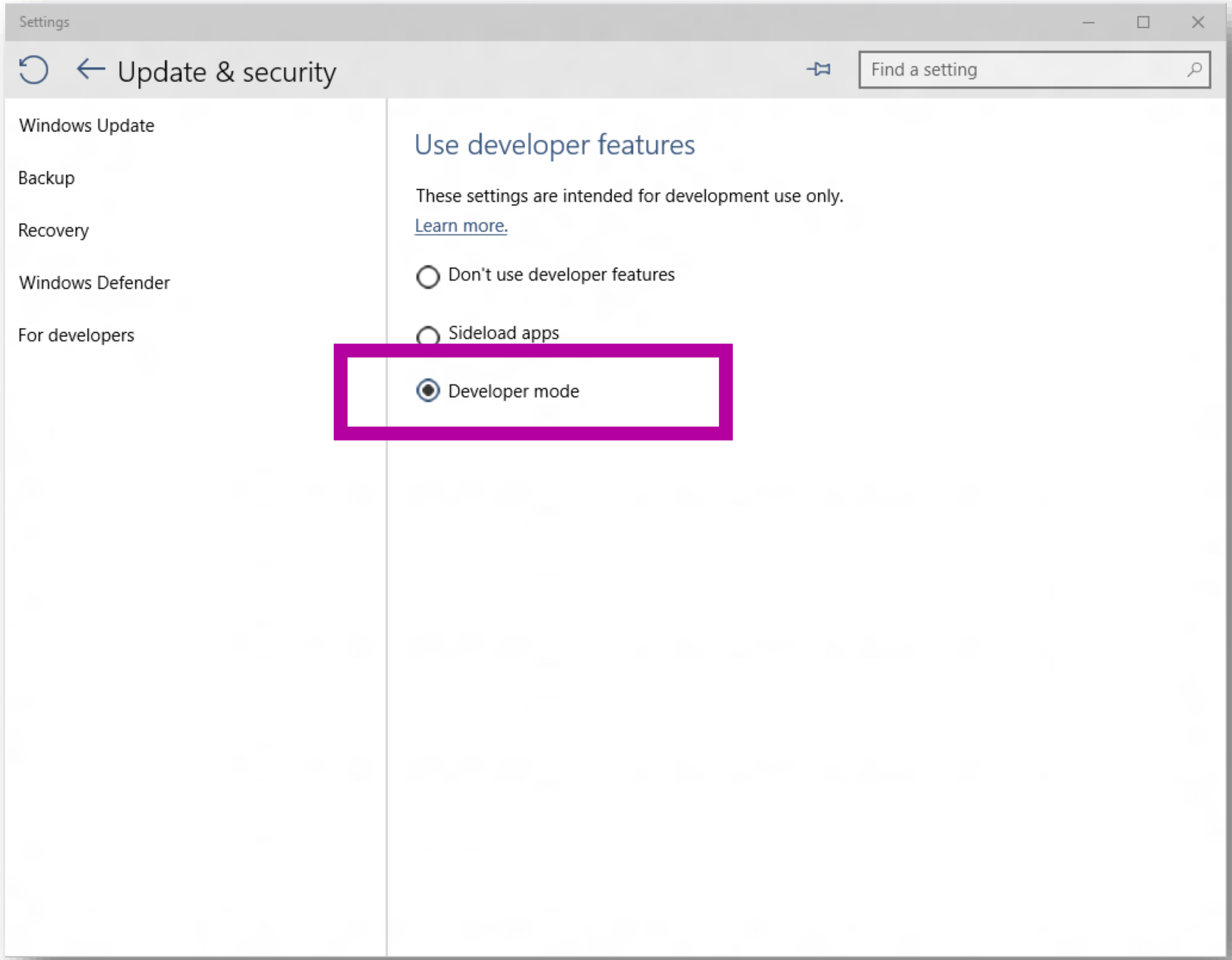
Don't use developer features

Sideload apps

[✓ Developer mode]

Device discovery

Off


Settings

Update & security

Find a setting

Use developer features

These settings are intended for development use only.
Learn more.

Don't use developer features

Sideload apps

[✓ Developer mode]

http://windows.Microsoft.com

Where can I develop?

Windows 10

Requires Visual Studio 2015

Windows 8.1 & Windows Server 2012 R2

The Visual Studio designer does not function

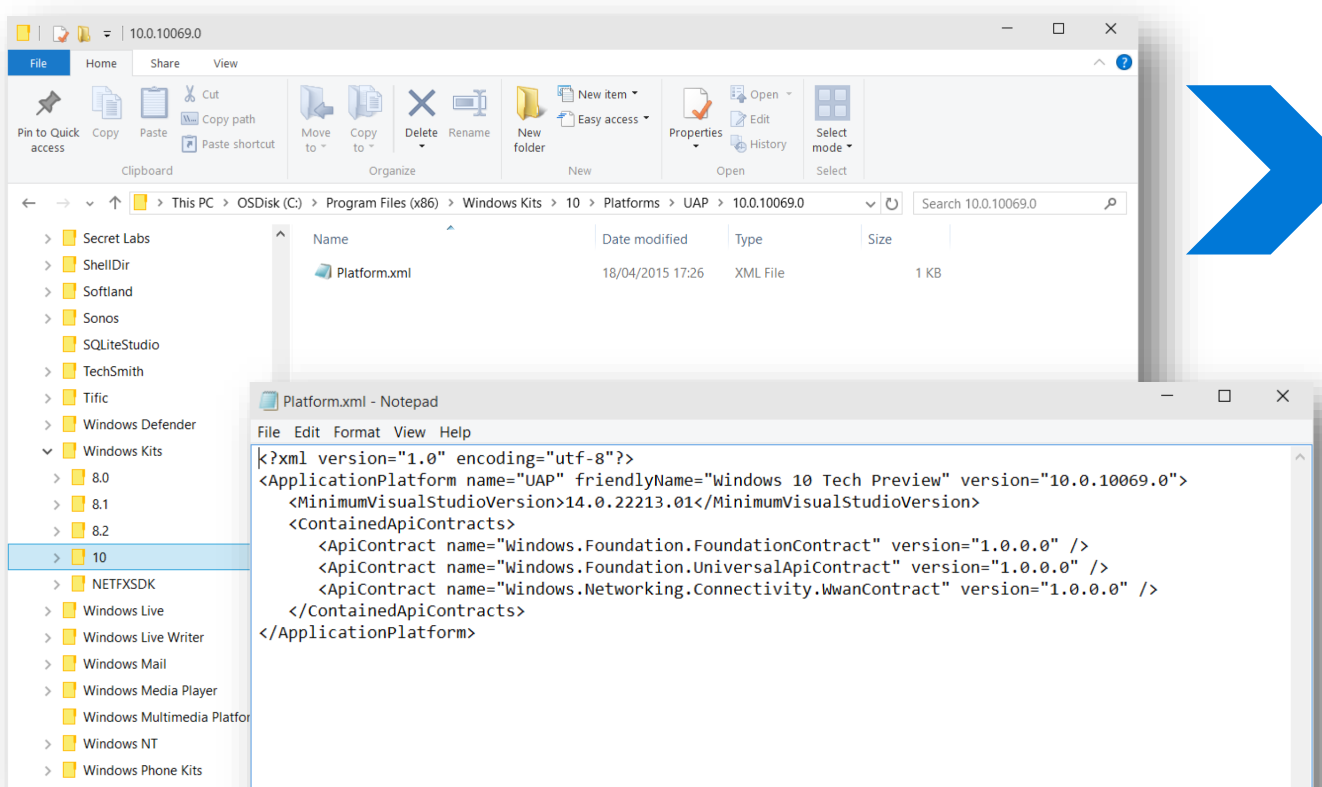
Debugging requires a Windows 10 device or Remote Debugging Tools

Universal Windows Platform

A single API surface

A guaranteed API surface

The same on all devices

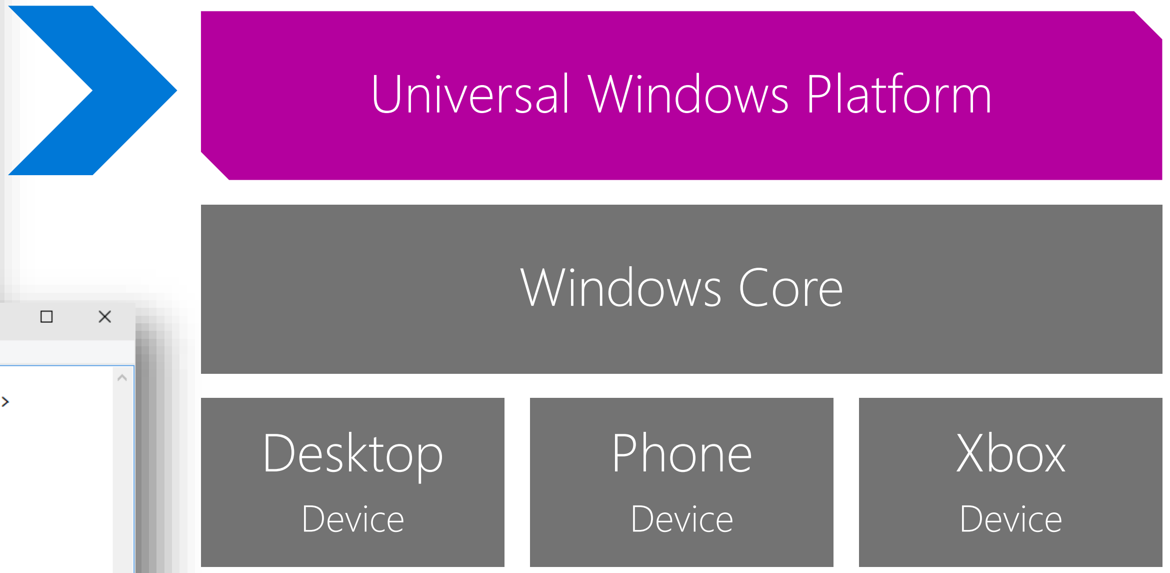


This PC > OSDisk (C:) > Program Files (x86) > Windows Kits > 10 > Platforms > UAP > 10.0.10069.0

```
Name            Date modified       Type        Size
Platform.xml    18/04/2015 17:26    XML File    1 KB
```

Platform.xml - Notepad

```
<?xml version="1.0" encoding="utf-8"?>
<ApplicationPlatform name="UAP" friendlyName="Windows 10 Tech Preview" version="10.0.10069.0">
   <MinimumVisualStudioVersion>14.0.22213.01</MinimumVisualStudioVersion>
   <ContainedApiContracts>
      <ApiContract name="Windows.Foundation.FoundationContract" version="1.0.0.0" />
      <ApiContract name="Windows.Foundation.UniversalApiContract" version="1.0.0.0" />
      <ApiContract name="Windows.Networking.Connectivity.WwanContract" version="1.0.0.0" />
   </ContainedApiContracts>
</ApplicationPlatform>
```



Apps don't target Windows 10, apps target the platform

Windows

```xml
<TargetPlatform
    Name="Microsoft.Universal"
    minVersion="10.0.10069.0"
    maxVersionTested="10.0.10190.0"/>
```

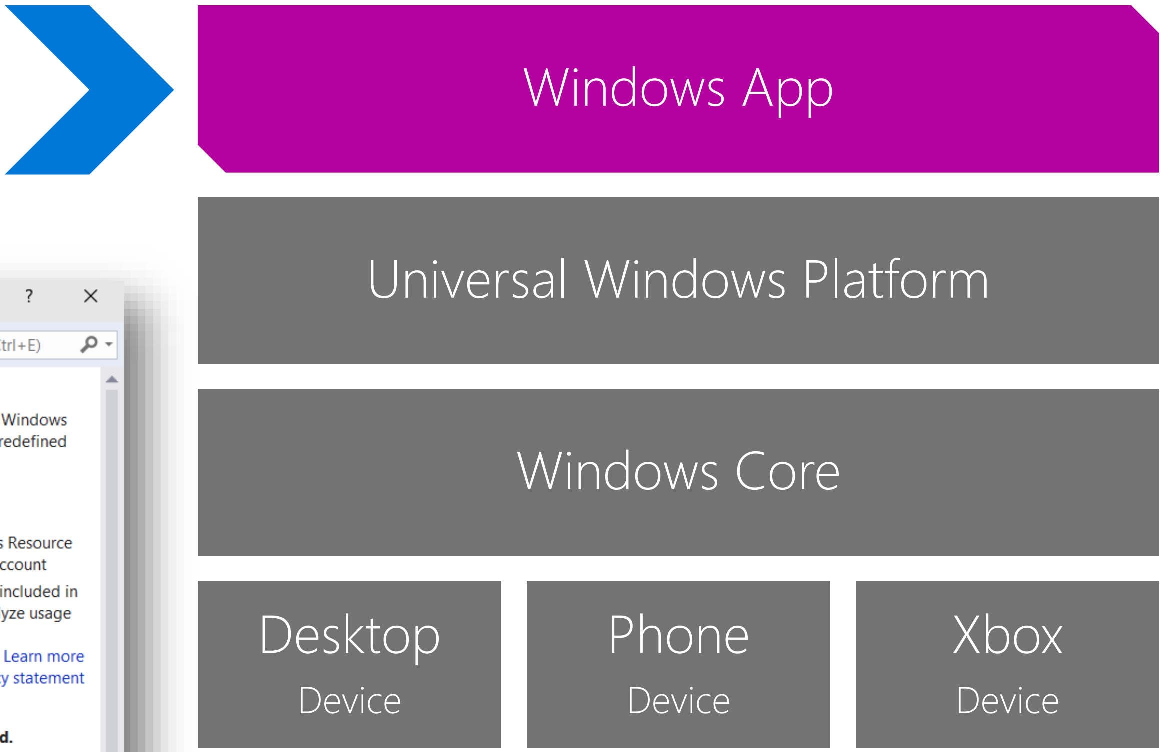
The Universal Windows Platform can update at its own cadence

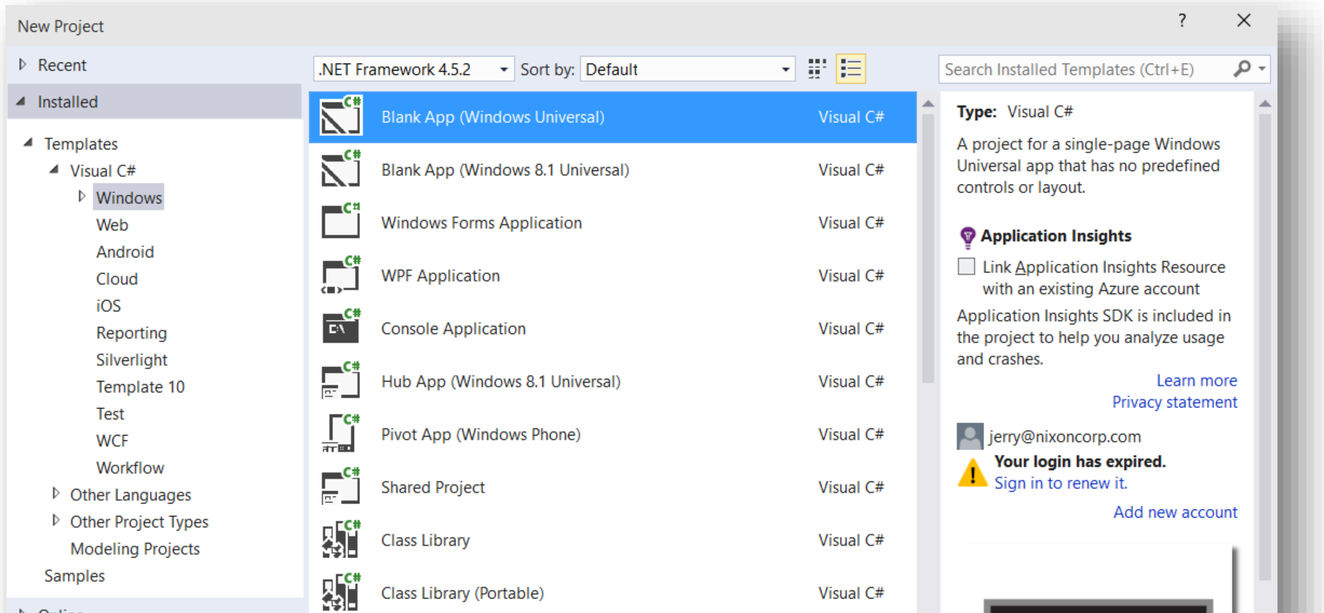
Windows

Windows app

A single binary

Running on any device
Testing for capabilities
Adjusting to devices




New Project

- Recent
- Installed
  - Templates
    - Visual C#
      - Windows
      - Web
      - Android
      - Cloud
      - iOS
      - Reporting
      - Silverlight
      - Template 10
      - Test
      - WCF
      - Workflow
    - Other Languages
    - Other Project Types
    - Modeling Projects
  - Samples

.NET Framework 4.5.2 | Sort by: Default | Search Installed Templates (Ctrl+E)

Blank App (Windows Universal)	Visual C#
Blank App (Windows 8.1 Universal)	Visual C#
Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
Hub App (Windows 8.1 Universal)	Visual C#
Pivot App (Windows Phone)	Visual C#
Shared Project	Visual C#
Class Library	Visual C#
Class Library (Portable)	Visual C#

Type: Visual C#

A project for a single-page Windows Universal app that has no predefined controls or layout.

Application Insights

☐ Link Application Insights Resource with an existing Azure account

Application Insights SDK is included in the project to help you analyze usage and crashes.

Learn more
Privacy statement

jerry@nixoncorp.com
Your login has expired.
Sign in to renew it.

Add new account

http://windows.Microsoft.com

Hello UWP

DEMO

Windows

Adaptive Code

Windows

What are Adaptive Apps?

Windows apps adapt to different versions of the platform

Windows apps adapt to different types of devices

Windows apps adapt to different screen sizes

Adaptive UI handles different screens

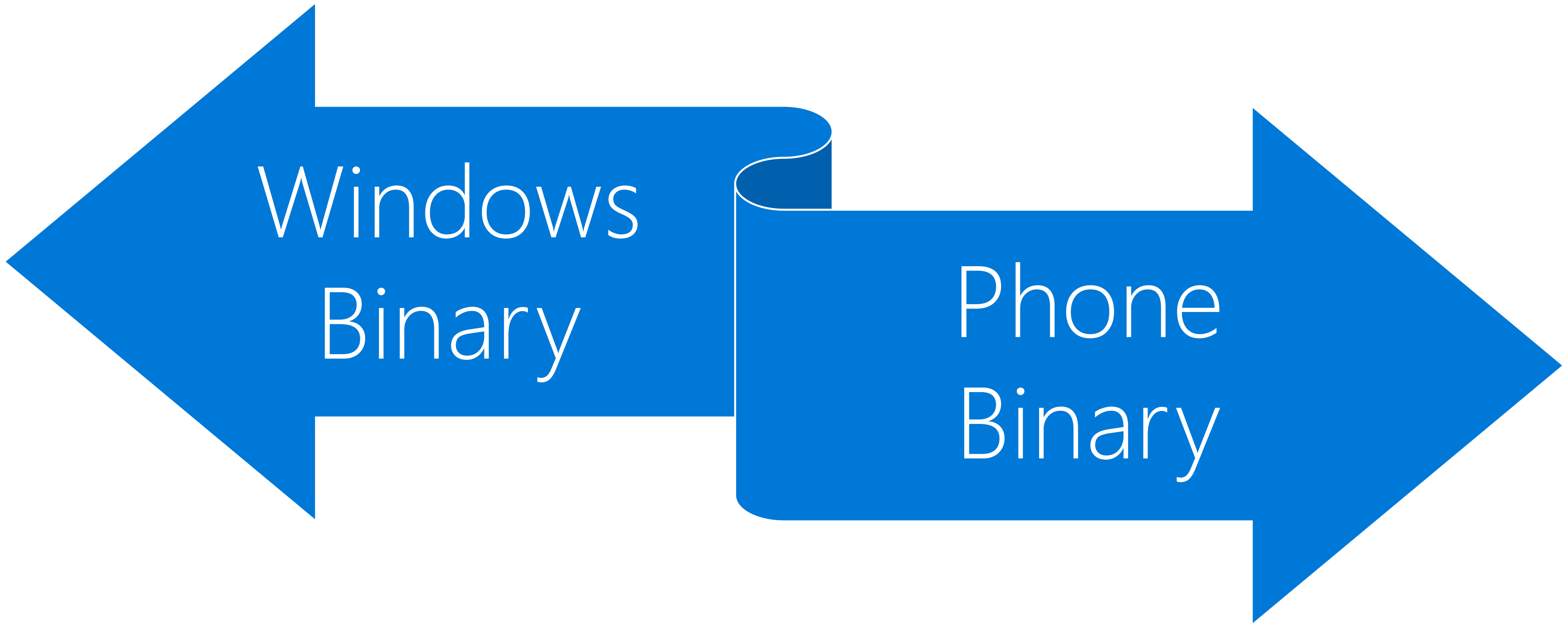
Adaptive Code can "light up" your app to conditionally execute code only when running on specific device families and/or particular versions of platform/extension APIs

- Conditionally take advantage of unique device capabilities
- Use newer APIs while still supporting down-level clients

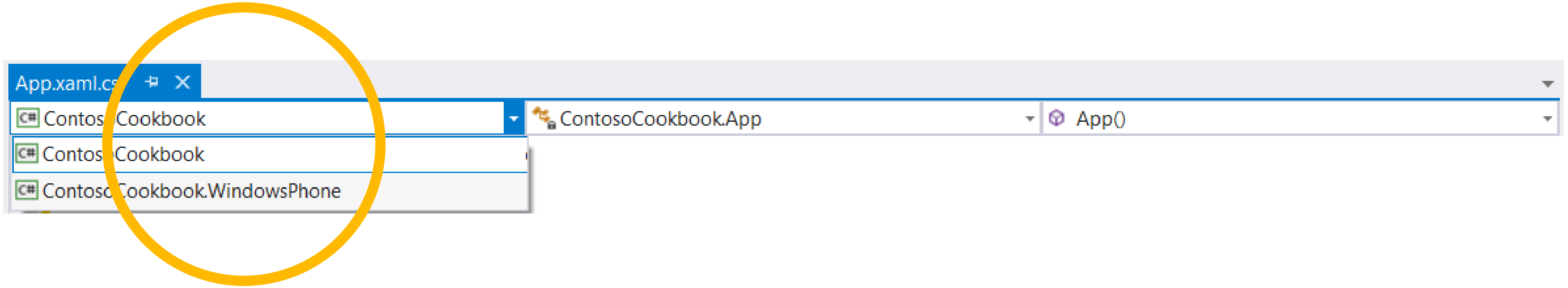
Looking back to Windows 8.1

Windows

Windows 8.1 Universal:
Shared code, two binaries



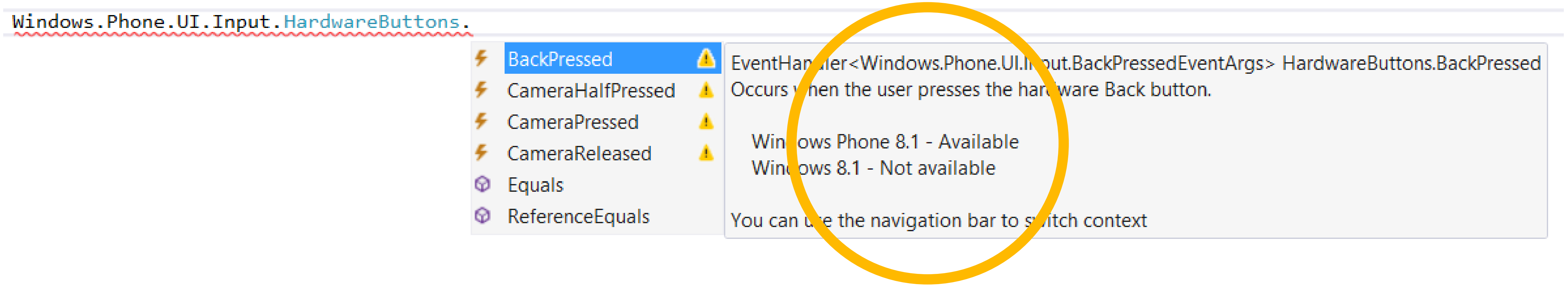
Not all APIs were available everywhere



App.xaml.cs

ContosoCookbook | ContosoCookbook.App | App()

ContosoCookbook
ContosoCookbook.WindowsPhone



```
Windows.Phone.UI.Input.HardwareButtons.
```

BackPressed
CameraHalfPressed
CameraPressed
CameraReleased
Equals
ReferenceEquals

EventHandler<Windows.Phone.UI.Input.BackPressedEventArgs> HardwareButtons.BackPressed
Occurs when the user presses the hardware Back button.

Windows Phone 8.1 - Available
Windows 8.1 - Not available

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
```

Looking Forward to UWP

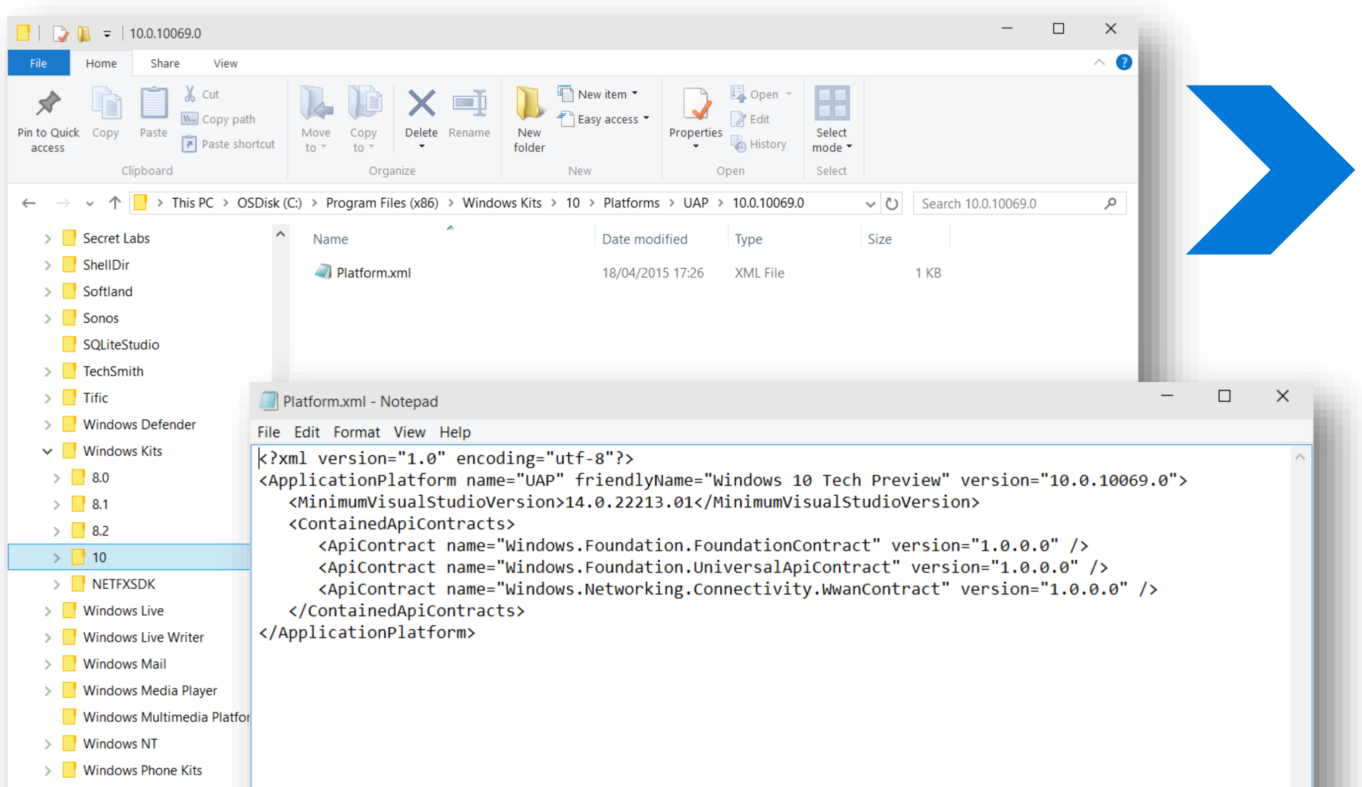
Windows

Universal Windows Platform

A single API surface

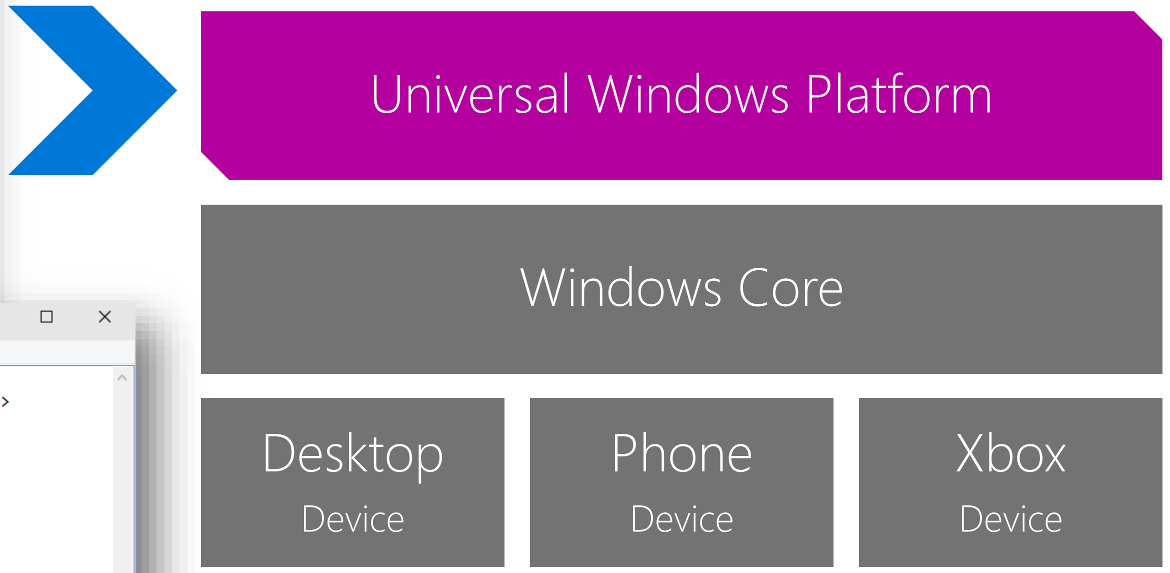
A guaranteed API surface

The same on all devices



Name	Date modified	Type	Size
Platform.xml	18/04/2015 17:26	XML File	1 KB

```
<?xml version="1.0" encoding="utf-8"?>
<ApplicationPlatform name="UAP" friendlyName="Windows 10 Tech Preview" version="10.0.10069.0">
   <MinimumVisualStudioVersion>14.0.22213.01</MinimumVisualStudioVersion>
   <ContainedApiContracts>
      <ApiContract name="Windows.Foundation.FoundationContract" version="1.0.0.0" />
      <ApiContract name="Windows.Foundation.UniversalApiContract" version="1.0.0.0" />
      <ApiContract name="Windows.Networking.Connectivity.WwanContract" version="1.0.0.0" />
   </ContainedApiContracts>
</ApplicationPlatform>
```



http://windows.Microsoft.com

Declare Device Family Dependencies

Dependency on a single device family:

```
<Dependencies>
    <TargetDeviceFamily Name="Windows.Universal"
                minVersion="10.0.10069.0" maxVersionTested="10.5.0.0" />
</Dependencies>
```

On more than one:

```
<Dependencies>
    <TargetDeviceFamily Name="Windows.Desktop"
                minVersion="10.0.10069.0" maxVersionTested="10.5.0.0" />
    <TargetDeviceFamily Name="Windows.Universal"
                minVersion="10.0.10069.0" maxVersionTested="10.5.0.0" />
</Dependencies>
```

Introducing Platform Extension SDKs

Windows

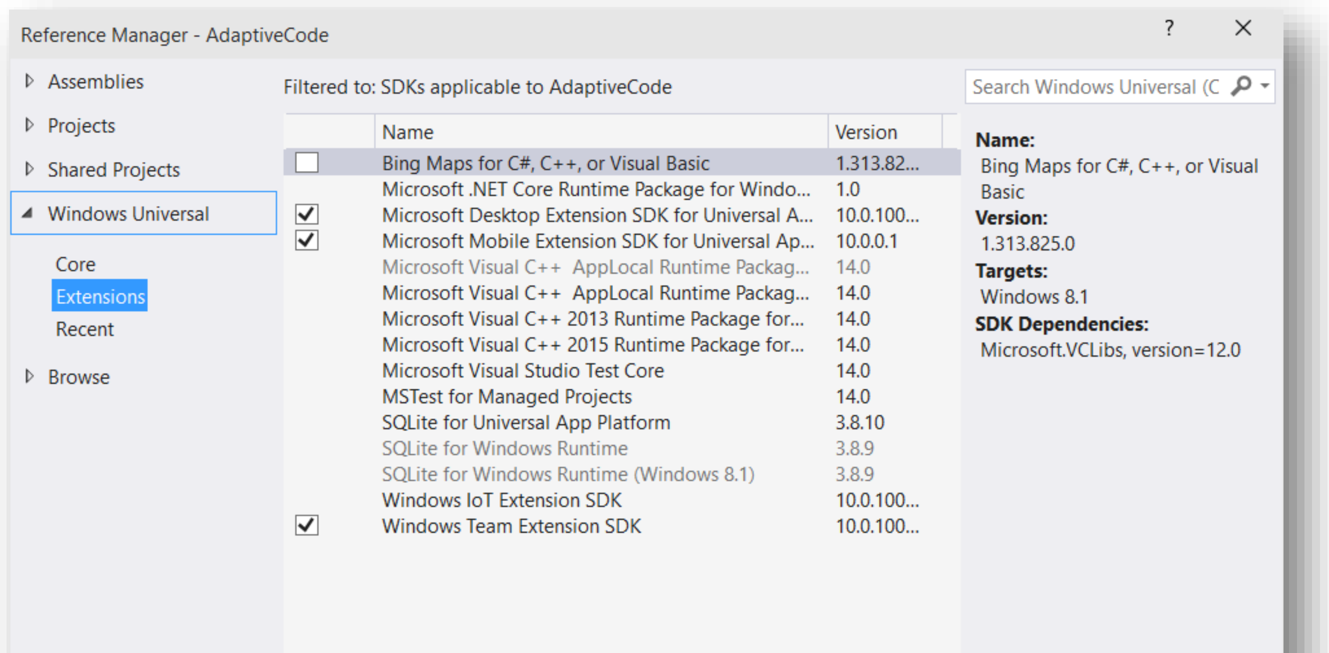
Platform extensions

Device-specific API

Family-specific capabilities

Compatible across devices

Unique update cadence



Reference Manager - AdaptiveCode

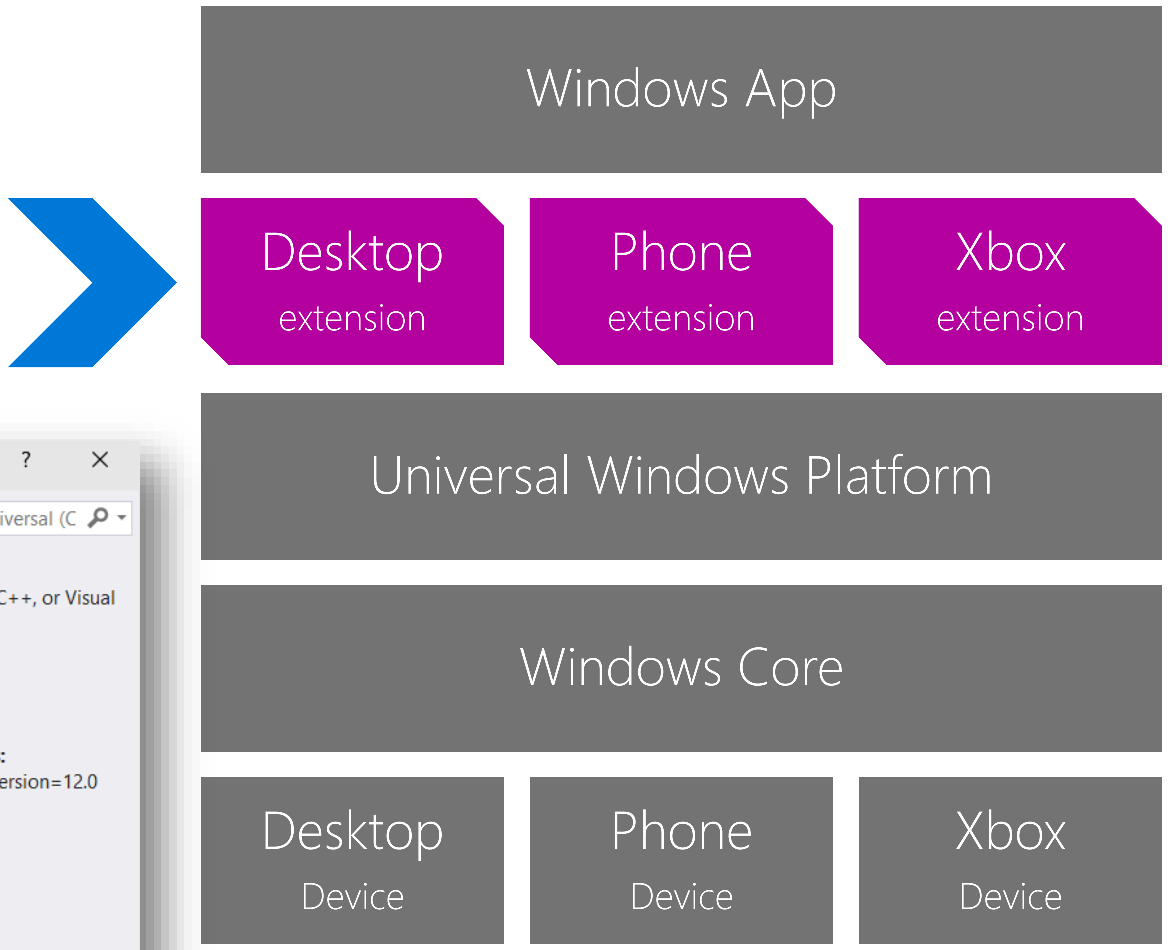
- Assemblies
- Projects
- Shared Projects
- Windows Universal
  - Core
  - Extensions
  - Recent
- Browse

Filtered to: SDKs applicable to AdaptiveCode

Search Windows Universal (C

	Name	Version
☐	Bing Maps for C#, C++, or Visual Basic	1.313.82...
	Microsoft .NET Core Runtime Package for Windo...	1.0
✓	Microsoft Desktop Extension SDK for Universal A...	10.0.100...
✓	Microsoft Mobile Extension SDK for Universal Ap...	10.0.0.1
	Microsoft Visual C++ AppLocal Runtime Packag...	14.0
	Microsoft Visual C++ AppLocal Runtime Packag...	14.0
	Microsoft Visual C++ 2013 Runtime Package for...	14.0
	Microsoft Visual C++ 2015 Runtime Package for...	14.0
	Microsoft Visual Studio Test Core	14.0
	MSTest for Managed Projects	14.0
	SQLite for Universal App Platform	3.8.10
	SQLite for Windows Runtime	3.8.9
	SQLite for Windows Runtime (Windows 8.1)	3.8.9
	Windows IoT Extension SDK	10.0.100...
✓	Windows Team Extension SDK	10.0.100...

Name: Bing Maps for C#, C++, or Visual Basic
Version: 1.313.825.0
Targets: Windows 8.1
SDK Dependencies: Microsoft.VCLibs, version=12.0



Extensions don't invalidate binaries on other devices

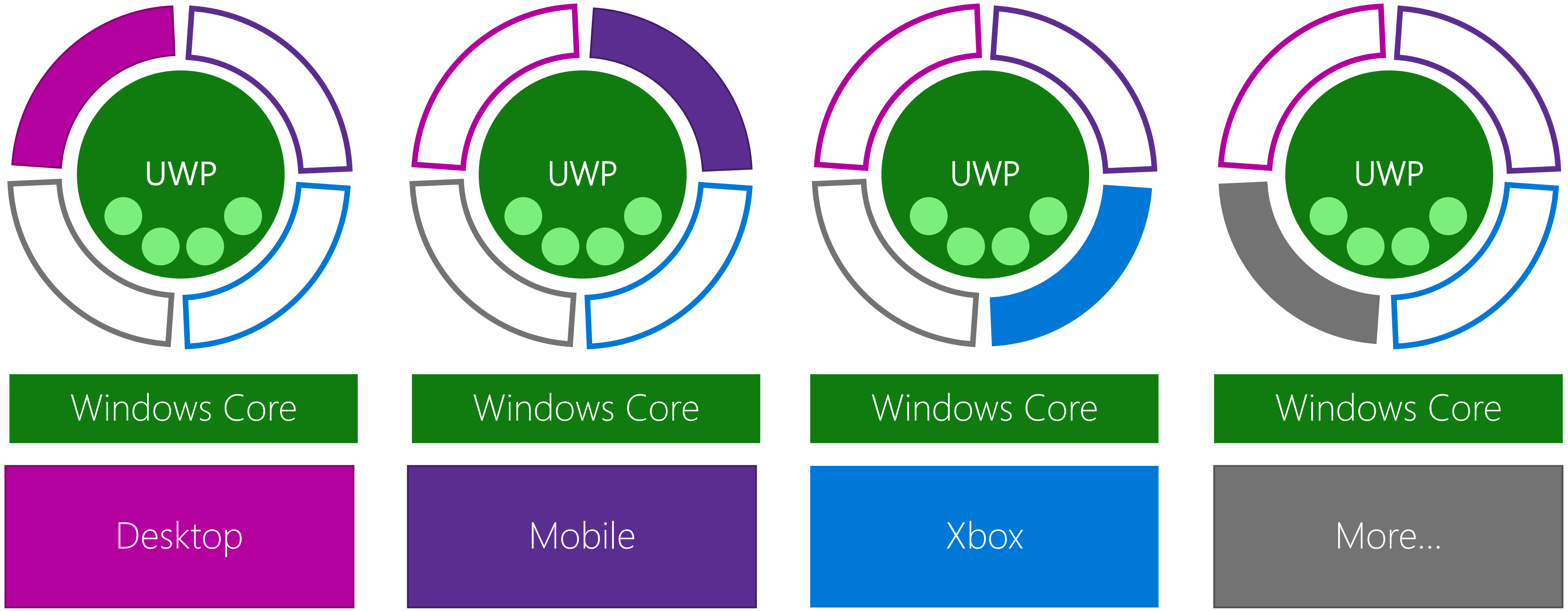
Windows

Extensions SDKs in VS 15

DEMO

Windows

Extension SDKs



Testing for capabilities

```
Windows.Foundation.Metadata.ApiInformation

        IsApiContractPresent
        IsEnumNamedValuePresent
        IsEventPresent
        IsMethodPresent
        IsPropertyPresent
        IsReadOnlyPropertyPresent
        IsTypePresent
        IsWriteablePropertyPresent
```

Test capabilities at runtime

```
var api = "Windows.Phone.UI.Input.HardwareButtons";
if (Windows.Foundation.Metadata.ApiInformation.IsTypePresent(api))
{
   Windows.Phone.UI.Input.HardwareButtons.CameraPressed
        += CameraButtonPressed;
}
```

The ApiInformation API tests for capabilities at runtime.

Windows

Which Extension SDKs Do I Need?

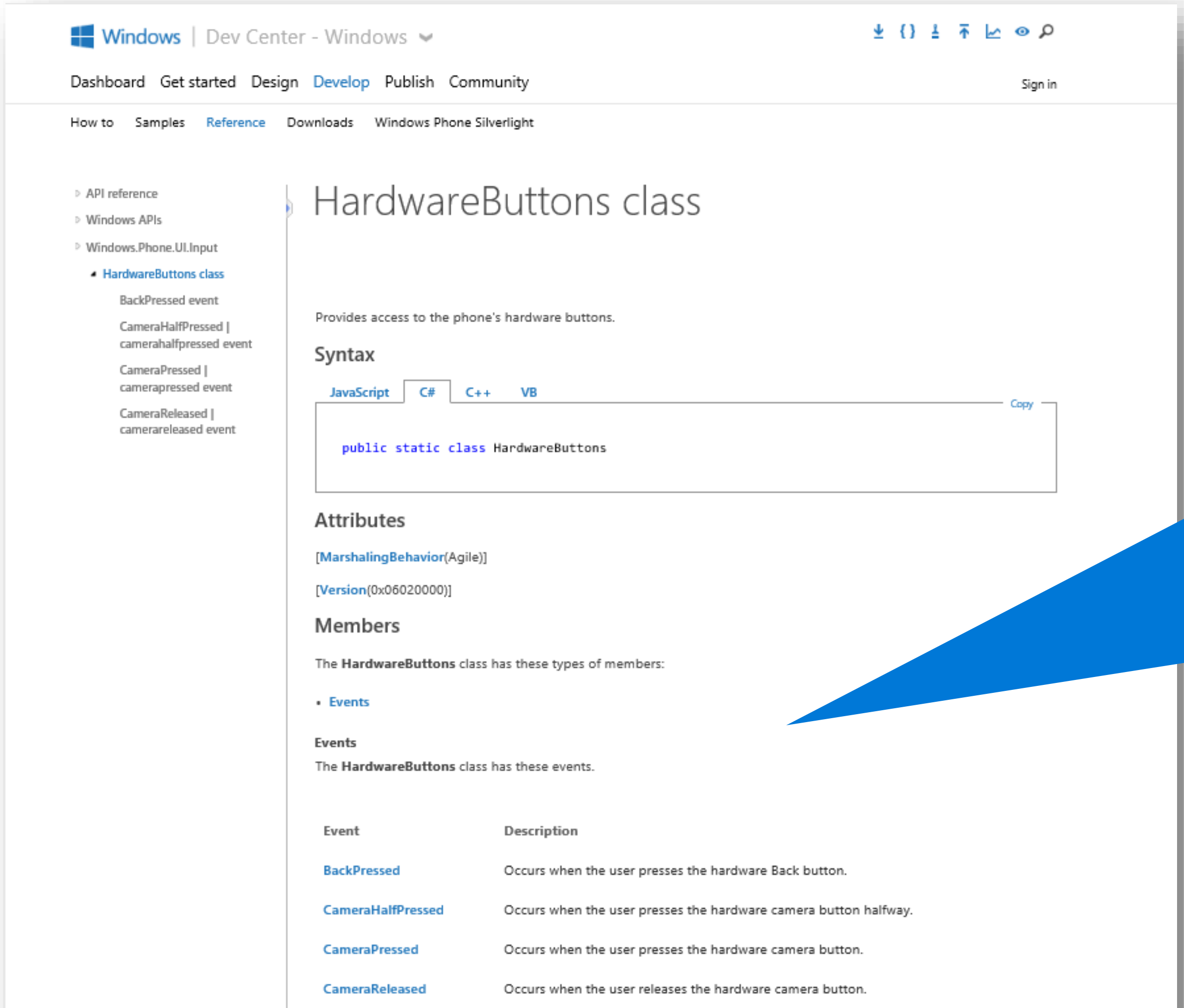
Many Apps need no Extension SDKs at all

The Windows Universal Core APIs cover nearly all common app needs

Use APIs in Extension SDKs to 'light up' your app when running on a specific device family

Identifying the Extension SDK

MSDN docs:



Windows | Dev Center - Windows

Dashboard Get started Design Develop Publish Community Sign in

How to Samples Reference Downloads Windows Phone Silverlight

- API reference
- Windows APIs
- Windows.Phone.UI.Input
  - HardwareButtons class
    - BackPressed event
    - CameraHalfPressed | camerahalfpressed event
    - CameraPressed | camerapressed event
    - CameraReleased | camerareleased event

HardwareButtons class

Provides access to the phone's hardware buttons.

Syntax

JavaScript C# C++ VB Copy

```
public static class HardwareButtons
```

Attributes

[MarshalingBehavior(Agile)]

[Version(0x06020000)]

Members

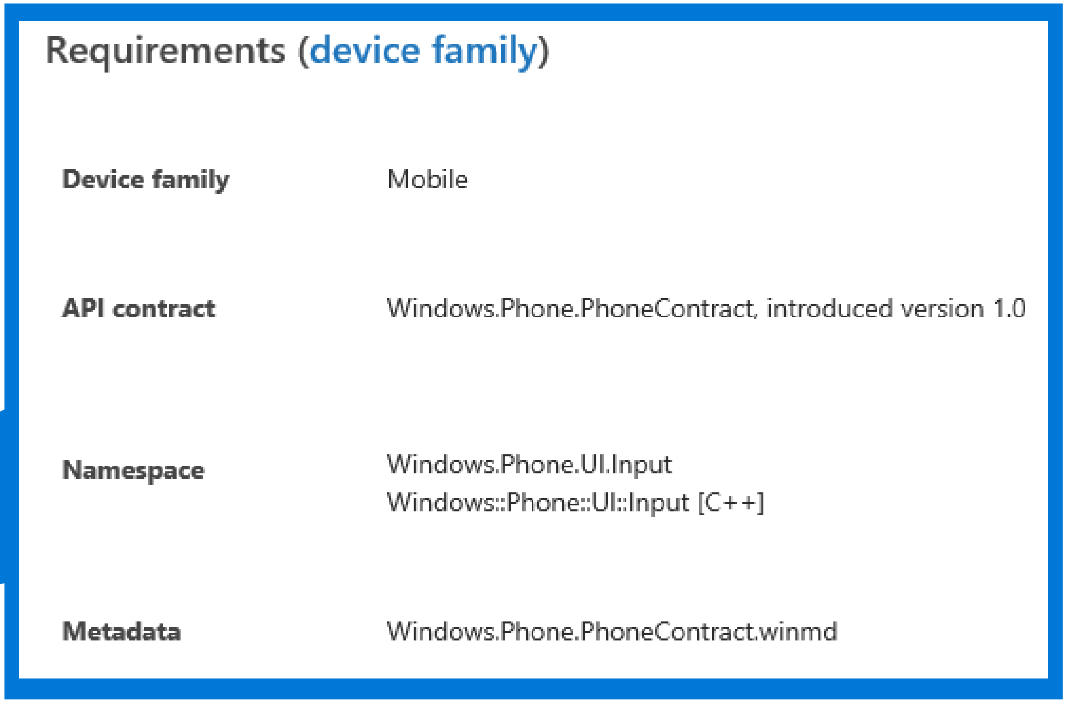
The **HardwareButtons** class has these types of members:

- Events

Events

The **HardwareButtons** class has these events.

Event	Description
BackPressed	Occurs when the user presses the hardware Back button.
CameraHalfPressed	Occurs when the user presses the hardware camera button halfway.
CameraPressed	Occurs when the user presses the hardware camera button.
CameraReleased	Occurs when the user releases the hardware camera button.

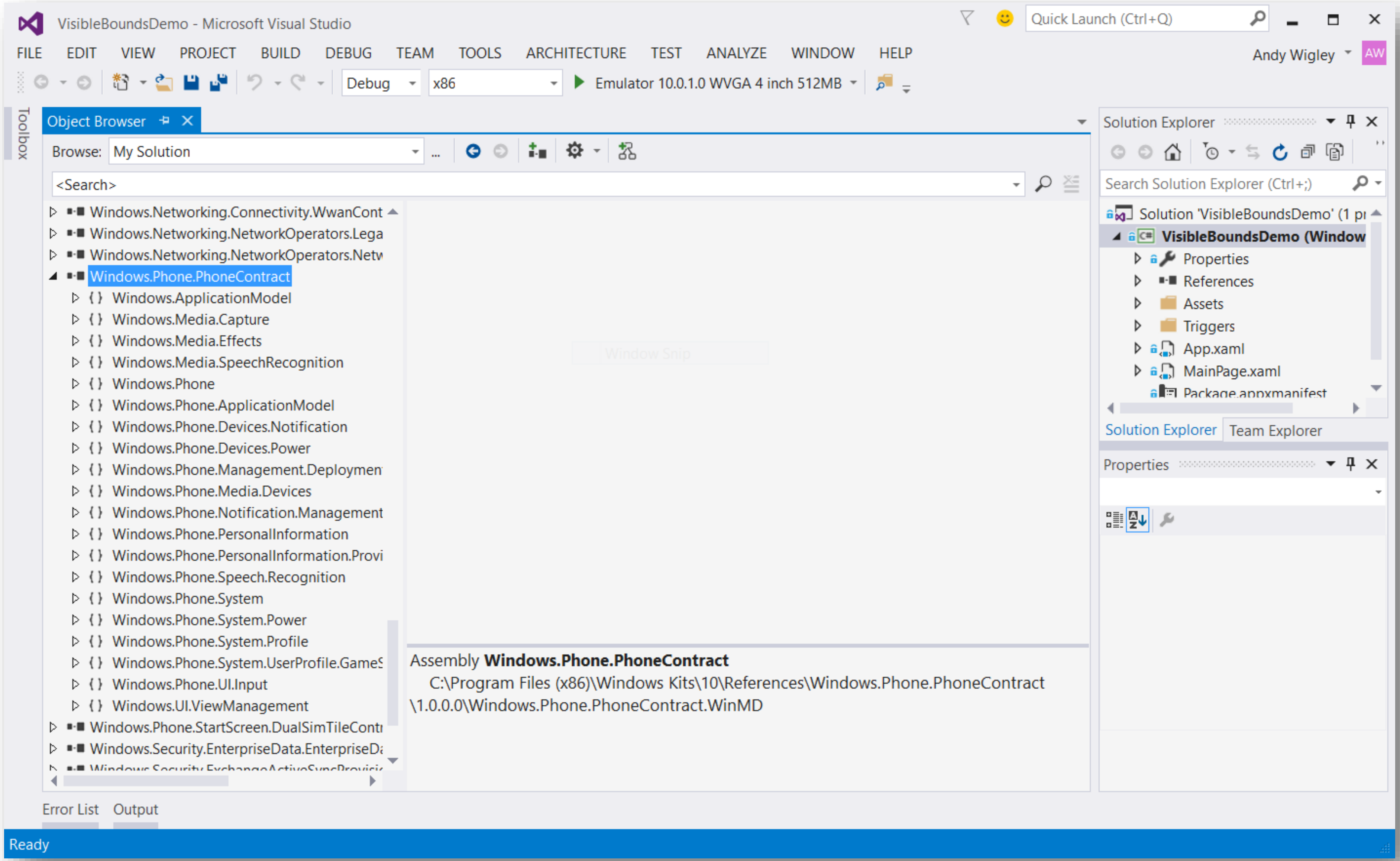


Requirements (device family)

Device family	Mobile
API contract	Windows.Phone.PhoneContract, introduced version 1.0
Namespace	Windows.Phone.UI.Input Windows::Phone::UI::Input [C++]
Metadata	Windows.Phone.PhoneContract.winmd

http://windows.Microsoft.com

Exploring API Contracts


VisibleBoundsDemo - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS ARCHITECTURE TEST ANALYZE WINDOW HELP
Andy Wigley
AW
Debug
x86
Emulator 10.0.1.0 WVGA 4 inch 512MB
Toolbox
Object Browser
Browse: My Solution
<Search>
Solution Explorer
Search Solution Explorer (Ctrl+;)
Properties
Solution Explorer
Team Explorer
Error List
Output
Ready

- Windows.Networking.Connectivity.WwanCont
- Windows.Networking.NetworkOperators.Lega
- Windows.Networking.NetworkOperators.Netw
- Windows.Phone.PhoneContract
  - Windows.ApplicationModel
  - Windows.Media.Capture
  - Windows.Media.Effects
  - Windows.Media.SpeechRecognition
  - Windows.Phone
  - Windows.Phone.ApplicationModel
  - Windows.Phone.Devices.Notification
  - Windows.Phone.Devices.Power
  - Windows.Phone.Management.Deploymen
  - Windows.Phone.Media.Devices
  - Windows.Phone.Notification.Management
  - Windows.Phone.PersonalInformation
  - Windows.Phone.PersonalInformation.Provi
  - Windows.Phone.Speech.Recognition
  - Windows.Phone.System
  - Windows.Phone.System.Power
  - Windows.Phone.System.Profile
  - Windows.Phone.System.UserProfile.GameS
  - Windows.Phone.UI.Input
  - Windows.UI.ViewManagement
- Windows.Phone.StartScreen.DualSimTileCont
- Windows.Security.EnterpriseData.EnterpriseD

Assembly **Windows.Phone.PhoneContract**
C:\Program Files (x86)\Windows Kits\10\References\Windows.Phone.PhoneContract\1.0.0.0\Windows.Phone.PhoneContract.WinMD

Solution 'VisibleBoundsDemo' (1 pr
- **VisibleBoundsDemo (Window**
  - Properties
  - References
  - Assets
  - Triggers
  - App.xaml
  - MainPage.xaml
  - Package.appxmanifest

http://windows.Microsoft.com

Adaptive Code and API versions

Windows

Using Specific Versions of an API

Adaptive code techniques are not only for handling device family-specific code

You write your app against a base UWP version, but 6 months later, UWP v.Next ships to users machines

Applies to Extension SDKs and Packages as well – new versions may offer new functionality

You want to keep supporting customers who haven't updated yet, but take advantage of up-level APIs for those who have

Package Dependency

```
<Dependencies>
  <PackageDependency
       Name="Microsoft.WinJS 1.0"
       Publisher="CN=Microsoft Corporation, O=Microsoft Corporation,
L=Redmond, S=Washington, C=US"
       minVersion ="1.5.0.0" />
</Dependencies>
```

Gate use of up-level APIs

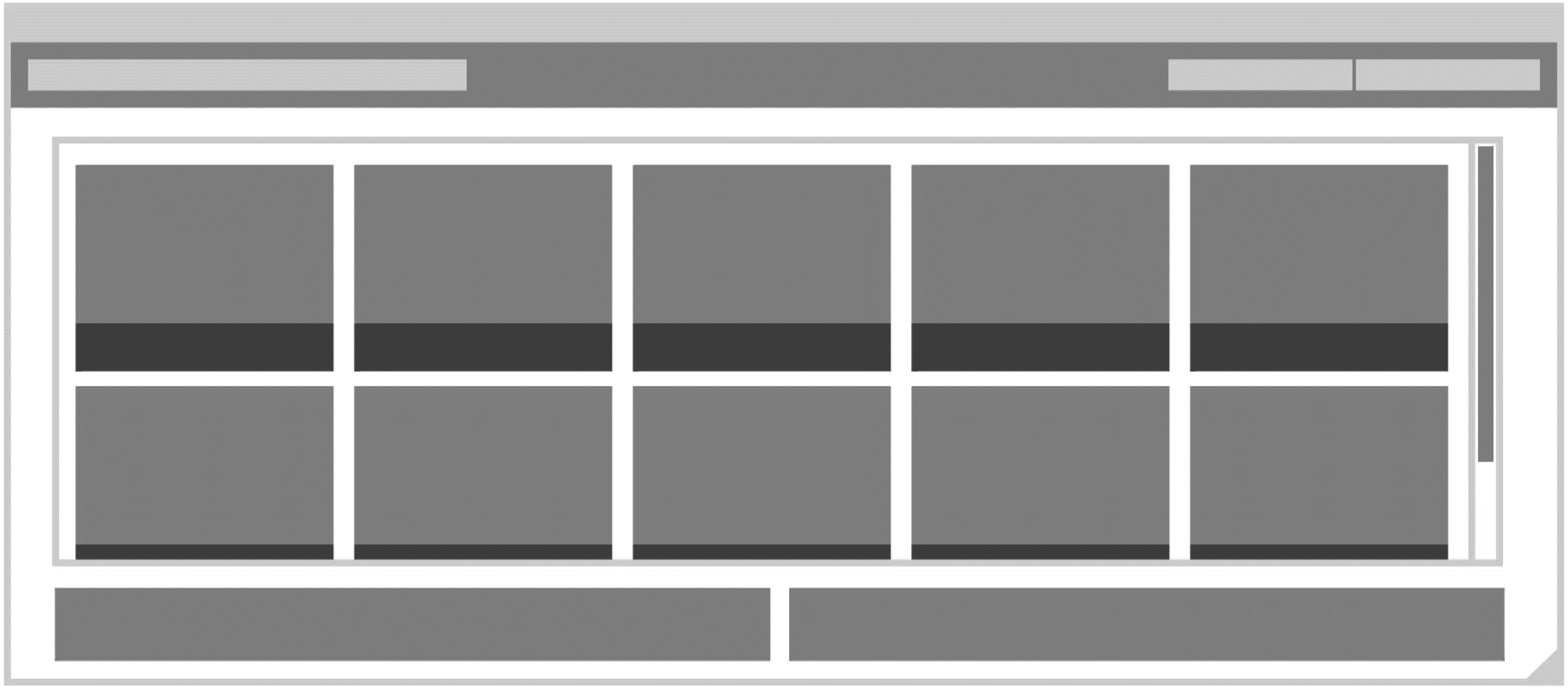
```
var contract = "Devices.Scanners.ScannerDeviceContract";
int majorVersionRequired = 3;

if (Windows.Foundation.Metadata.ApiInformation.
          IsApiContractPresent(contract, majorVersionRequired ))
{
   // Call the API that is present in V3 and above
   ...
}
else
{
  // Your original code supporting users who haven’t upgraded yet
  ...
}
```

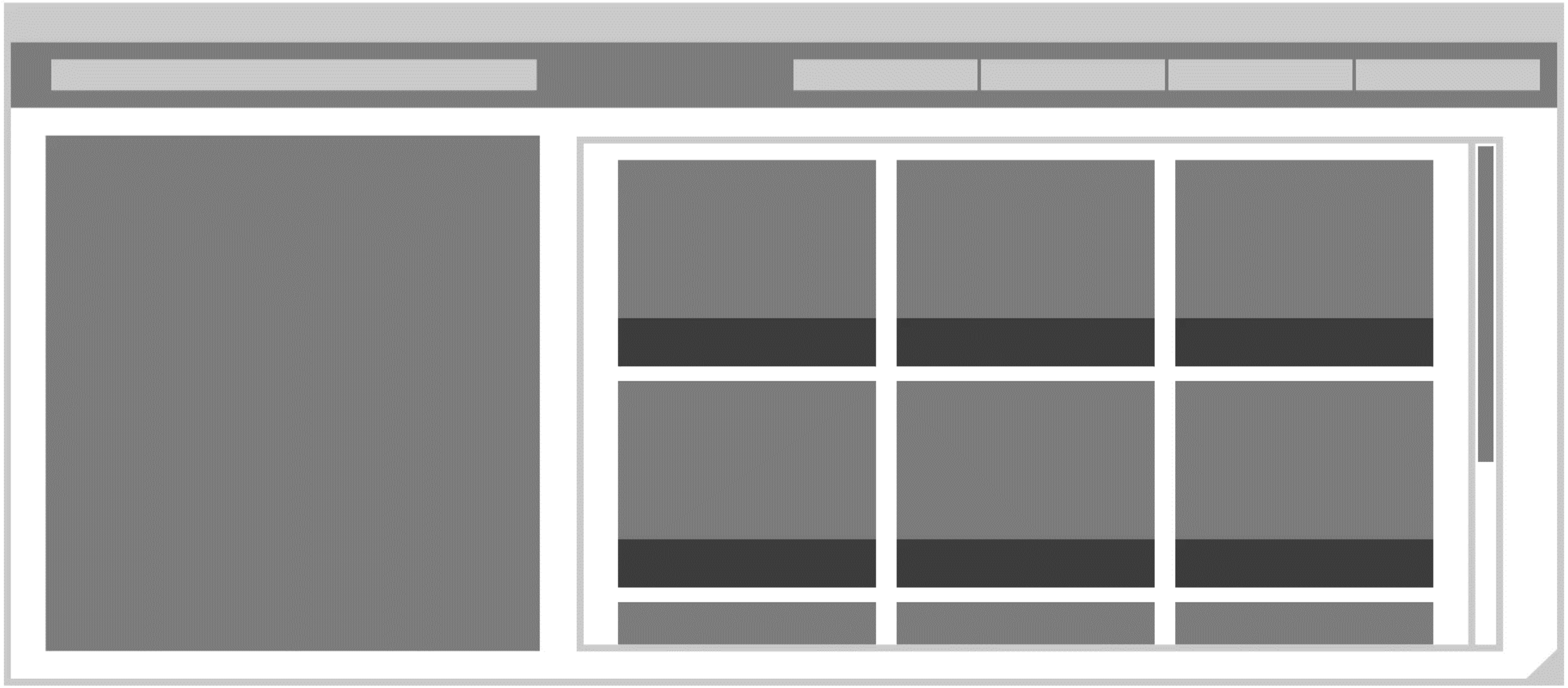
Adaptive design and UI

Windows

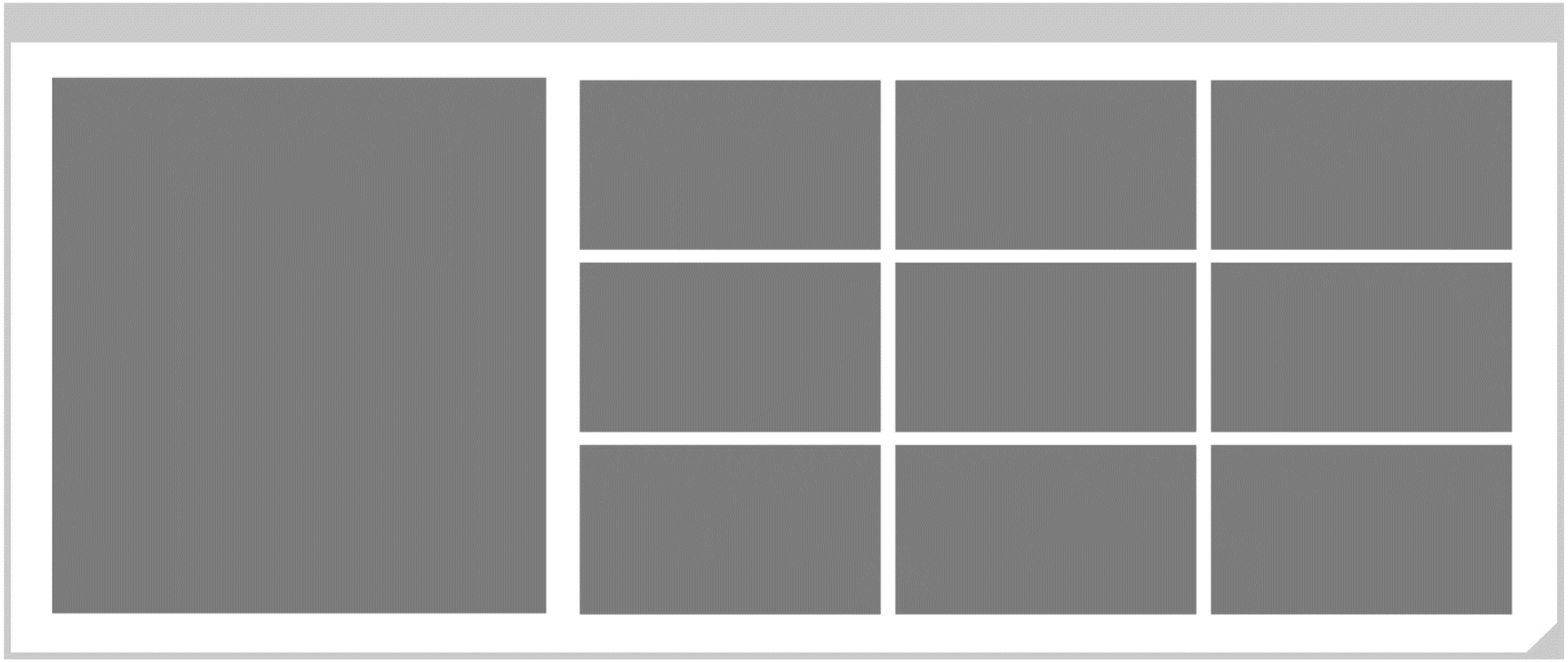
Responsive



Adaptive

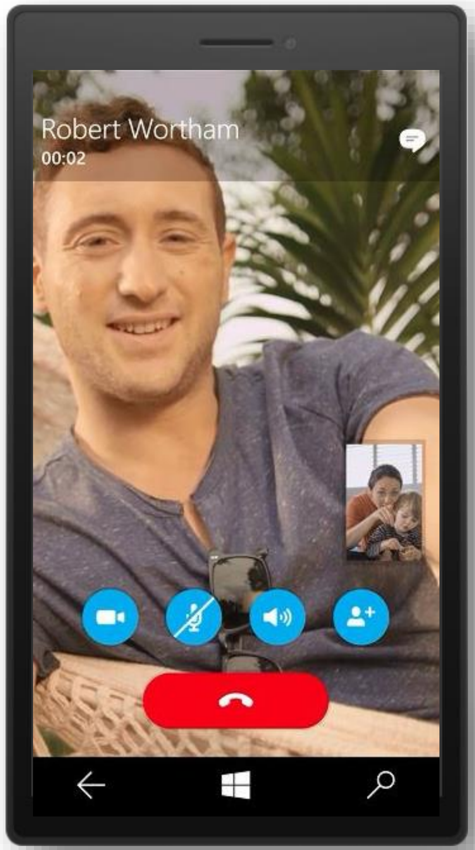


Scaling

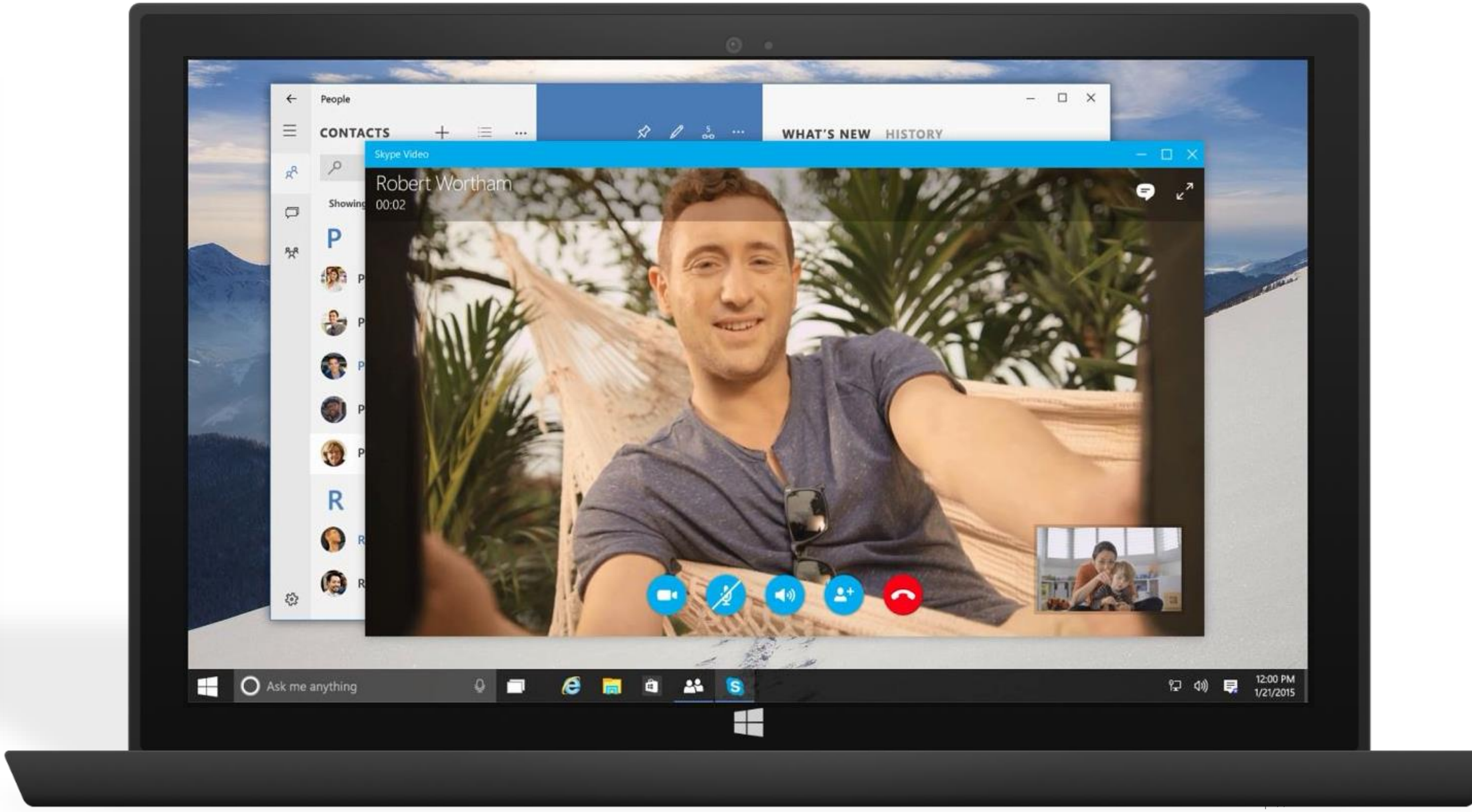


Adaptive design

Phone (portrait)

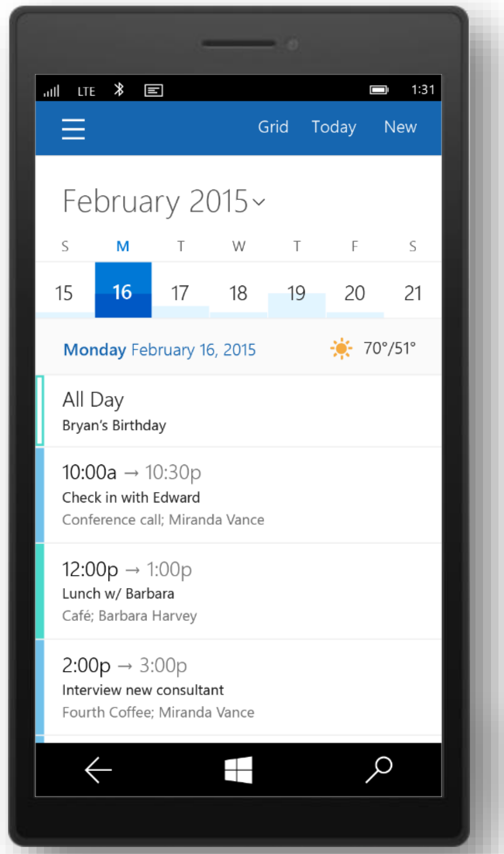


Tablet (landscape) / Desktop



Tailored design

Phone (portrait)



Monday February 16, 2015 70°/51°

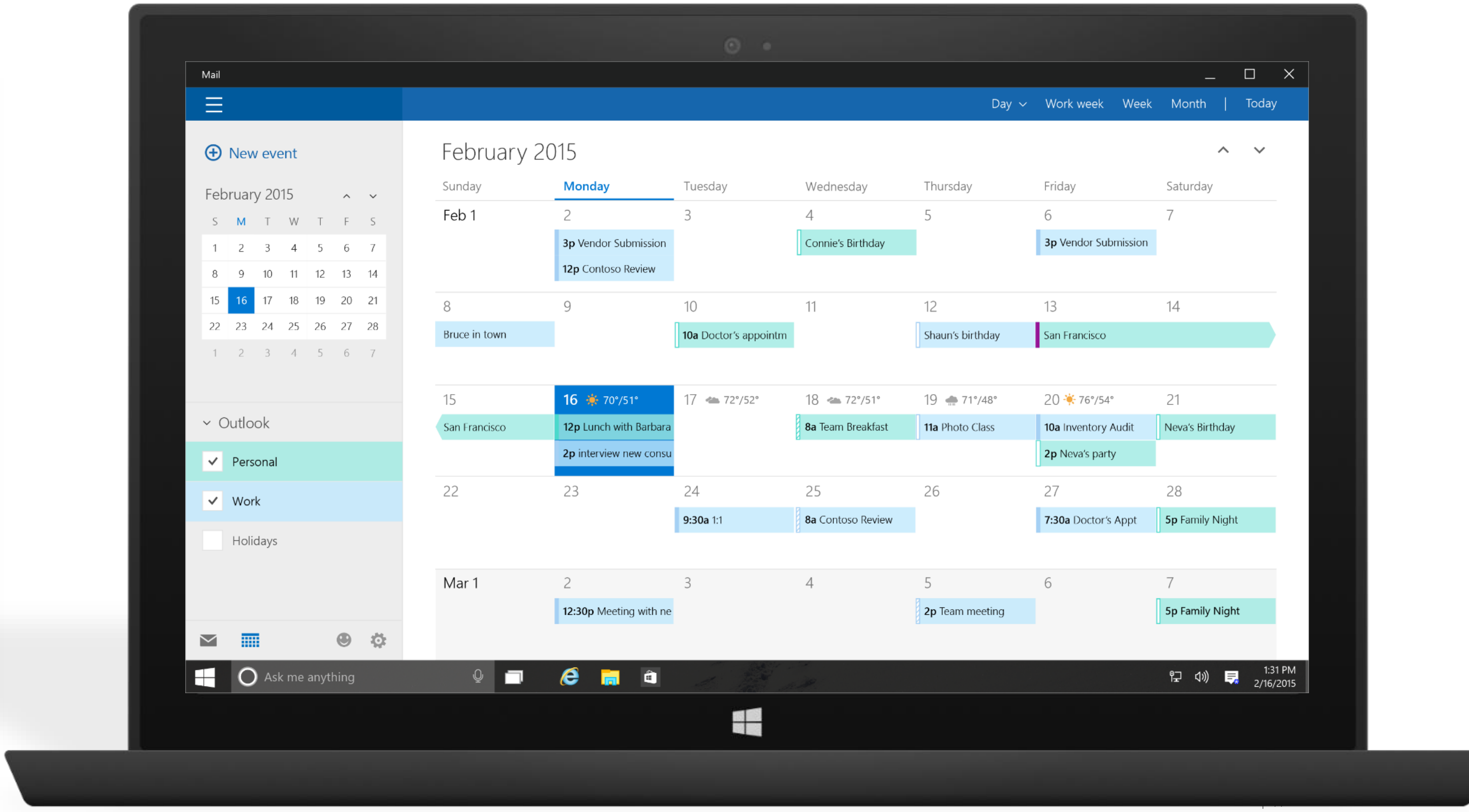
All Day
Bryan's Birthday

10:00a → 10:30p
Check in with Edward
Conference call; Miranda Vance

12:00p → 1:00p
Lunch w/ Barbara
Café; Barbara Harvey

2:00p → 3:00p
Interview new consultant
Fourth Coffee; Miranda Vance

Tablet (landscape) / Desktop

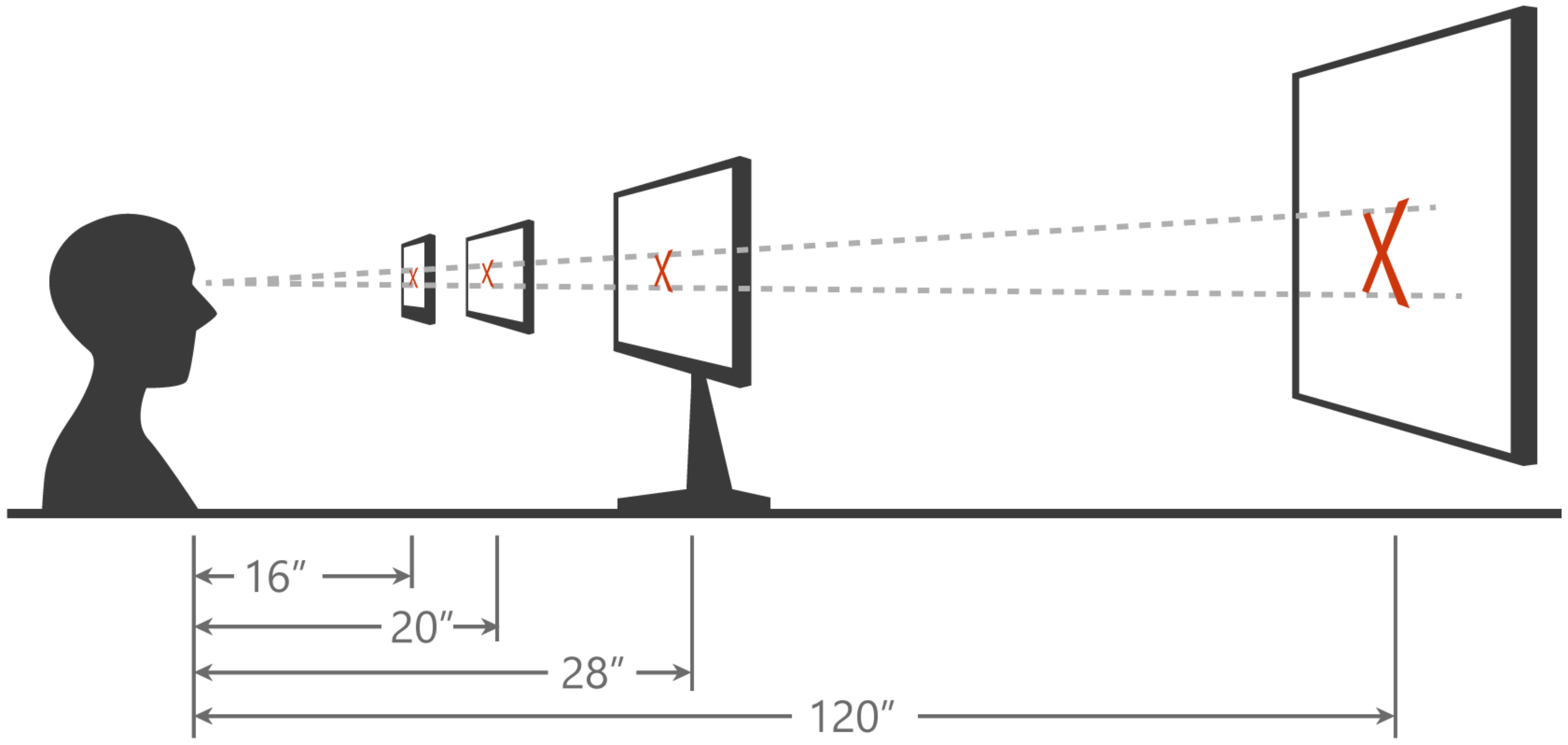


February 2015

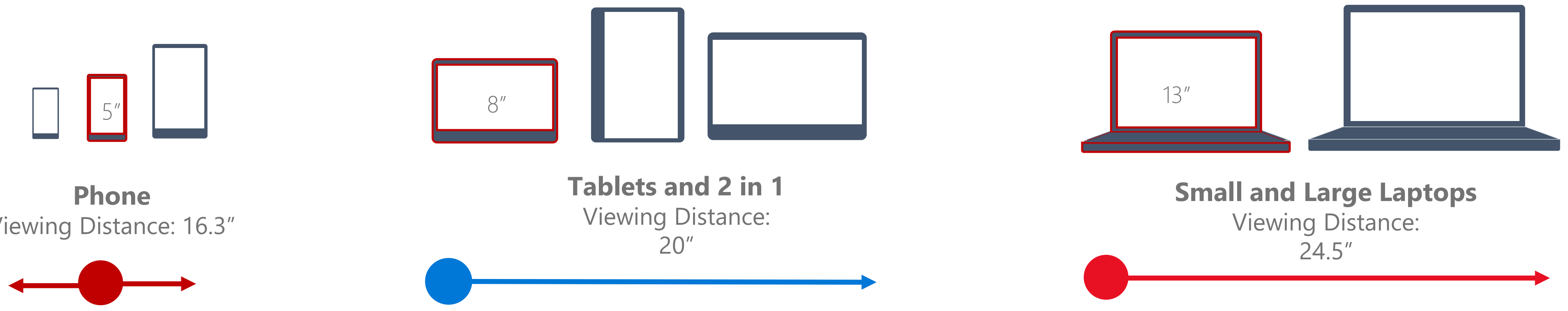
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Feb 1	2 3p Vendor Submission 12p Contoso Review	3	4 Connie's Birthday	5	6 3p Vendor Submission	7
8 Bruce in town	9	10 10a Doctor's appointm	11	12 Shaun's birthday	13 San Francisco	14
15 San Francisco	16 70°/51° 12p Lunch with Barbara 2p interview new consu	17 72°/52°	18 72°/51° 8a Team Breakfast	19 71°/48° 11a Photo Class	20 76°/54° 10a Inventory Audit 2p Neva's party	21 Neva's Birthday
22	23	24 9:30a 1:1	25 8a Contoso Review	26	27 7:30a Doctor's Appt	28 5p Family Night
Mar 1	2 12:30p Meeting with ne	3	4	5 2p Team meeting	6	7 5p Family Night

Continuum for convertibles and Phones

Scaling algorithm



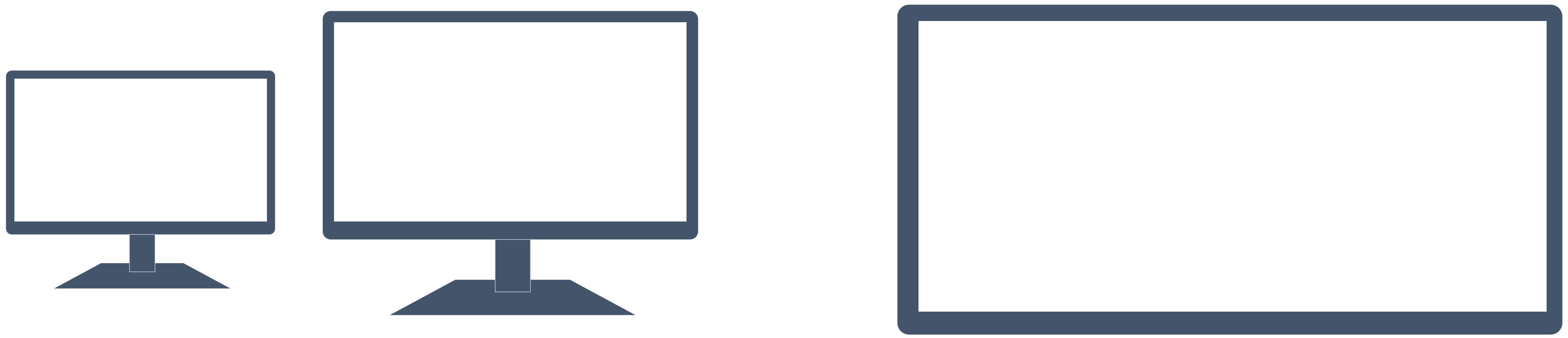
Planning your design



Phone
Viewing Distance: 16.3″

Tablets and 2 in 1
Viewing Distance:
20″

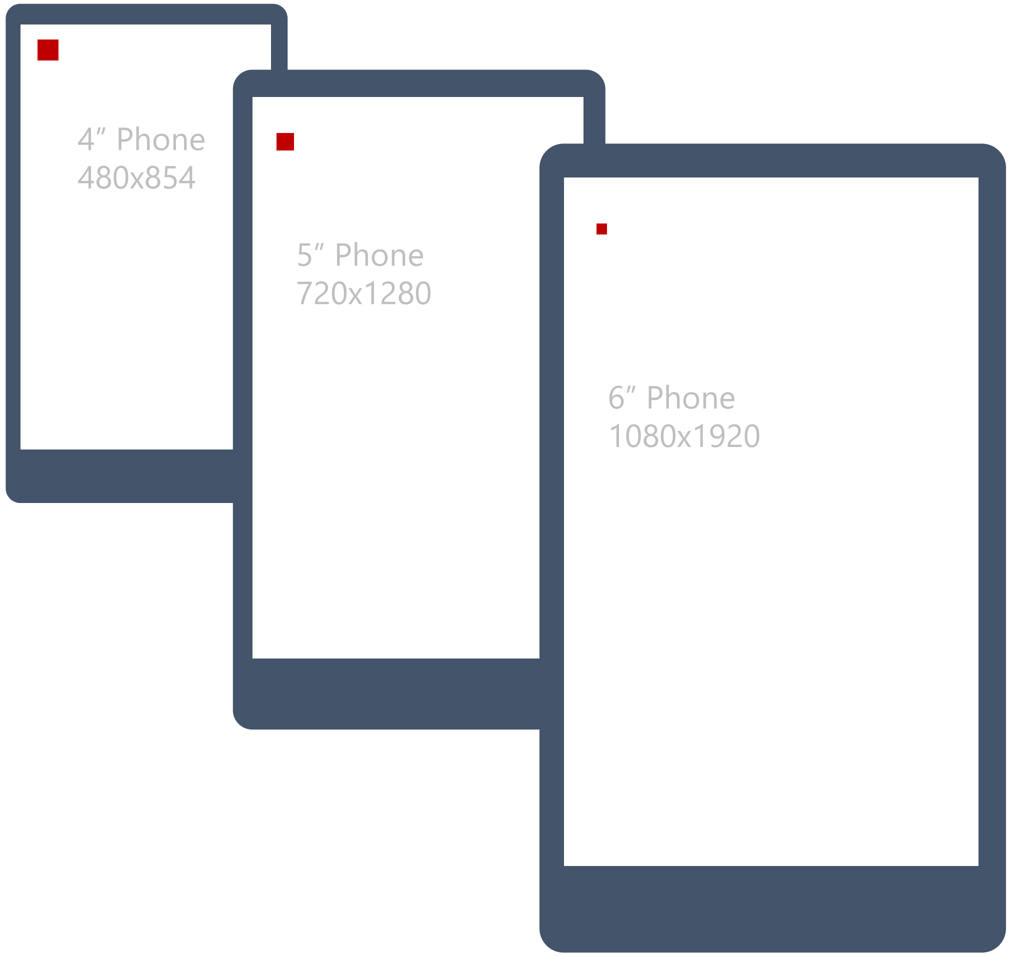
Small and Large Laptops
Viewing Distance:
24.5″

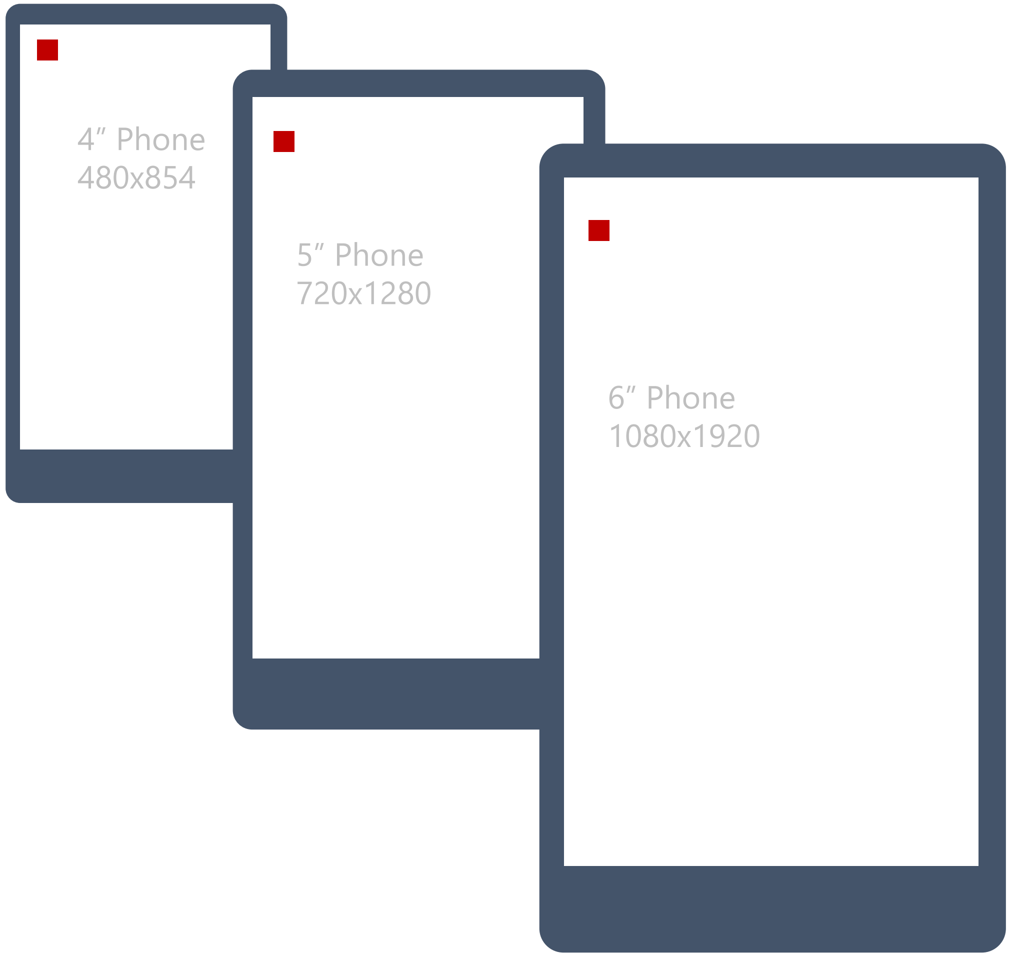


Small and Large Desktop Monitors
Viewing Distance:
28″

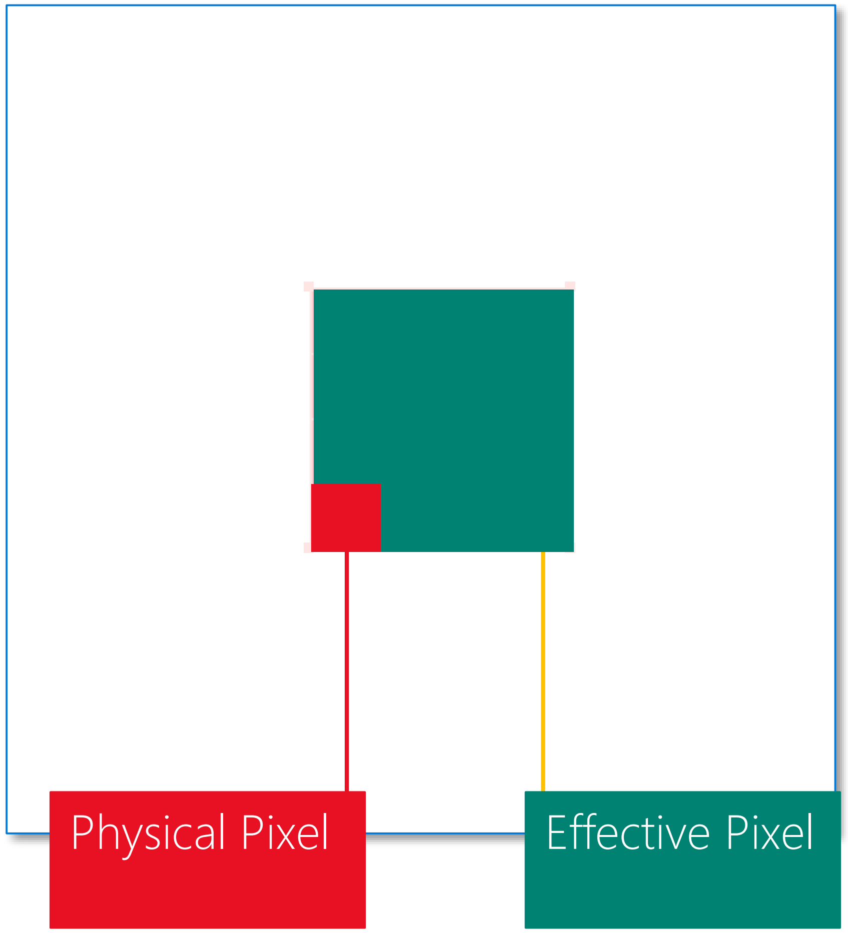
TV
Viewing Distance:
84″

Physical versus effective pixel





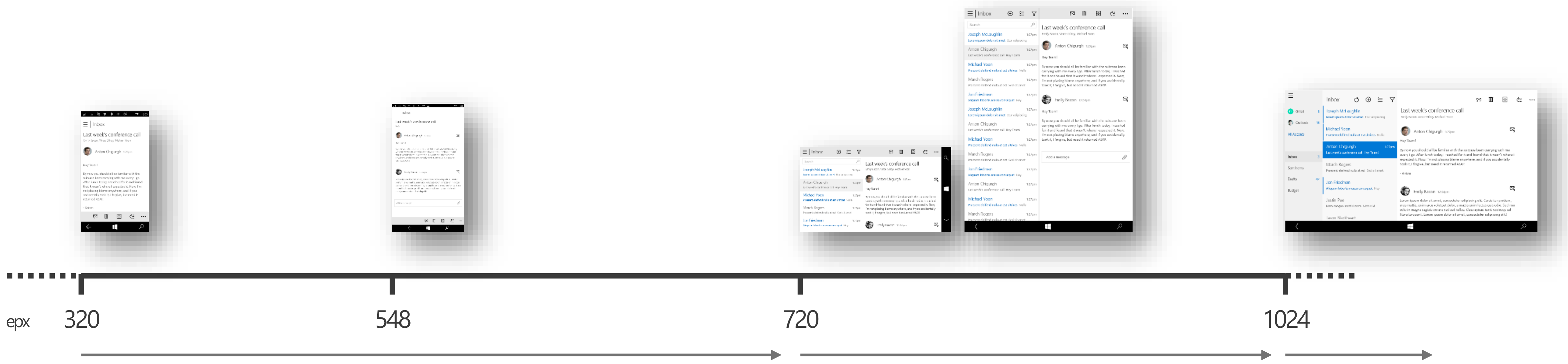
Effective pixel



Ignore scale, resolution, & dpi.
Design in Effective Pixels

Windows

Snap points



phone

- limited landscape support
- one frame at a time
- sys tray on the left
- steering wheel on the right
- 4+ actions on the bottom
- tabs are centered

- limited landscape support
- one frame at a time
- sys tray on the left
- steering wheel on the right
- 4+ actions on the bottom
- single column stops scaling
- tabs are centered

phablet & tablet

- full landscape support
- two frames
- actions at the top
- one "..." visible - TBD
- tabs are left aligned
- Show search field if search was represented as an icon on smaller devices

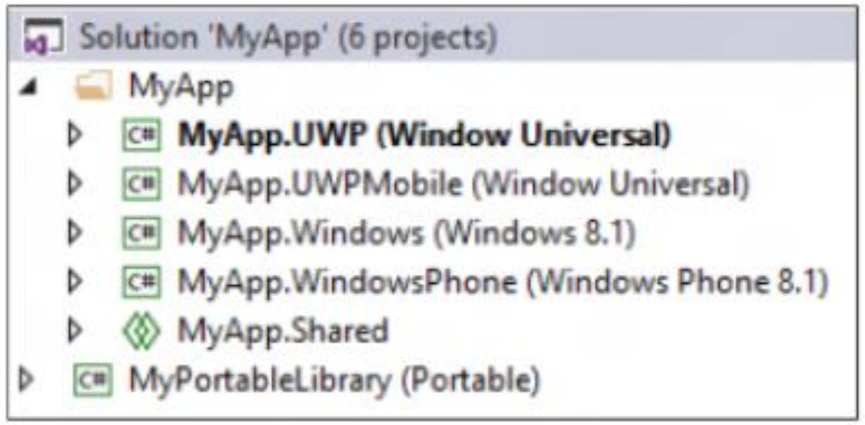
desktop

- full landscape support
- three frames
- compact nav pane
- actions at the top
- one "..." visible
- tabs left aligned

Nothing is stopping you from creating a multi-headed solution

Windows

Dedicated, targeted apps



Solution 'MyApp' (6 projects)
MyApp
MyApp.UWP (Window Universal)
MyApp.UWPMobile (Window Universal)
MyApp.Windows (Windows 8.1)
MyApp.WindowsPhone (Windows Phone 8.1)
MyApp.Shared
MyPortableLibrary (Portable)

Adaptive tooling

Windows

Visual States

Define XAML views

Unique layout for distinct states

Simplify animation

Automatically implement state transitions

Build in Blend

Design and preview states and transitions

Visual states let designers define many looks of a view

Windows

Adaptive triggers are a zero-code solution

Adaptive triggers

Code-free state transition

Two built-in triggers

MinWindowHeight (Taller than this)
MinWindowWidth (Wider than this)

```
<VisualState x:Name="VisualState500min">
    <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="501" />
    </VisualState.StateTriggers>
</VisualState>
```

How to set the visual state

VisualStateManager.Goto(element, state, transition)

```
public MainPage()
{
    this.InitializeComponent();
    this.SizeChanged += (s, e) =>
    {
        var state = "VisualState000min";
        if (e.NewSize.Width > 500)
            state = "VisualState500min";
        else if (e.NewSize.Width > 800)
            state = "VisualState800min";
        else if (e.NewSize.Width > 1000)
            state = "VisualState1000min";
        VisualStateManager.GoToState(this, state, true);
    };
}
```

http://windows.Microsoft.com

Visual states

DEMO

Windows

Managed languages are more efficient than ever

Windows

Every Windows app will be compiled with .Net Native

Windows

.NET Native

Next generation compiler in the cloud

Every Windows apps, only Windows app (right now)

Apps use the standard C++ optimizer

As optimizer performance improves, so does .Net native

Apps with .Net bootstrapper

Includes garbage collection

There is no runtime

This is machine code

Real benefits with .Net Native

50% faster average startup time

14% less average memory usage

More information:

MVA Course:
A Developer Guide to Windows 10

June 8 - 12
MVP Webcast Windows 10 Developer Readiness:
http://aka.ms/Win10MVP

Microsoft

futurice